American International Journal of Sciences and Engineering Research
ISSN 2641-0303 (Print) ISSN 2641-0311 (Online)
Vol. 2, No. 2; 2019
Impact Factor: 8.9
Published by American Center of Science and Education, USA

Autonomous 4WD Smart Car Parallel Self-Parking System by Using Fuzzy Logic Controller

Md Mamunur Rashid,
Bsc.Student, Faculty of Engineering
Electrical and Electronics Engineering Department
Al-Madinah International University, Malaysia
Email: mamunursiraj93@gmail.com

Mirza Mustafizur Rahman,
Bsc.Student ,Faculty of Engineering
Electrical and Electronics Engineering Department
Al-Madinah International University, Malaysia
Email: mahinboss96@gmail.com

Md Rashidul Islam,
Bsc.Student ,Faculty of Engineering
Electrical and Electronics Engineering Department
Al-Madinah International University, Malaysia
Email: rashidulde@gmail.com

Omar Naser Alwahedy,
Bsc.Student ,Faculty of Engineering
Electrical and Electronics Engineering Department
Al-Madinah International University, Malaysia
Email: monster.omar2@gmail.com

Abubakar Abdullahi
Bsc.Student ,Faculty of Engineering
Electrical and Electronics Engineering Department
Al-Madinah International University, Malaysia
Email: abubakarabdullahi536@gmail.com

Received: January 19, 2019 **Accepted:** January 25, 2019 **Online Published:** February 1, 2019

Abstract
Automatic car parking obtains information about available parking space, process it and then place the car at certain position. It is inevitable for the people to update with the growing technology and generally people are facing problems on parking vehicles in parking slot in a city. The Automatic car parking which enables the user to find the nearest parking area and gives availability of parking slots in that respective parking area with the help of LCD display and it mainly focus on reducing the time in finding the parking slots and also it avoids the unnecessary travelling through filled parking slots in a parking area. Thus, it reduces the fuel consumption which in turn reduces carbon footprints in an atmosphere. Sometimes, it is very difficult to find a suitable parking place in parking lot. This research paper has proposed a suitable solution to this problem.

Keywords: Autonomous 4WD, Smart Car Parallel, Self-Parking System, Fuzzy Logic Controller.

1. Introduction

1.1 Background

Parallel parking can be challenging for drivers, including myself. The typical modern day car does not contain any systems in place to make parking easier. The main goal of this project is to design a simple prototype parking system that can perform parallel parking maneuvers autonomously. This system would include a series of proximity sensors as well as a central microprocessor that controls the car. This system would ideally work on both a scale model of a car as well as a life-sized car. This project explores how sensor input and an algorithm can be used for practical applications. This paper covers the various components used to create the parking system, including an Arduino microprocessor, ultrasonic and infrared sensors, H-bridges, and servo motors. It also includes the parking algorithm implemented within the system.

As the population increased in the cities, the usage of vehicles got increased. It causes problem for parking which leads to traffic congestion, driver frustration, and air pollution. When we visit the various public places like shopping malls, multiplex cinema hall and hotels during the festival time or weekends it creates more parking problem. In the recent research found that a driver takes nearly 8 minutes to park his vehicle because here spends more time in searching the parking lot. The searching leads to 30 to 40 percent to traffic congestion. Here we going to see how to reduce the parking problem automatic car parking using offerings are transforming cities by improving infrastructure, creating more efficient and cost effective municipal services, enhancing public transportation, reducing traffic congestion, and keeping citizens safe and more engaged in the community. Car parking is an issue of significance both at the local and at the strategic level of planning. This project's main purpose is to produce a real life solution to the car parking problem which the whole world is facing frequently. People usually roam around in the parking lots trying to find a suitable place to park in to solve that problem we have created the automatic car parking system, using an open source hardware, programmable sensors and the use of computers to provide an interface to understand the digital output produced.

1.2 Problem statement

Most of the drivers nowadays having a problem of parking especially when the parking space is tight. The driver will normally take some times to parking and even sometimes it will result not that nice…Therefore, this research has carried out to help the drivers to save time , easy parking and gives them safety.

Nowadays, most of the office buildings and shopping mall had built underground parking and multilevel parking to overcome the number of cars which is increasing rapidly. However, drivers are still difficult to find an available parking slot to park their car. The process of looking for a parking lot is time consuming, confusing and wasting fuel as well. At this point of time, someone may miss or late for their important event. This might cause frustration for the drivers. Eventually the effect of lacking parking slot will causes the officer to have bad mood or consumer to leave the shopping mall without purchase anything. The side effect of this problem is serious and need a better solution to handle it.

1.3 Research objectives

The objectives of this research are:

- To develop an automatic parking system which are parallel parking and side parking by using a fuzzy logic controller
- To evaluate the performances of the vehicles in terms of time and movement of the vehicles in each parking space
- To validate the effectiveness of fuzzy system in experiments

1.4 Scope and Limitations

1.4.1. Scope of the project

The project is focused on achieving a single task (automatic parking) by integration of sensors and actuators controlled by microcontroller and strategy planning/coding, therefore the vehicle platform is not built from the parts but from modifying a RC toy car instead for saving the time. There are generally three kinds of parking patterns: parallel, front/back-in perpendicular, and with an angle (usually 45 degrees), and this project is just focused on the parallel parking. The modified toy car is expected to do the following tasks in a complete automatic parking process:

- Drive along an imitated road-side environment and detect the distance from the car to the road-side obstacles such as parked cars or just curb on the right-hand side.

- Once the length of a parking space larger than the length of the car plus a buffering distance is detected, the car will stop automatically.
- Perform a smooth and efficient parking behavior according to the relative positions of the car and the parking space.

➢ To achieve such a project, we have the following work lined up:

- **Designing** of a data collection device that collect the current, voltage, real, reactive power of the place where the audit is being performed through the help of sensors, transducers etc.
- **Coding** of an application that conducts the audit on the basis of non-intrusive algorithm. It includes the complete database of the appliances that may be present.
- **Arduino** (graphical user interface) that displays the result of the project in the form of graphs and tables.

1.4.2 Project algorithm and sensors position

In autonomous parking, we need to create algorithms and position sensors according to certain assumptions. Our assumptions will be as follows in this project. In the scenario, the left side of the road will consist of walls and park areas. As you can see on the video, there are 4 sensors in total, 2 on the left side of the car and one on the rear and front side.

The two sensors on the left side of the car understand that the wall is 15 cm smaller than the measured value and move forward. It records this in memory. The two sensors on the edge measure continuously, and when these values are the same as the resultant values, you have to decide how to park.

1.5 Significance of Project

The significance of this project is as below:

1) Space saving:

All APS takes advantage of a common concept to decrease the area of parking spaces - removing the driver and passengers from the car before it is parked. With either fully automated or semi-automated APS, the car is driven up to an entry point to the APS and the driver and passengers exit the car. The car is then moved automatically or semi-automatically (with some attendant action required) to its parking space.

The space-saving provided by the APS, compared to the multi-story parking garage, is derived primarily from a significant reduction in space not directly related to the parking of the car:

- Parking space width and depth (and distances between parking spaces) are dramatically reduced since no allowance need be made for driving the car into the parking space or for the opening of car doors (for drivers and passengers)
- No driving lanes or ramps are needed to drive the car to/from the entrance/exit to a parking space
- Ceiling height is minimized since there is no pedestrian traffic (drivers and passengers) in the parking area, and
- No walkways, stairways or elevators are needed to accommodate pedestrians in the parking area.

With the elimination of ramps, driving lanes, pedestrians and the reduction in ceiling heights, the APS requires substantially less structural material than the multi-story parking garage. Many APS utilize a steel framework (some use thin concrete slabs) rather than the monolithic concrete design of the multi-story parking garage. These factors contribute to an overall volume reduction and further space savings for the APS.

Other significance :

2) The parked cars and their contents are more secure since there is no public access to parked cars.
3) Minor parking lot damage such as scrapes and dents are eliminated.
4) Drivers and passengers are safer not having to walk through parking lots or garages.
5) Driving around in search of a parking space is eliminated, thereby reducing engine emissions.
6) Only minimal ventilation and lighting systems are needed.
7) Handicap access is improved.
8) The volume and visual impact of the parking structure is minimized.
9) Shorter construction time.

