Design and Development of an Autonomous Automotive Parking System

Mechatronics Department

Kareem Abdelkader, Omar Ehab, Shehab Gomaa, Ayman Hamdy, Abdelrahman Emad, Daniel George, Peter Sameh, Abdelrahman Abdelmoez, Youssef Yacoub

Super-visor Dr. Ahmed Abdelaziz

Pyramids Higher Institute for Engineering and Technology

2024

Abstract

This project focuses on the development a single task of the automotive autonomous driving tasks, which is Automatic Parking throughout the integration of sensors and actuators controlled by a microcontroller together with trajectory planning and coding of the proper parking path. The phases of this project will be presented in the remainder of this report, which goes from conceptual design to critical design and finally to the implementation and testing phase of our prototype. Car is an example of a non-holonomic system in which the number of control commands available is less than the number of coordinates representing its location and direction; According to non-

holonomic constraints, a shortest path algorithm for the self-parking problem under specific initial conditions has been proposed and proven. Appropriate project components have been selected to increase the efficiency and accuracy and would be explained. The platform (car model) has utilized the same nondimensional values of the real car, but at a smaller size. The stages of developing and writing the required control codes to implement the self-parking algorithms will be discussed.

Self-parking is a difficult task that requires the car to perceive its surroundings, make decisions, and control its actuators precisely. In this project, a self-parking system is proposed that uses an array of sensors, such as ultrasonic sensor, to detect the car's surroundings. The system also includes a software unit that processes sensor data and generates the proper steering control signals to drive the car on the parking path. A servo motor was used for the steering system, and two DC motors for the propulsion system. The microcontroller uses several algorithms that control the movement of the robot car to achieve accurate and reliable parking maneuvers according to the available parking area

Keywords

Development – Design – Components – Path Planning – Experimental Methods and Results

1. Introduction

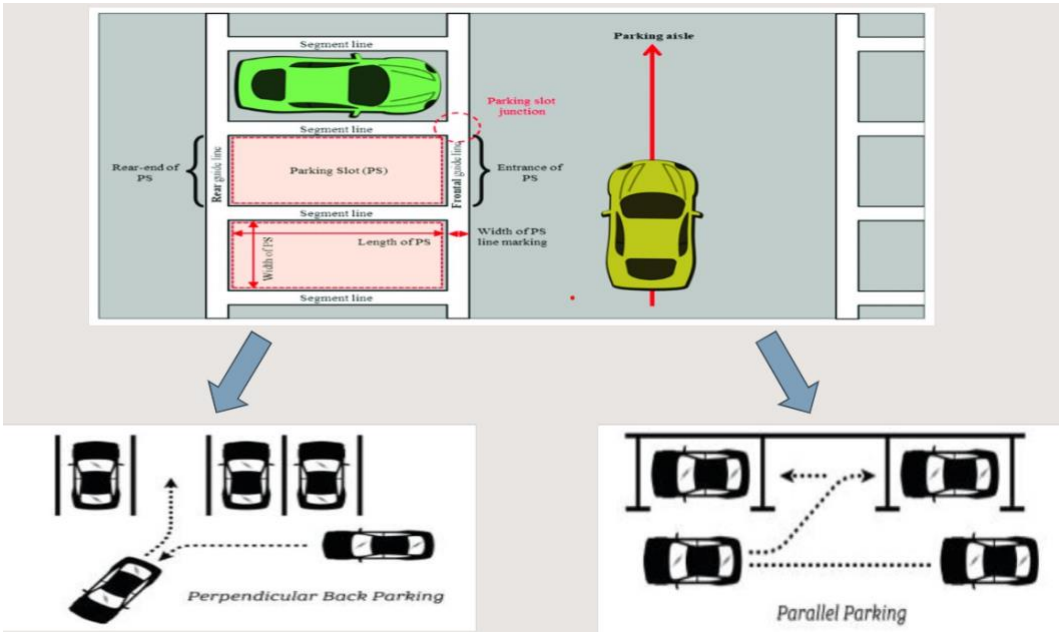
1.1 History and Motivation

This project aims at enhancing the vehicle automation through what is called “Advanced Driver Assistance Systems (ADAS)”. One of the ADAS is the Self-Parking System, which has been selected for development in this project.