2. Literature Review

2.1 Related previous research

The smart parking system implemented mainly in the Europe, United States and Japan is developed with the incorporation of advanced technologies and researches from various academic disciplines [6]. Now-a-days, there is a rapid growth in parking system. Manpower is needed for each car parking slot to select a parking slot manually and give direction to drive properly into slot [6]. So, there is a need to develop an automatic parking system which will reduce manual work as well as will be useful for careful parking of cars and other vehicles [7]. Parking system routinely experience parking related challenges, especially in the urban and metropolitan areas. While doing a survey we have found that this automatic car parking system has been proposed by various researchers using different technology. In some paper some researchers have proposed this system using Around View Monitor(AVM). In their paper they have discusses fusion of AVM and ultrasonic sensor, used to detect the vacant parking slot in the automatic car parking system. The AVM provides a virtually 360-degree scene of the car in bird ‘s eye view. The AVM helps the driver to maneuver into parking spots. Through the bird ‘s eye view, a driver can check for obstacle around the vehicle. First, the parking slot marking detected in the AVM image sequence. A tree structure-based method detects the parking slot marking using individual AVM image sequence and image registration technique. Second, empty slot is detected using ultrasonic sensors. The probability of parking slot occupancy is calculated utilizing ultrasonic sensor data acquired while the vehicle is passing by parking slots, and finally the selected empty slot is tracked and the vehicle is properly parked in selected parking slots [7]. Some other researchers have discussed this system using another technology i.e. GSM Technology. The functionality of the technology is that user sends a message to the GSM modem which is placed at the parking end. The GSM modem will send a conformation message to the user whether the slot is vacant or not. If it is vacant then the user has to message the exact time and duration he/she wants to park the vehicle in the parking slot. Then the GSM modem will send a password and the parking lot number to access the reserved parking lot. Once the conformation message has been sent, the counter for the reservation time will automatically start for sending message [8]. Another paper attempts to discuss this system using FPGA Technology. In their paper they have discuss how to implement an automatic car parking system using FPGA technology, where the access in the parking which is made by barrier, if there are vacancies with the lifting of the barrier a ticket is issued with a client code and there starts a timer for measuring the time left in the parking. The analog signals transferred through a digital analog converter as input signals in the FPGA. To work with FPGA Xilinx software has to be used [9]. Another paper discusses a system using some digital key along with some robotic technique. when a car enters the entry of the automated car parking system, an IR detection subsystem detects the presence. Then the driver is promoted to enter a valid key and to choose the option of either parking or retrieving the car. Each key is checked for accuracy and assigned a designated parking slot. Upon entering the correct key, car is picked up along with the pallet from the stack system and placed in the designated spot. When drivers return to pick up the car he enters the valid key for which the system will check in its database and the car is return back to the drive way. The stack system will pull down the pallets to make room for incoming pallet. The system includes robotic lift with motors for picking the car and placing it in the designating spots [10]. Another paper discusses a system where microcontroller 89S51 has been used, in their paper they have discussed a system which is automated with the user being given a unique ID corresponding to the trolley being allocated to him/her. The idea is to park and move cars with no disturbance to the already parked cars in their system [11]. some other researchers have discussed this system using RFID. According to their system, the vehicle owner has to first register the vehicle with the parking owner and get the RFID tag. When the car has to be parked, the RFID tag is placed near the RFID reader, which is installed near the entry gate of the parking lot. As soon as the RFID tag is read by the reader, the system automatically deducts the specified amount from the RFID tag and the entry gate boomer opens to allow the car inside the parking area. At the same time, the parking counter increments by one. Similarly, the door is opened at the exit gate and the parking counter decremented [12]. After doing study on various system using various technology, we have tried to discuss a system using Radio frequency technology (RFID), IR (infrared)sensors, Microcontroller. RFID technology is very useful in automation of vehicle parking system in mall/building.

There are many kinds of methods that have been proposed by many researchers. Those methods whether use single intelligent or combinations of two intelligent. Xiaochuan Wang and Simon X. Yang (5) have developneuro-fuzzy control system for obstacle avoidance of a nonholonomic mobile robot. They combined four infrared sensors for distance to obstacle detection around the mobile robot. The distance information is processed by the proposed neuro-fuzzy controller to adjust the velocities of two separate driven wheels of the robot. They constructed eighteen fuzzy

rules for the obstacle avoidance. Based on their research it showed that the development of two algorithms is more effective than using single algorithm.

Besides that, Gustavo Pessin, Fernando Osório, Alberto Y. Hata and Denis F. Wolf [6] give an idea to develop multirobot system by using combination of two algorithm (Genetic Algorithm and Artificial Neural Network). In this project, Genetic Algorithm being used to planning, evolves positioning strategy for mobile robot performance. They developed a mobile robot with distance sensor that being control by multilayer perceptron (MLP) in ANN. ANN was trained and also to control the robot's actuators. Its allow mobile robot to move in dynamic environment with obstacle avoidance. Based on their results, it shows that the ANN satisfactorily controls the mobile robot.

In addition, M. A. Jeffril and N. Sariff [7] also proposed the same concept. This paper describes that the approaches are efficient to avoid few number and shape of static obstacle in environment. They constructed fuzzy rules by grouping IR sensor which means each of sensors sharing the same rule. This project proved that the mobile robot will able to avoid and explore in the environments based on Neural Network controller learning algorithm as well as increase the overall performance of the robot itself. VelappaGanapathy, Soh Chin Yun, and Jefry [8] Ng have developed mobile robot by using Fuzzy Logic and Artificial Neural Network that focused on exploring the four combinations of training algorithms. They proposed the concept of Path Remembering that will make the mobile robot to come out from acute obstacles. In order to prevent the mobile robot re-entering the same acute obstacle they used virtual wall building method. In order to reach the goal, they determined each of the angles.

Furthermore,T.Khelchandra, H. Jie, and S. Debnath [9] was design an path planning mobile robot using neuro-fuzzy technique to helping each of limitation. Theycome out with new method to solving motion of mobile robot in static environment. ANN is being trained to identifies shorten path or choose suitable path to reach goal and the fuzzy logic used to reduce ANN limitation in performance. Although there are blockage in their path the can choose the collisions free path from a lot of path. In other research, Hema,Paulraj M, Abdul Hamid,, K.F.Sim, RajkumarPalaniappan [10] uses the vision sensor to recognize objects and obstacles in front of the robot. The ultrasonic sensor being interfaced to helps avoid obstacles around the robot and to estimate the distance of a detected object. The output of the proposed sensor system aids the mobile robot with a gripper system to navigate and to pick up the objects that are lying on the floor. However, only the vision system is highlighted in this paper. A simple neural network model was developed to identify obstacles and objects.

Regarding on those researches, it can summarized that the uses of Fuzzy Logic, Artificial Neural Network and other algorithms by using different method gives different performances for the desired output. Therefore, for this project, the hybrid technique includes Fuzzy Logic and ANN is proposed to solve path planning obstacles avoidance problem in a static environment. In this project the fuzzy rules is construct to use as controller. The concept to take all the possibility for all sensors is used in constructing the fuzzy rules. Then, the addition of another algorithm which is ANN will be used to train the behavior of the robot that corresponds to the IR sensors on the robot modules. This approach is focused on the ability of the robot to move as well as avoid the obstacles in any different of complexity of environment.

2.2 Hardware Implementation

2.2.1 Arduino

Arduino is basically an open-source computer hardware/software platform for building digital devices and interactive objects that can sense and control the physical world around them. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. It comes with an open supply hardware feature that permits users to develop their own kit. The software of the Arduino is well-suited to all kinds of operation systems like Linux, Windows, and Macintosh, etc. It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed. For beginners, it is very simple to use and also cheap. It can be used to create such devices that can interact with the environment using sensors and actuators. Some common examples include robot, thermostats and motion detectors. This paper introduces the use of Arduino in one such area, that is, Automatic car parking, so that people can become aware of where free parking place is available and save time, while avoiding traffic congestion.

2.2.2 Ultrasonic Sensors [x4]

The Arduino Ultrasonic Range Detection Sensor is used with Arduino in order to calculate distances from objects. So, if we start with the Arduino Ultrasonic Range Detection Sensor, it's an IC that works by sending an ultrasound

pulse at around 40 KHz. It then waits and listens for the pulse to echo back, calculating the time taken in microseconds (1 microsecond = 1.0 × 10-6 seconds). We can trigger a pulse as fast as 20 times a second and it can determine objects up to 10 meters away and as near as 4cm.It needs a 5Vpower supply to run. And then it waits and listens for the pulse to echo back, by calculating the time taken in microseconds. Adding the Arduino Ultrasonic Range Detection Sensor to the Arduino is very easy, only 4 pins to worry about. Power, Ground, Trigger and Echo. Since it needs 5V and Arduino provides 5V we are obviously going to use this to power it.

Fig 1: Ultrasonic sensor

2.2.3 Arduino Uno [x1]

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 6 analog inputs, 14 digital input/output pins (of which 6 can be used as PWM outputs), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; all we have to do is simply connect in to PC with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Uno board is the first in a series of USB Arduino board and the reference model for the Arduino platform.

2.2.4 Mega Arduino

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

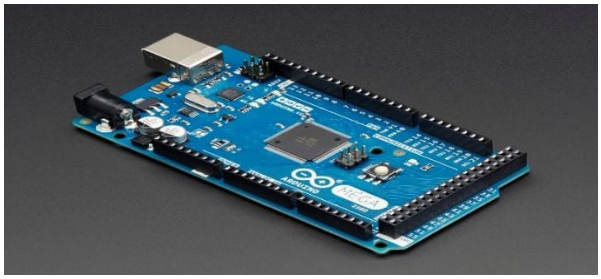


Fig 2: Mega Arduino

- **Technical details**

Dimensions:

- Length:101.98mm/4.01in
- Width:53.63mm/2.11in
- Height:15.29mm/0.60in
- Weight:34.9g/1.23oz

- **Specs:**

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

2.2.5 The Arduino motor Shield

Overview:

The Arduino motor Shield is a shield that lets you control various loads that a typical Arduino pin cannot drive. The motor shield has quite a few features such as current measuring and the ability to drive a single stepper motor. At the heart of this shield is the L298P dual full bridge driver that can handle up to 3 amps for very short durations or 2 amps continuously per channel. Thorough example code is available for all the sections in the attached zipped folder.

2.2.6 Power Supply requirements

The motors attached to the motor shield need an ample power supply. By using the USB connection, the current will often be limited to 500mA or 1 amp. Many motors will draw more current than the amount the USB source can supply. To reduce the risk of possible damange to a usb port an external power supply should be used.

2.2.6 Direction Control

To control the motor's direction, Pin 12 (Channel A) and Pin 13 (Channel B) are used. To drive the motor forward* this pins needs to be brought high. The pins can be driven low to put the motors into reverse.

Important note: Changing direction rapidly can cause unexpected effects. From a mechanical standpoint, going from forward to reverse rapidly could potentially damage a gear box. From an electrical standpoint, it can cause large current and voltage spikes. To resolve these issues, a motor needs to be taken from one direction to another with a small pause in-between. An example of this can be found in BasicControl.ino, attached.

*Since the "forward" direction of the motors depends on application, for this tutorial "forward" will refer to positive voltage on the + screw terminal on the shield.

2.2.7 Speed Control

To control the motors speed Pin 3 (Channel A) and Pin 11 (Channel B) can use PWM signals to vary the speed of the motors. To use the PWM feature on the arduino the analogWrite function needs to be called. In the function a pin needs to be defined and a speed between 0-255 needs to defined. An example of this can be found in Basic Control.ino, attached.

2.2.8 Current Sensing

Another feature of the Arduino Motor Shield is the ability to determine the amount of current the motor (or any inductive load) is drawing. Current sensing can be useful for robotics applications, such as traction control and determining if the robot is pushing an object. The current sense pins are A0 (Channel A) and A1 (Channel B). The Motor Shield will output 3.3v on the current sense pins when the maximum channel current (2 amps) is reached.

After a small amount of math, it can be determined that each integer will represent 2.96mA. So for example, if the analogRead(A0) produces a value of 121 the motor (or load) is drawing 0.36 amps. An example of this can be found in Current_to_Serial.ino, attached.

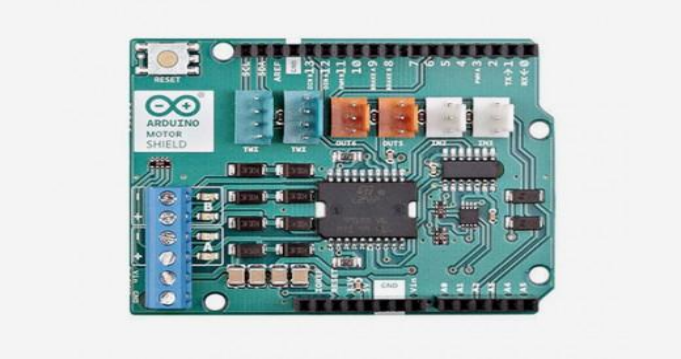


Fig 3: Arduino motor shield

This system will indicate empty and filled car parking slots at the entry. Now a days Car parking at shopping malls and markets is becoming a big issue and is causing to traffic jam. To avoid this problem, we are designing this system to indicate empty slots and filled slots, so that a car driving person can directly take his car to that particular empty slot. We will indicate the slot state by LEDs and also in LCD screen. We will use IR sensors to sense the presence of car in the slot.

2.3 Software Implementation

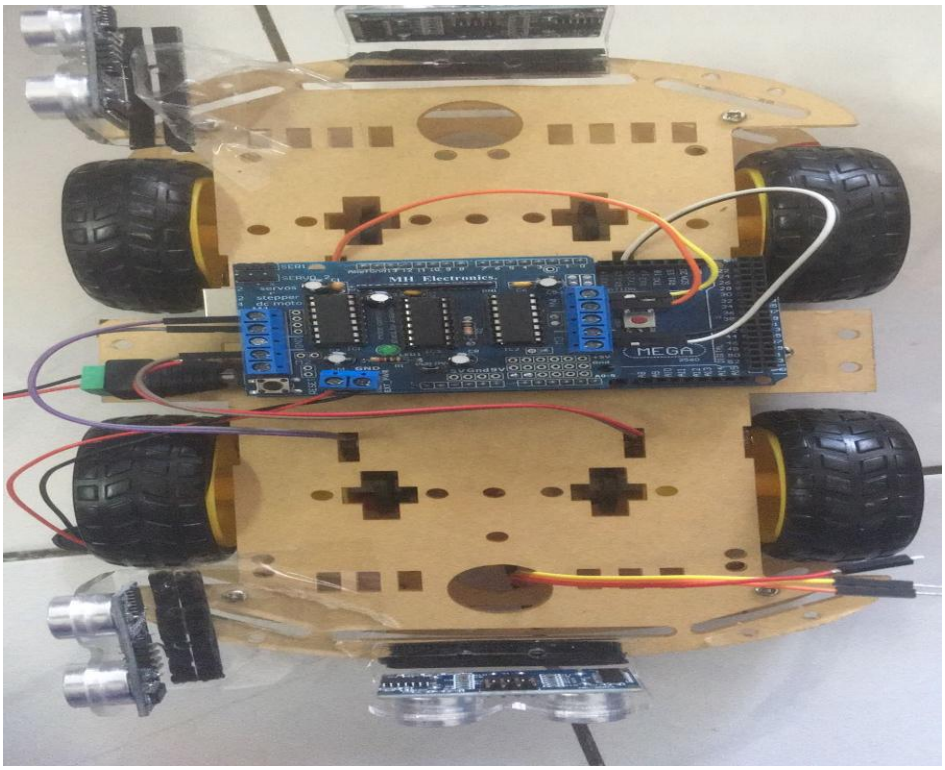
This project is designed by using main software which are Arduino. This software is used to simulate with a static environment avoiding obstacle. The performance of the car robot will be observed graphically in this software. In Arduino, special programming language is used for allowing user to control the robot. Another which is functioned for constructing the Artificial Neural Network that will train the robot behavior. This software will achieve the project's objectives.

3. Research Methodology

3.1 Automatic parking system design

3.1.1 Overall Project

In this chapter the result of our project will be stated accordingly. The final car includes the elements detailed above. Two Ultrasonic sensors are secured along the left side and 1 in the front and 1 at the back of the car. The sensor placement is shown in Figure 1 down below. Each sensor is connected to its own analog pin on the Arduino, so that each pin can read in specific outputs from their corresponding sensors. The Arduino also controls a servo that turns the front wheels of the car. The rotation speed and direction of the rear wheels are controlled by the motor



Which sensor to use:

After comparing to other sensors available, the ultrasonic sensor is the most accurate one, the IR sensor is not very accurate also doesn't work fine outside in the sunlight, and the available IR sensors had a range less than the ultrasonic sensors.

Detection a car not a moving object:

This is solved by using more than one ultrasonic sensor to detect from different points and distances. Also by using delay and multiple readings to ensure that there is a car.

Running out of pins:

Our project utilized all of the pins in the Arduino, and in order to overcome this problem, we used shift Register.

DC Motor power: Detection a car not a moving object:

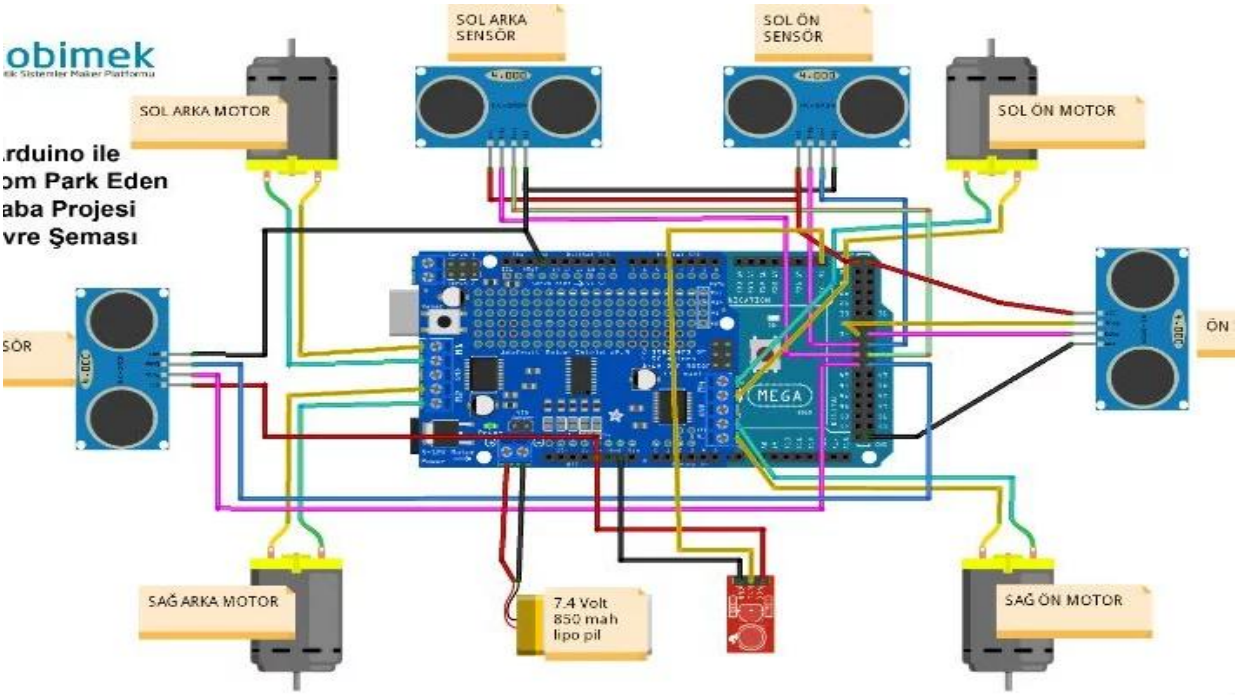
DC Motor get slow at parking because of low power , so sometimes will get small error regarding to low power to motors. So need to increase the voltage battery to overcome this problem.