Parking is a necessary maneuver that occurs repeatedly as a result of daily use of the vehicle, it is a challenge for inexperienced drivers, and it has the highest rate of vehicle collisions. Self-parking aim at: Improve parking efficiency, reduce driver fatigue, and reduce the risk of accidents.

1.2 Goals and Objectives

The goal of car Self-Parking is to efficiently achieve car parking, perpendicular/parallel to the pavement, within the smallest available parking area and without the risk of collision.



1.3 Short Description of the Project

- Developing the kinematic equations required for lane maneuvers.
- Developing an algorithm that enables the car to recognize different conditions of the surrounding environment and know the most appropriate way to park.
- Designing the car and its systems to simulate real cars.
- Proving the car's ability to park itself in different parking spaces successfully.

1.4 Literature Review and Theoretical

This section provides a short review of some related systems and introduces a theoretical background of the project, based on our research, we came across very limited products that are related to the current project.

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. 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. 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. 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. 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. 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. 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. 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. 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.

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 have developed neuro-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 give an idea to develop multi-robot 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 perception (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.

Regarding on those researches, it can summarize that the uses of Fuzzy Logic, Artificial Neural Network, Non-holonomic Systems and other algorithms by using different method gives different performances for the desired output. Therefore, for this project, the hybrid technique includes Non–holonomic Constraints Systems and RTOS is proposed to solve path planning obstacles avoidance problem in a dynamic environment.

2. Development

2.1 Overview

Based on the search for the importance of self-parking systems for cars, and with the development of technology and the use of electronics in all fields, including the automotive field, we were keen to search for applications for these systems in cars, so we searched for how this project works and what are the correct sequences and algorithms, and we reviewed the projects implemented in the field of self-parking to learn and gain experience.

Then we moved on to search for the appropriate algorithm to implement the project objectives accurately and efficiently by reviewing references and scientific research that addressed the same topic, then comparing the results with the expected results of the project and the extent to which they can be implemented through simulation programs such as MATLAB. After studying the problems, we proposed one type of the shortest collision-free paths in a specific case to approach the car to the target by following a fixed path while avoiding obstacles, and an iterative path planning algorithm was presented to park the car in a narrow space that is not sufficient to operate the parking process at the same time.

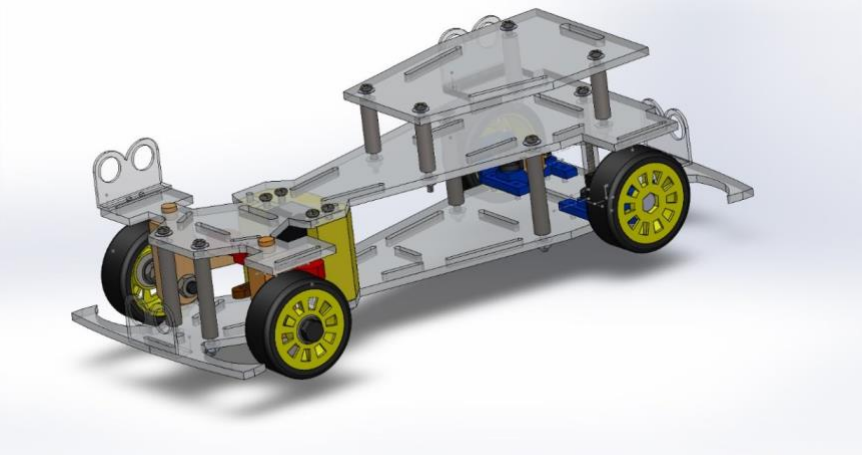
Then we moved on to work on designing a car based on the dimensions and design of real cars to achieve realism for our program using SOLIDWORK.