3.1.2 Materials used in the project

- Arduino Mega
- Adafruit Motor Shield
- 4 Dc Motor Robot Kit

- 4 Pieces HC-SR04 Ultrasonic Sensor
- LM 393 Infrared Speed Sensor
- Lipo Battery (7.4V 850 mAh is enough)
- Jumper cables

3.1.3 Circuit Diagram



Pin Connections of Ultrasonic Sensors

Front Sensor => Trig Pin: D34, Echo Pin: D35

Left Front Sensor => Trig Pin: D36, Echo Pin: D37

Left Rear Sensor => Trig Pin: D38, Echo Pin: D39

Rear Sensor => Trig Pin: D40, Echo Pin: D41

Motor Shield Dc Motor Pin Connections
Left Front Motor => M4

Right Front Motor => M3

Left Rear Motor => M1

Right Rear Engine => M2

LM393 Speed Sensor Pin Connections VCC => 5V: OUT => D21: GND => GND

3.1.4 System Flowchart

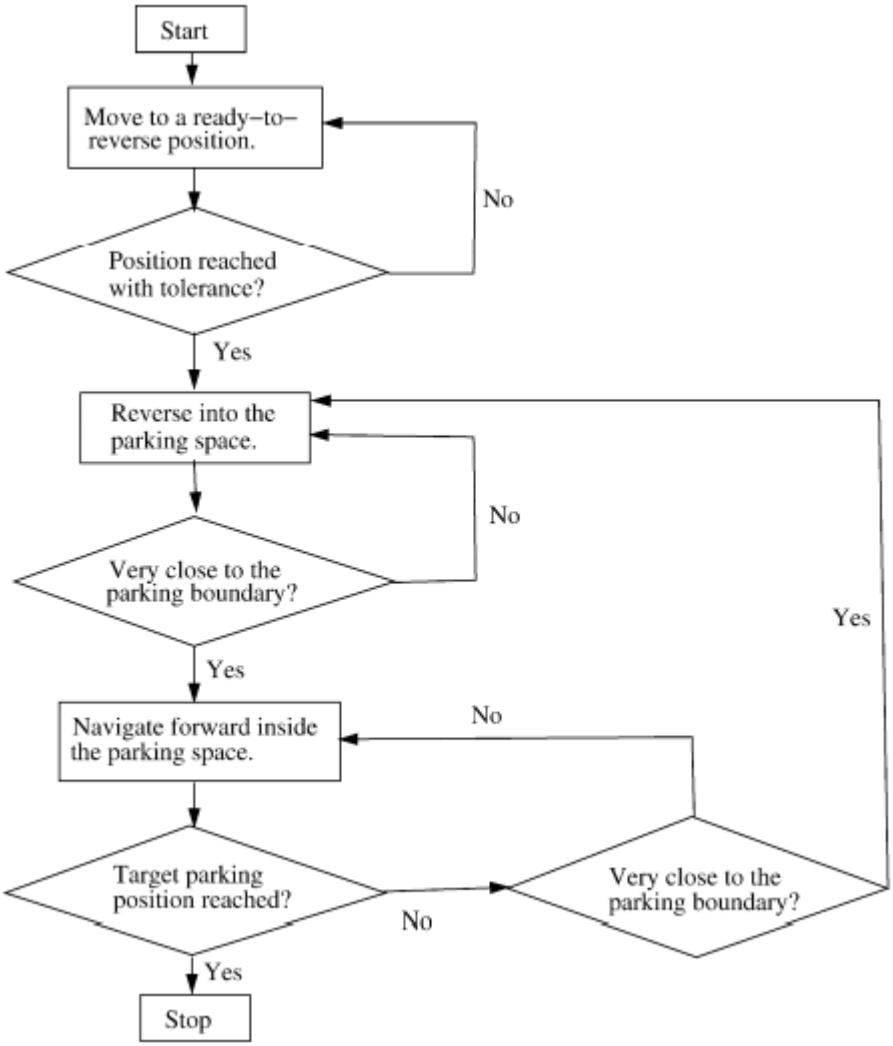


Fig. 5. Flowchart of the parallel parking algorithm.

3.2 Fuzzy logic controller design

3.2.1 System Algorithm using Fuzzy logic

The two sensors on the left side of the car understand that the wall is 15 cm smaller than the measured value and move forward. It records this in memory. The two sensors on the edge measure continuously, and when these values are the same as the resultant values, you have to decide how to park.

Park Method Selection Algorithm

- Case 1: If the measured value is bigger than the car and smaller than the length of the car, the parallel parking system will operate.

- Case 2: If the measured value is greater than the length of the car, the robot will park vertically.
- Case 3: If the measured value is shorter than the expected values car will stop.

➢ In details :

The Decision model that begins the parking procedures are based on three premises depending upon the available parking space or distance between cars.
Case 1: If the measured value is bigger than the car and smaller than the length of the car, the parallel parking system will operate.
In this case, the car crosses the parking area and the car stops when two sensors on the side see the wall again. He comes back a little and turns right 45 degrees. While moving backwards, the rear sensor goes into the park area by measuring and starts to turn left. During the left movement, the sensors at the edges measure continuously and the two sensors continue to turn left until the

measured value equals each other. Stop when you are equal. The front sensor measures and goes forward until it is small by 10 cm and stops when it is small by 10 cm. Parking is over.
Case 2: If the measured value is greater than the length of the car, the robot will park vertically.
If the sensors at the edges measure the value too much over the length of the car, the car stops and turns 90 degrees to the left. They start moving towards the parking lot. At this time, the front sensor continuously measures and the car stops if the measured value is less than 10cm. Park operation is completed.
Case 3: Stop.

3.2.2 Parallel parking algorithm

In this case, the car crosses the parking area and the car stops when two sensors on the side see the wall again. He comes back a little and turns right 45 degrees. While moving backwards, the rear sensor goes into the park area by measuring and starts to turn left. During the left movement, the sensors at the edges measure continuously and the two sensors continue to turn left until the measured value equals each other. Stop when you are equal. The front sensor measures and goes forward until it is small by 10 cm and stops when it is small by 10 cm. Parking is over.

3.2.3 Vertical Parking Algorithm

If the sensors at the edges measure the value too much over the length of the car, the car stops and turns 90 degrees to the left. They start moving towards the parking lot. At this time, the front sensor continuously measures and the car stops if the measured value is less than 10cm. Park operation is completed.

3.2.4 Output rules of Fuzzy Logic

The Decision model that begins the parking procedures are based on three premises depending upon the available parking space or distance between cars (DBC):
1. If parking space is large (over 28 cm), forward parking movements are performed.
2. If parking space is enough (more than 15 cm), backward parking movements are performed.
3. But if parking space is short (less than 15 cm), no action is performed.
Hence, parking space is one of five measurements inputs in the system which helps the system to decide whether it is possible to park the car or not. The other four measurements correspond to distance from side walk (DFS), Fig. 4; distance from front car (DFF), Fig. 5; distance from back car (DFB), Fig. 6; and inclination (Inclination), Fig. 7 .Once system has acquired the parking system value and it is in the permissible parking range, the car must take an initial position so that the scaled car perform movements to park and it stops until it satisfy the desirable parking conditions of rules. Then the fuzzy model which decides which action has to be taken at the beginning of the process is named decision model. It determines whether backward model or forward model has to be performer after parking space has been measured. One has to understand actions as the movements that an autonomous control has to perform instead a human driver. And these actions are in function of the controlling system outputs. The controlling car outputs are the following: (1) Car direction or tires angle, Fig. 8, (2) Speed of Car, Fig. 9 and (3) direction of

movement, Fig. 9, Forwards or Backwards. The fuzzy process normally begins when the input variables are compared with their belonged fuzzy sets, then a membership degree value is delivered and fuzzy outputs are calculated. Finally, fuzzy outputs are defuzzicated by centroid method, at least in this case. Each model has a specific number of rules that are taken to develop inference step of the mathematical base. To develop the inference step "Minimum—Maximum" or Mandami is used. Table 1 shows the ten rules decision model which development, in function of the space between cars measurement, makes the car moves towards an initial position to begin parking forward or backward. Once having decided which king of parking to perform (backward or forward) and having achieved the initial position, other rules are applied to control the car.

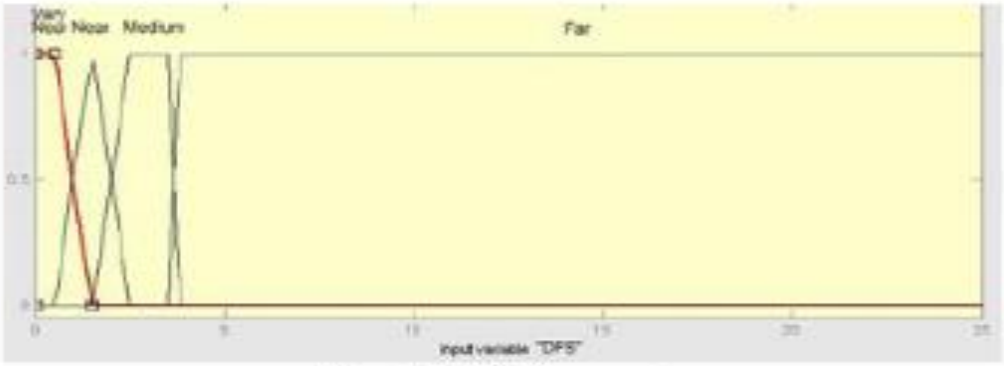


Fig. 4. DFS fuzzy set.

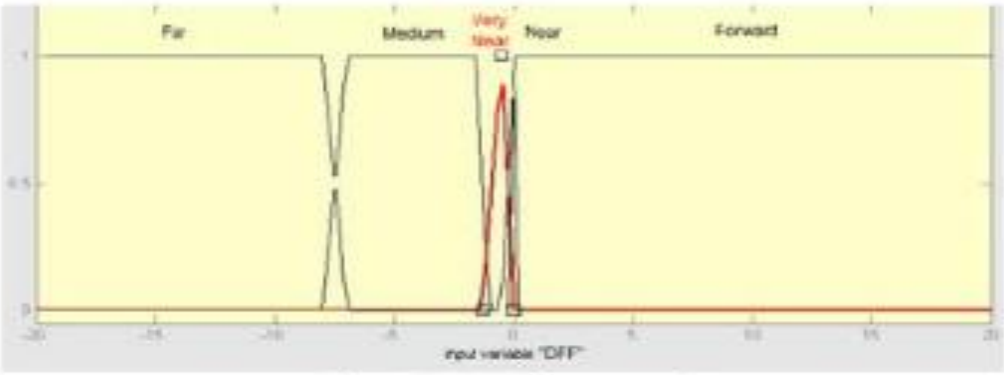


Fig. 5. DFF fuzzy set.'

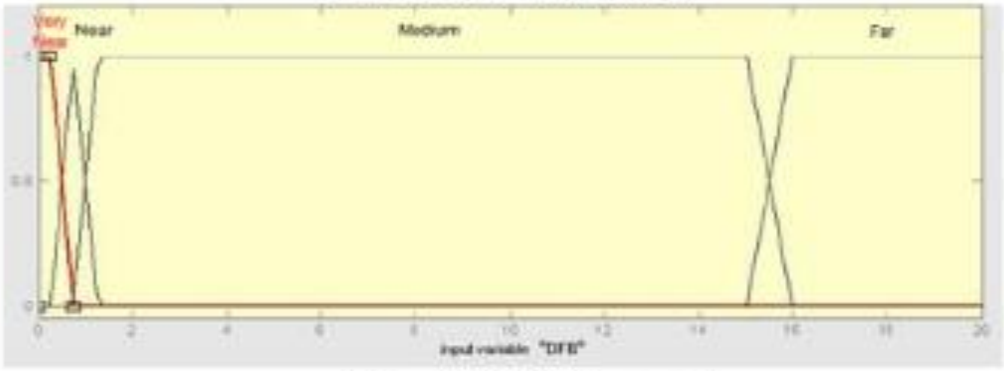


Fig. 6. DFB fuzzy set.

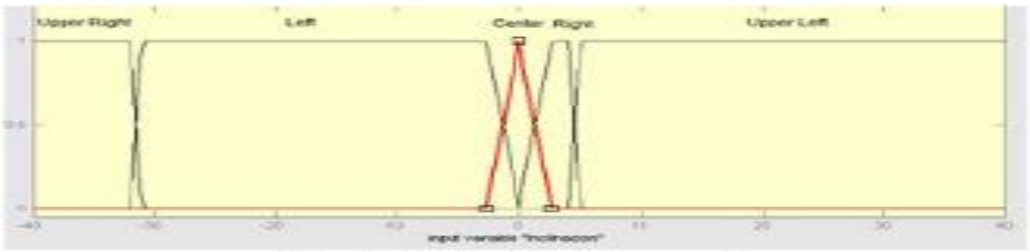


Fig. 7. Inclination fuzzy set.

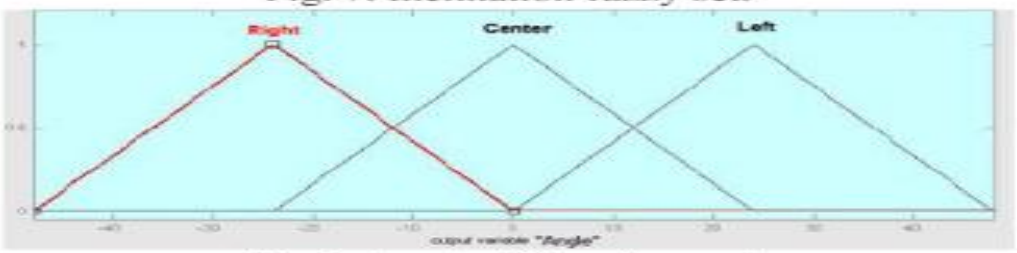


Fig. 8. Angle of tires fuzzy set.

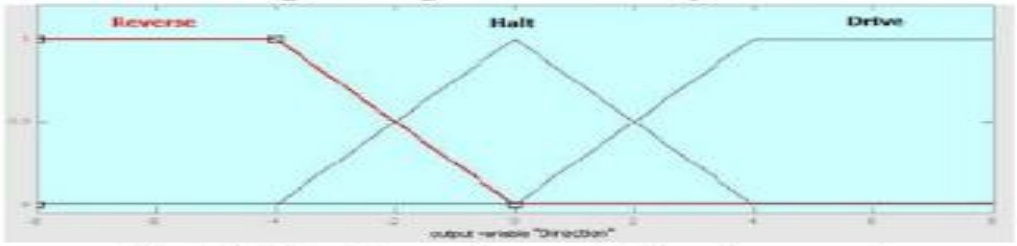


Fig. 9. Direction and speed of car fuzzy set.

Table 1: Decision Model Rules

Decision Rules								
Rule	X	Activated	Constant Decision	Terminado In	Angle	Direction	Decision	Terminado Out
1		No			Center	Halt	No	No
2	Atrasado	Yes	Forward	No	Center	Drive	No	No
3	Posición	Yes	Forward	No	Center	Halt	Forward	Yes
4	Adelantado	Yes	Forward	No	Center	Reverse	No	No
5	Atrasado	Yes	Backward	No	Center	Drive	No	No
6	Posición	Yes	Backward	No	Center	Halt	Backward	Yes
7	Adelantado	Yes	Backward	No	Center	Reverse	No	No
8			No	Yes	Center	Halt	No	Yes
9			Forward	Yes	Center	Halt	Forward	Yes
10			Backward	Yes	Center	Halt	Backward	Yes

3.2.5 Design of Fuzzy Controller for Automatic Parking

This section discusses the design of the automatic parking system. The data obtained by the car-like robot in motion are derived from the ultrasonic range sensor, so the location of the sensor is crucial. The obtained data are imported into the automatic parking system, and then the automatic parking system controls the vehicle's dynamic state. First, the actual parking situation is simulated. When the vehicle runs to a fixed point, the user issues the automatic stopping instruction, so the peripheral parking information is searched. Whether there is the parking space for reversing the car into a garage or roadside parking is identified according to the information obtained by the ultrasonic range sensor. The sensed state is fed back to the user side, and the automatic parking behavior pattern is executed. Figure 3 shows the flow chart of this automatic parking system.

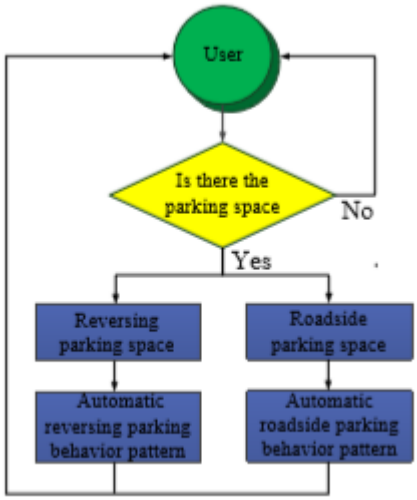


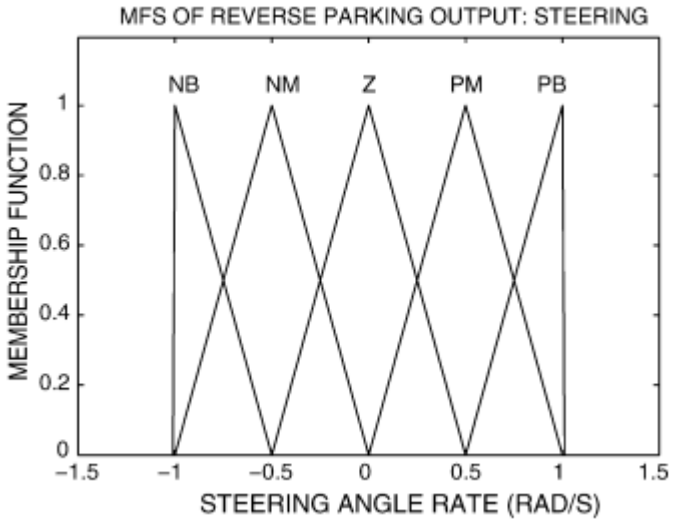
Figure 4: The flow chart of the proposed Automatic parking system

3.2.6 FLC for navigating the vehicle forward inside the parking space

The developed fuzzy logic controller has one input, orientation angle θ, and one output, the steering rate $\dot{\theta}$. The membership functions and fuzzy inference rules are exactly the same as that of the orientation adjustment described in the first step since this step accomplishes essentially the same task,that is adjusting the orientation while simultaneously moving the vehicle forward.