In parallel with the design, work was done to select the appropriate components for the project, such as SOC, sensors, communication buses, and motors. The selection was based on criteria for each component that are compatible with the project and other projects similar to ours.

Then we move on to developing the algorithm and software necessary and appropriate for the project and testing each part.

3. Design

The car platform is designed to be a scaled model for a real platform, i.e. a scale for the standard average dimensions for real cars, so that the developed algorithm for the parking maneuver would work for full scale models if divided by the scale factor.



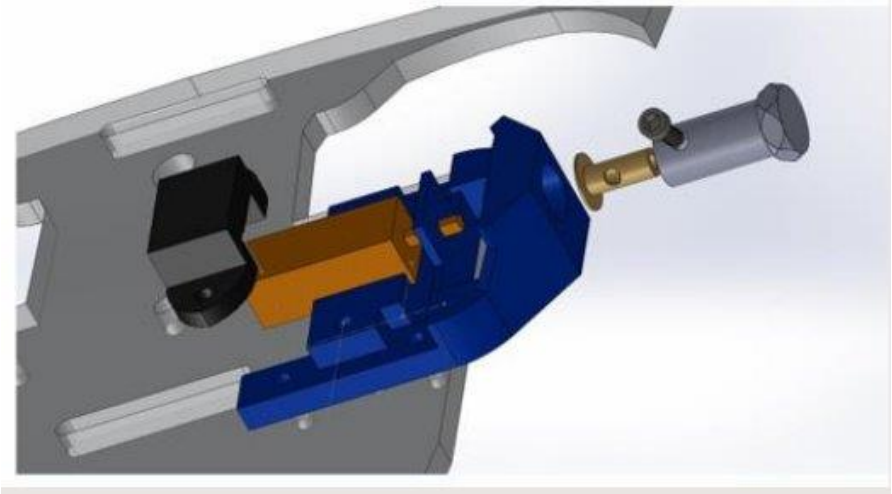
Rear Wheel Drive, is a drivetrain layout in motor vehicles where the engine power is delivered to the rear wheels, propelling the car forward.

WHY WAS THIS CHOSEN?

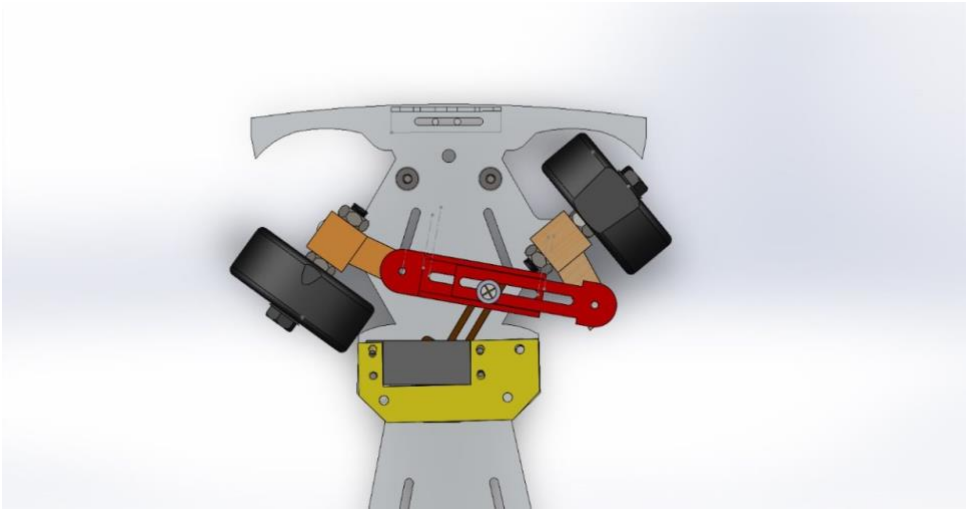
- Simpler and more robust design
- Requires less servicing
- Easier to fix when repairs are needed

Powertrain Parts:

- Wheels
- Coupling
- Housing
- Bolts & Nuts
- Motor

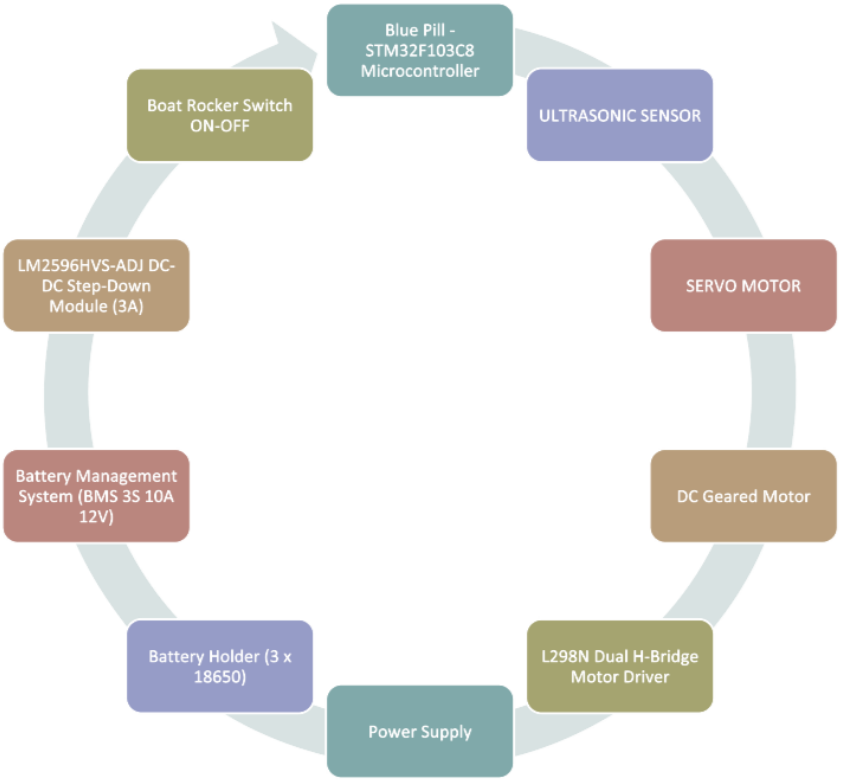


While the steering wheel appears simple, it uses scientific principles to achieve smooth handling. A key concept is Ackermann steering, which accounts for the different angles needed by the inner and outer wheels during turns. This principle ensures precise cornering and is a fundamental part of modern vehicle design. Over time, steering systems have evolved from mechanical to electronic and hydraulic, but the core principle of Ackermann steering remains essential for optimal car handling.



4. Components

To reach the final goal of the project, which is to self-park the car, the appropriate components must be used to work together in terms of the possibility of interface between them, speed, accuracy, and price. Therefore, each component was studied and compared to its counterpart through practical experiments and nominations.



4.1 SOC; Blue pill board with STM32F103c8 Arm® Microcontroller.

This board was selected to ensure that the selected hardware can run algorithms in real-time and can handle the computational requirements of sensors and actuators.

STM32F103c8 Arm® 32-bit Cortex®-M3 CPU core 72 MHz maximum frequency, 1.25 DMIPS/MHz. 64 or 128 Kbytes of Flash

memory. 20 Kbytes of SRAM. Up to 40 fast I/O ports. Seven timers. Up to nine communication interfaces (I2C, USART, CAN, LIN, USB, and SPI).

4.2 Distance measurement sensor; Ultrasonic sensor.

Ultrasonic was selected because it is more accrued, detects a range of materials: Ultrasonic position sensors can detect and measure objects irrespective of their surface or color. and needed it to know the distance (in a different phases) to uses this distance in the mathematical modeling. It uses sound waves to measure the distance. It consists of 2 transducers; one acts as transmitter and other acts as receiver. The transmission emits an ultrasound wave at 40 KHz and if it there an object it will bounce back to the receiver.