Table 3
Fuzzy rules for the backing up step

	$X_{a1} \backslash y_{d1}$	S	B	VB
$\theta = N$	S	PB	PB	
	B	PM	PB	PB
	VB			PM
$\theta = Z$	S	Z	Z	
	B	Z	PB	PB
	VB			Z
$\theta = P$	S	NB	Z	
	B	NM	Z	PM
	VB			NB



3.3 Testing

The vehicle localization errors in odometry accumulate over time due to inaccuracies in the kinematic model, precision limitations of the encoders, and wheel slippage. The errors can be cleared whenever the onboard PC is restarted. In one set of parking experiments, the entire parking process was repeated five times without reinitializing the robot odometry. It was observed that, due to localization errors, the ready-to-reverse position reached by the first parking step was different for each parking process. The desired ready-to-reverse position was x = 1.951 m, y = 1.248 m and θ = 0 rad relative to the local coordinate system. Table 4 compares the odometry readings from the onboard PC and manually measured data from the five experiments. It is seen that while the data from odometry reading matched the desired target position well, the data from the real (i.e., the manual) measurement did not. In particular, localization errors along the x-axis were much bigger than that along the y-axis and orientation angle, and the later experiments yielded larger errors. The differences in the ready-to-reverse positions led to different reverse-motion trajectories in the second step of the parking process. In the first four experiments, the mobile robot was successfully parked inside the parking space by the fuzzy controller.

In the fifth experiment, the vehicle localization error along x-axis was already 1.951 − 1.5560 = 0.395 m. Under this case, the robot hit the edge BK showing the limitation of the algorithm.

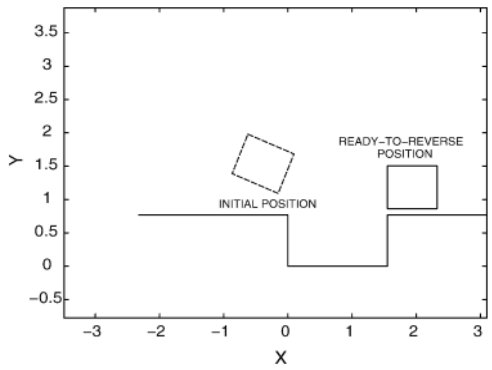


Fig. 4. Reach a ready-to-reverse position.

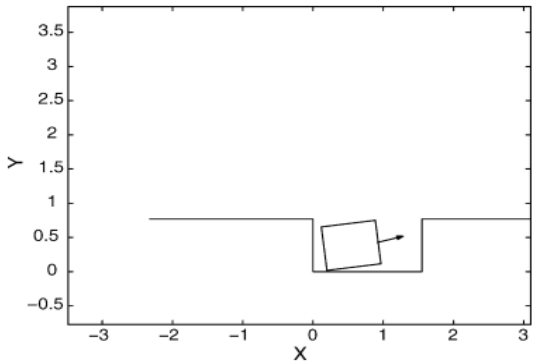


Fig. 7. Adjust forward inside the space.

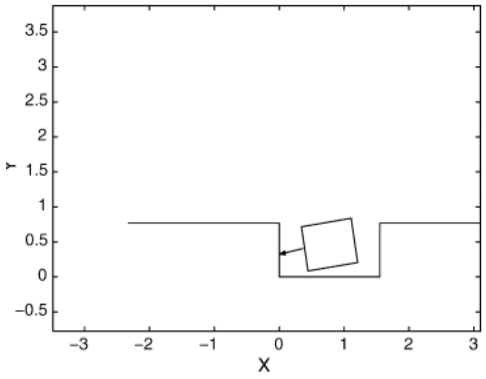


Fig. 6. Reverse with decreasing θ.

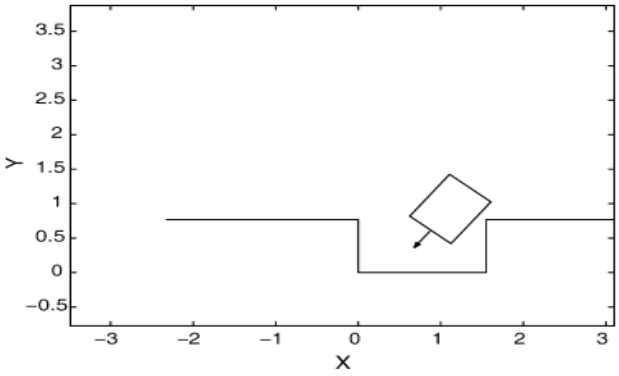


Fig. 5. Reverse with increasing θ.

Table 4 Vehicle localization errors

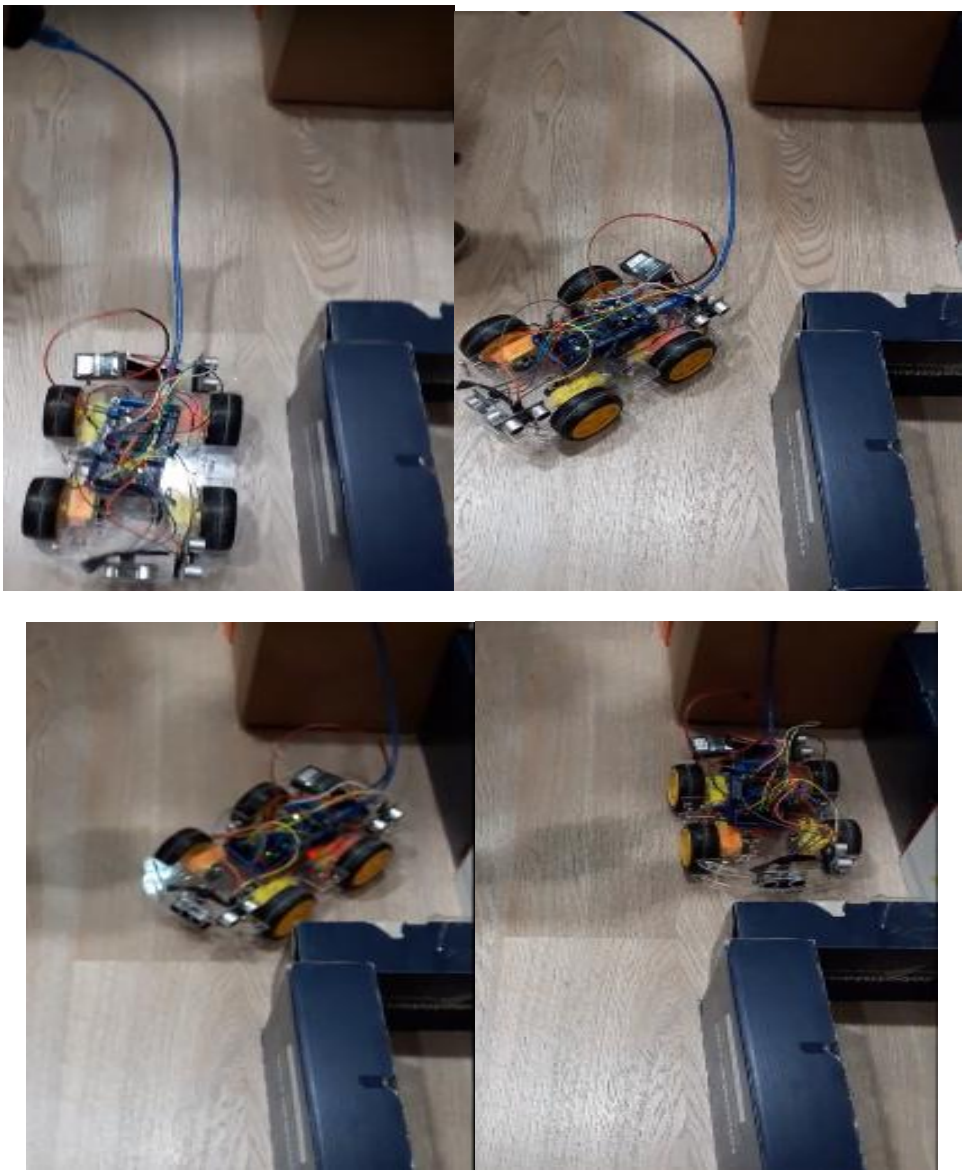
Experiment no.	Odometry x (m)	Odometry y (m)	Odometry θ (rad)	Measured x (m)	Measured y (m)	Measured θ (rad)
1	1.9596	1.2416	−7.7182 × l0−4	1.7650	1.2365	0.0120
2	1.9525	1.2402	−6.0643 × l0−4	1.7095	1.2440	0.0159
3	1.9579	1.2455	−10.1991 × l0−4	1.6613	1.2290	−0.0649
4	1.9599	1.2456	−5.7887 × l0−4	1.6075	1.2370	−0.0156
5	1.9559	1.2485	−9.0965 × l0−4	1.5560	1.2485	−0.0097

4. Results and Discussion

4.1 Parallel Parking System

In this case, the car crosses the parking area and the car stops when two sensors on the side see the wall again. He comes back a little and turns right 45 degrees. While moving backwards, the rear sensor goes into the park area by measuring and starts to turn left. During the left movement, the sensors at the edges measure ontinuously and the two sensors continue to turn left until the measured value equals each other. Stop when you are equal. The front sensor measures and goes forward until it is small by 10 cm and stops when it is small by 10 cm. Parking is over.

4.1.1 Movement of vehicles



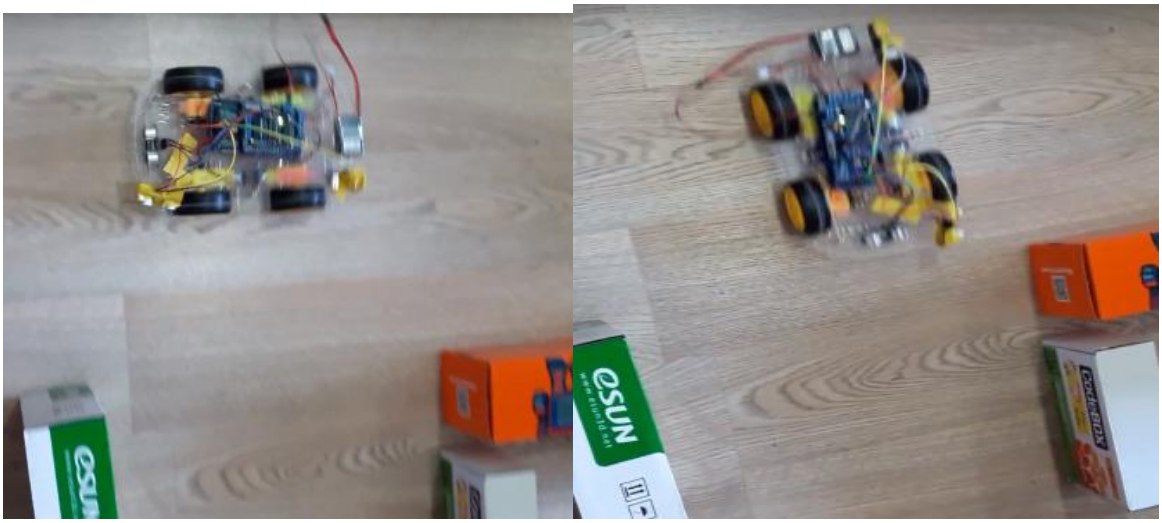
4.1.2 Movement, time, distance

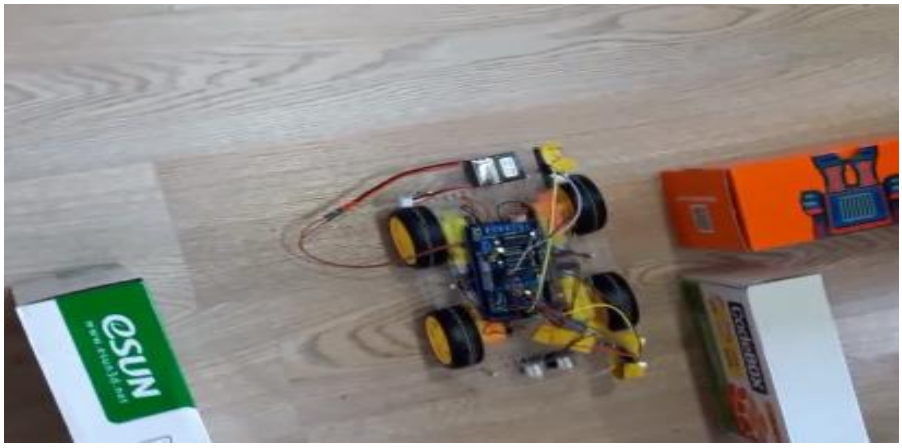
No. Run	Achieve parking	Time	Iteration	Distance
1	Success	9 sec	34	center
2	Success	10 sec	34	center
3	Failed	15 sec	30	Shifted 20 %
4	Failed	11 sec	35	Shifted 15 %
5	Success	10 sec	34	center

4.2 Vertical parking system

If the sensors at the edges measure the value too much over the length of the car, the car stops and turns 90 degrees to the left. They start moving towards the parking lot. At this time, the front sensor continuously measures and the car stops if the measured value is less than 10cm. Park operation is completed.

4.2.1 Movement of vehicles





4.2.2 Movement, time and distance

No Run	Achieve parking	Time	Iteration	Distance
1	Success	6 sec	34	Center
2	success	7 sec	34	Center
3	success	10 sec	36	Center
4	failed	11 sec	30	Shifted right 2 cm from the center.
5	Success	8 sec	33	Semi center

4.3 Steering angle system

	-90		+90		0	
	L	R	L	R	L	R
Forward	0	0	0	0	1	1
Reverse	0	0	0	0	1	1
right	1	1	1	1	0	0
left	1	1	0	0	0	0

Table 1. The sensor and the motor table

5. Conclusion

Automatic car parking is one of the important factors in traffic area, multiplex's, apartments, shopping malls, etc. Car parking system that is discussed here is automated without human being that means if the driver leaves it at the starting of the system the elevator takes the car to the parking slot.

As a conclusion, this development of car robot by using Fuzzy Logic (FL) gives the best performances in term of efficiency, flexibility and robustness. Based on the result obtained, the objectives of this project are achieved. The car robot is successfully detecting the empty parking slot in a particular environment. The time taken by the robot to finish the 20 numbers of loops is also reduced. In addition, the concept of taking all the possibilities for all sensors which considered all 3 rules give high impact to the movement of the robot compared to only take a one rule in random sensors.

Automated car parking is very useful in this mordent world. Where finding a small place have become a big problem. This system is present in some cars in the world.

6. Schedule and Costing

6.1 Project management

We can describe the whole project management by some steps:

1. Discussion about the idea of our capstone project.

2. Thinking about our project title.

3. Selecting the project title.

4. Submission the project proposal.

5. Programing, coding and Simulink using Arduino.

6. Complete the Hardwiring task.

7. The report writing and project submission.

Every task has been assigned weeks base and the following Gantt Chart according to the work involved in attaining that some categories. The major time has been assigned to the coding, Simulink and Arduino application since it is the core of the project and in order to accomplish our goal it has been to be completed with dedication and determination.

GANTT CHART:

Step's	W1	W2	W3	W4	W5	w6	W7	W8	W9	W10	W11	W12
Discussion about the project idea												
Thinking about our project title												
Selecting the project title.												
Submission the project proposal.												
Programing, coding												
Using Arduino.												
Hardwiring task.												
report writing and project submission												

6.2 Costing of the project

The component	Price (RM)-estimated
Smart car body (contains the 4 WD motor)	130
Arduino mega	80
8 ultrasonic sensors	90
Small components contains (micro controllerships, breadboard ,batteries and connectors cables)	200
Adafruit Motor Shield	30
LM 393 Infrared Speed Sensor	35
Replacement component (instead of the burned one while testing) extra	140
Total	705 RM

References

[1] Li, T. H. S. and Chang, S. J. "Autonomous Fuzzy Parking Control of a Car-Like Mobile Robot." IEEE Transactions on Systems, July 451-465, 2003. 12

[2] I. E. Paromtchik, and C. Leggier, "Motion Generation and Control for Parking an Autonomous Vehicle, " Proc. of the IEEE Int. Conf. on Robotics and Automation, Minneapolis, USA, (April 22-28, 1996), pp. 3117-3122

[3] Lyon, D., "Parallel parking with curvature and nonholonomic constraints." Proceedings of the Intelligent Vehicles '92 Symposium, Jul. 341-346, 1992.

[4]Murray, R. M. and Sastry, S. S. "Steering Non-holonomic Systems Using Sinusoids." Proceedings of the 29th conference on decision and control, Dec. 2097-2101. 1990.

[5] Zhu, C. and Rajamani, R. "Global Positioning System-Based Vehicle Control for Automated Parking." Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering January 1, 37-52, 2006.

[6] Noor N.M, Z Razak and Mood Yamani, ―Car Parking System: A Review of Smart Parking System and its Technology‖, Information Technology Journal, 2009.

[7] A.A Kamble and A Dehankar ―Review on Automatic Car Parking Indicator System‖, International Journal on recent and innovation trends in computing and communication, Vol 3 no.4 pp 2158-2161.

[8] K Sushma,PRaveendraBabu and J.Nageshwara Reddy, ―Reservation Based Vehicle Parking System using GSM and RFID Technology‖,International Journal of Engineering Research and Applications Vol 3 no.5 2013.

[9] R.Khan ,Z.Khan,Y.AShah,K.Ahmed,A.Manzoor and A.Ali, ―Intelligent Car Parking Management System on FPGA‖,International Journal of Computer Science issues Vol 10 no.3 2013.

[10] A.Wafa,N.Zeba, ―Automated Car Parking‖, 2012.

[12] C.Patel, M.Swami, P.Saikia, S.Shah, ―Rotary Automated Car Parking system‖,International Journal of EngineeringScience and Innovative Technology (IJESIT) Volume 4, Issue 2, March 2015.

[13] B.Waraich, ―RFID-Based Automated Vehicle parking system‖.

[14] Mariam Al-Sagban and RachedDhaouadi, "Neural-Based Navigation of a Differential-Drive Mobile Robot," in 2012 12th International Conference on Control, Automation, Robotics & Vision Guangzhou, China, 5-7th December 2012 (ICARCV 2012).

[15] A. L. Howell, R. T. R. McGrann, and R. R. Eckert, "Teaching concepts in fuzzy logic using low cost robots, PDAs, and custom software," in Frontiers in Education Conference, 2008. FIE 2008. 38th Annual, 2008, pp. T3H-7-T3H-11.

[16] U. Farooq, K. M. Hasan, G. Abbas, and M. U. Asad, "Comparative analysis of zero order Sugeno and Mamdani fuzzy logic controllers for obstacle avoidance behavior in mobile robot navigation," in Current Trends in Information Technology (CTIT), 2011 International Conference and Workshop on, 2011, pp. 113-119.