Ultrasonic Sensor Vs IR Sensor:

	Ultrasonic Sensor	**IR SENSOR**
DETECTION TYPE	continuous or discrete	Presence/absence
RANGE	Generally, longer (up to 10 meters)	Generally shorter (up to 5 meters)
ACCURACY	More accurate for larger distances	Less accurate, affected by object shape and material

4 sensors were used, according to the standards (park on right hand), so 2 sensors were placed on the right side to determine the lateral

distance. A sensor was placed at the front and rear of the car to avoid collisions when moving.

4.3 DC Motor

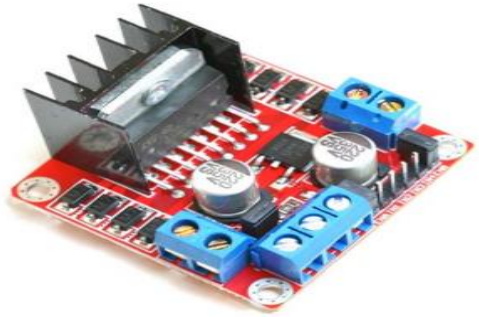
Two DC motor are used to provide power to the wheels and cause forward and backward motion, it is installed with the rear wheels.

DC motor Vs Stepper motor:

	DC MOTOR	**STEPPER MOTOR**
CONTROL	Offer variable speed control, allowing smooth acceleration and deceleration	Require a precise sequence of pulses to move a specific number of steps.
MOTION	Have continuous motion	Have incremental motion
WEIGHT AND SIZE	A smaller size and weight which makes them more suitable	Big size and weight
NOISE AND HEAT	Often quieter and generate less heat	More noise and heat generation

4.4 L298N Dual H-Bridge Motor Driver

Used to provide suitable power to the motor and to protect MCU from high voltage or current issues.



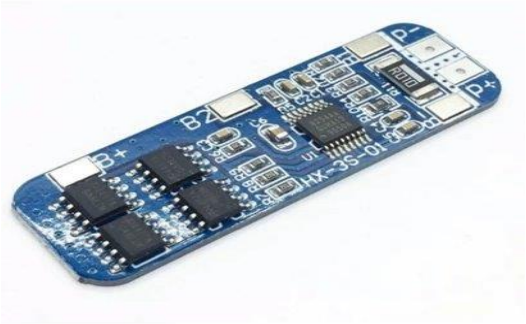
4.5 Servo Motor

It is a closed-loop system that uses position feedback to control its motion and final position. Consists of DC motor, gear box, potentiometer, and control unit. Potentiometer provides position feedback to the CU which compares current position to the target one.

4.6 Power Supply; Rechargeable lithium batteries 3.7V. We used 3 batteries.

4.7 Battery Management System (BMS)

Is technology dedicated to the oversight of a battery pack, which is an assembly of battery cells electrically organized in a row-x-column matrix configuration to enable delivery of a targeted range of voltage and current for a duration of time against expected load scenarios.



5. Path Planning Algorithm

5.1 Overview

Requires an appropriate addressing scheme that the robot can follow. To solve the robot navigation problem, we need to find answers to the three following questions:

- What is the proper starting point location?
- What is the end point location?
- What is the proper trajectory that connects the start & end points?

5.2 Vehicle Kinematic Model

Our case study will follow non-holonomic constraints for vehicles and use a low-speed method. System simulation and verification done using the MATLAB program.

Non-holonomic Systems:

- Understanding the behavior of cars, especially during maneuvers like turning.
- Its final position depends not only on the overall distance traveled but also on the specific path.

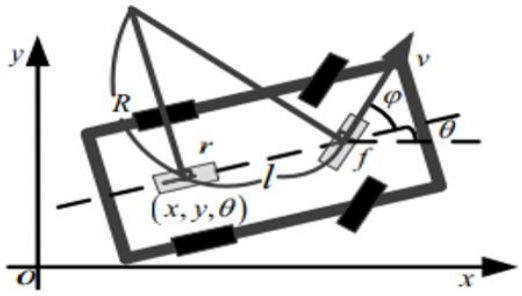
The low-speed (non-sliding) method:

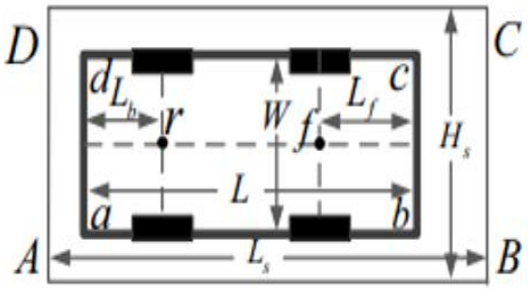
- Avoid Sliding.
- Increased response to changes in the surrounding environment.