[17] R. R. Janghell, AnupamShukla, RituTiwari and Rahul Kala, "Breast Cancer Diagnosis using Artificial Neural Network Models," in 2010 International Conference on Soft Computing and Pattern Recognition on, 2010 IEEE, pp 89-94.

[18] Xiaochuan Wang and Simon X. Yang, "A Neuro-Fuzzy Approach to Obstacle Avoidance of a Nonholonomic Mobile Robot," in 2003 lntemalional Conference on Advanced Intelligent MechatronicS (AIM 2W3, 2003 IEEEIASME.

[19] Gustavo Pessin, Fernando Osório, Alberto Y. Hata and Denis F. Wolf, "Intelligent Control and Evolutionary Strategies Applied to Multirobotic Systems," in 2010 IEEE.

[20] M. A. Jeffril and N. Sariff, "The integration of fuzzy logic and artificial neural network methods for mobile robot obstacle avoidance in a static environment," in System Engineering and Technology (ICSET), 2013 IEEE 3rd International Conference on, 2013, pp. 325-330.

[21] VelappaGanapathy, Soh Chin Yun, and Jefry Ng, "Fuzzy and Neural Controllers for Acute Obstacle Avoidance in Mobile Robot Navigation," in 2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics Suntec Convention and Exhibition Center Singapore, July 14-17, 2009

[22] T. Khelchandra, H. Jie, and S. Debnath, " Path Planning of Mobile Robot with Neuro-Fuzzy Technique," in 2012 12th Internasional Conference on Intelligent Systems Design and Applications (ISDA).

[23] Hema,Paulraj M, Abdul Hamid, K.F.Sim and RajkumarPalaniappan, "An Intelligent vision System for Object Localization and Obstacle Avoidance for an Indoor Service Robot," in 2011 IEEE Student Conference on Research and Development.

Appendix

1) Psudo code of Arduino program for automatic parking system

```
/*// Autonomous Park Car

// Developed by Omar  Alwaheidi

// definitions of engines
```

```
*/
#include<AFMotor.h>
AF_DCMotorleft_front_Motor(4);
AF_DCMotorright_front_Motor(3);
AF_DCMotorleft_back_Motorr(1);
AF_DCMotorright_back_Motor(2);
#include<Ultrasonic.h>
Ultrasonic ultrasonic_back(40,41),ultrasonic_left_back(38,39),ultrasonic_left_front(36,37),ultrasonic_front(34,35);
//  define ultrasonic sensors
#define left 0 //left direction command
#define right 1 //right direction command
#define forward 2 //forward direction command
#define backward 3 //backward command
#define minimum_limit 15 //Width of the car (cm)
#define minimum_limit1 28 //the length of the car (cm)
byte park_status  = 0; //park status
int signal_pin = 21; //signal pin speed
volatile int val;
int counter = 0;
int current_status = 0;
int previous_situation = 0;
void say(int saydir)
{
for (int i = 0 ;i<= saydir; i+1)
{
val = digitalRead(signal_pin);
if (val == LOW) {

current_status = 0;
}
```

```
else {
current_status = 1;
}
if(current_status != previous_situation)
{
if(current_status == 1)
{
counter = counter + 1;
Serial.println(counter);
i = i+1;
}
else
{
i = i ;
}
previous_situation = current_status;
}
if (i == saydir)
{
left_front_Motor.run(RELEASE);
right_front_Motor.run(RELEASE);
left_back_Motorr.run(RELEASE);
right_back_Motor.run(RELEASE);
}
}
}
void motor_pinSetup()
{
left_front_Motor.run(RELEASE);
right_front_Motor.run(RELEASE);
```

```
left_back_Motorr.run(RELEASE);
right_back_Motor.run(RELEASE);
}
// Movement functions
void Robot_Movement(byte motor, byte spd)
{
if (motor == forward)
{
left_front_Motor.setSpeed(spd);
right_front_Motor.setSpeed(spd);
left_back_Motorr.setSpeed(spd);
right_back_Motor.setSpeed(spd);
left_front_Motor.run(FORWARD);
right_front_Motor.run(FORWARD);
left_back_Motorr.run(FORWARD);
right_back_Motor.run(FORWARD);
}
if (motor == backward)
{
left_front_Motor.setSpeed(spd);
right_front_Motor.setSpeed(spd);
left_back_Motorr.setSpeed(spd);
right_back_Motor.setSpeed(spd);
left_front_Motor.run(BACKWARD);
right_front_Motor.run(BACKWARD);
left_back_Motorr.run(BACKWARD);
right_back_Motor.run(BACKWARD);
}
if (motor == left)
{
```

```
left_front_Motor.setSpeed(spd);
right_front_Motor.setSpeed(spd);
left_back_Motorr.setSpeed(spd);
right_back_Motor.setSpeed(spd);
left_front_Motor.run(BACKWARD);
right_front_Motor.run(FORWARD);
left_back_Motorr.run(BACKWARD);
right_back_Motor.run(FORWARD);
}
if (motor == right)
{
left_front_Motor.setSpeed(spd);
right_front_Motor.setSpeed(spd);
left_back_Motorr.setSpeed(spd);
right_back_Motor.setSpeed(spd);
left_front_Motor.run(FORWARD);
right_front_Motor.run(BACKWARD);
left_back_Motorr.run(FORWARD);
right_back_Motor.run(BACKWARD);
}
}
void Robot_Stop()
{
left_front_Motor.run(RELEASE);
right_front_Motor.run(RELEASE);
left_back_Motorr.run(RELEASE);
right_back_Motor.run(RELEASE);
}
// Search for parking
bool Parking_Place_Control()
```

```
{
long front_Sensor = ultrasonic_front.Ranging(CM);
long right_Sensor = ultrasonic_left_front.Ranging(CM);
long right_back_Sensor =ultrasonic_left_back.Ranging(CM);
if( (right_Sensor<= minimum_limit)&&(right_back_Sensor<= minimum_limit)&&(park_status  == 0))
{
Robot_Movement(forward, 100);
park_status  = 1; Serial.println(park_status );
}
if((right_Sensor>minimum_limit)&&(right_Sensor<minimum_limit1)&&(right_back_Sensor>minimum_limit)&&(
right_back_Sensor<minimum_limit1)&&(park_status  == 1))
{
Robot_Movement(forward, 100);
park_status  = 2;Serial.println(park_status );
}
if((right_Sensor>= minimum_limit1)&&(right_back_Sensor>= minimum_limit1)&&(park_status  == 1))
{
/* Vertical Parking Decision */
Robot_Stop() ;
delay(500);
park_status  = 10;Serial.println(park_status );
}
if((right_Sensor<= minimum_limit)&&(right_back_Sensor<= minimum_limit)&&(park_status  == 2))
{
/* Parallel Parking Decision */
park_status  = 3; Serial.println(park_status );
}
return park_status ;
}
void Park_bul()
```

```
{
Parking_Place_Control();
if(park_status  == 3 )
{
Robot_Stop();Serial.println(park_status );
delay(400);
park_status  = 4;
}
if(park_status  == 4 )
{
Robot_Movement(backward,120);
say(18);
Robot_Stop();Serial.println(park_status );
delay(500);
Robot_Movement(right,150);
say(9);
Robot_Stop();
delay(500);
park_status  = 5;
}
if(park_status  == 5)
{
Robot_Movement(backward,120);
long arka_Sensor = ultrasonic_back.Ranging(CM);Serial.println(arka_Sensor)
if(arka_Sensor>0 &&arka_Sensor<= 13)
{
Robot_Stop();
delay(400);
park_status  = 6;
}
```

```
return arka_Sensor;
}
if(park_status  == 6)
{
Robot_Movement(left,150);
long right_Sensor = ultrasonic_left_front.Ranging(CM); Serial.println(right_Sensor);
long right_back_Sensor = ultrasonic_left_back.Ranging(CM); Serial.println(right_back_Sensor);
if(right_Sensor == right_back_Sensor)
{
Robot_Stop();
park_status  = 7;
}
return right_Sensor,right_back_Sensor;
}
if(park_status  == 7)
{
long front_Sensor = ultrasonic_front.Ranging(CM);
if(front_Sensor<=6)
{
Robot_Stop();
park_status  = 8;
}
else
{
Robot_Movement(forward,100);
}
return front_Sensor;
}
if (park_status  ==10)
{
```

```
Robot_Movement(left,180);
say(14);
Robot_Stop();
delay(500);
park_status  = 7;
}
}
void setup()
{
Serial.begin(9600);
attachInterrupt(5, say, CHANGE);
pinMode (signal_pin, INPUT) ;
motor_pinSetup();
}
void loop()
{
Park_bul();
}
```

2) Data sheet of Arduino mega microcontroller.

http://www.mantech.co.za/datasheets/products/a000047.pdf

3) Datasheet of Motor control shield

https://media.swymhome.com/parts/159/files/219/Overview%20with%20Example%20code?t=1491694883

4) Datasheet of LM 393 Infrared Speed Sensor

file:///C:/Users/Omar%20Alwahedy/Downloads/lm393-motor-speed-measuring-sensor-module-for-arduino%20(1).pdf

5) Related review papers topic of automatic parking system

A. Fuzzy Logic Controller forRobot Navigation in an unknown Environment.

B. Fuzzy Logic Control for An Autonomous Robot.

C. Robust Automatic parallel parking in tight spaces via fuzzy logic.

D. Self-Parking System Based in a Fuzzy Logic Approach.

E. An Intelligent Auto Parking System for Vehicles.

F. IPLMS: An Intelligent Parking Lot Management System.

G. Smart parking system.