5.3 Vehicle and Parking Space Considerations

It is assumed that the vehicle moves with the non-sliding method in the parking process because of the low speed. In the reference

coordinate system, r is the midpoint of the rear wheel, f is the midpoint of the front wheel, x=x(t) and y=y(t) are the coordinates of r, θ=θ(t) is the course angle of the car with respect to the global coordinate system, ϕ=ϕ(t) is the steering angle, v=v(t) is the velocity of f, l is the wheel base, and R is the turning radius of r.





The parking environment can be constructed with information from sensors. In which abcd represents the four corners of the car, while ABCD represents the four corners of parking space, let the length of the rear and front overhang be Lb and Lf, L and H are the length and width of the car, and the length and width of parking space, which dominate the difficulty of parking, are defined as Ls and Hs.

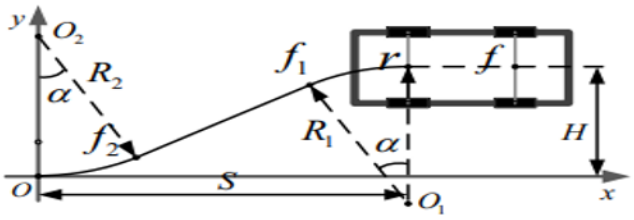
5.4 Planning Shortest Path for Parallel and Perpendicular

The working procedure of a path planner is partitioned into two parts:

- Firstly, the system decides whether the space is big enough to park by analyzing the information from the ultrasonic sensors.
- Secondly, the planner generates a feasible collision-free parking path with the consideration of choosing the appropriate start and end positions if the space meets the parking requirement.

5.4.1 Shortest Path Design for Parallel Parking

Circle Straight-Line Circle (CSC) is the chosen path,



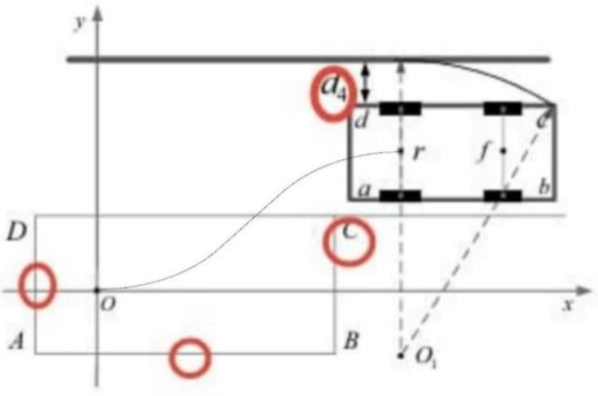
Path margins:

- From start point (r) to f1
- From f1 to f2
- From f2 to end point (O)

The path can be described as the equations:

$$\begin{cases} r \to f_1 : (x-S)^2 + \left[y - (H - R_{\min}) \right]^2 = R_{\min}^2 \\ f_1 \to f_2 : kx - y + m = 0 \\ f_2 \to O : x^2 + (y - R_{\min})^2 = R_{\min}^2 \end{cases}.$$

Collision-free regions, regions that collision may be occur:



Minimum Parking Space Length & width

$$\min L_s = \sqrt{R_b^{\,2} - \left(R_{\min} + W/2 + d_2 - H_s\right)^2} + L_b \qquad \min H_s = W + d_2$$

Collision with AB (d2)

$$d_2 = \sqrt{\left(R_{\min} + W/2\right)^2 + L_b^{\,2}} - R_{\min} - W/2$$

Collision between ab and CD (d3)

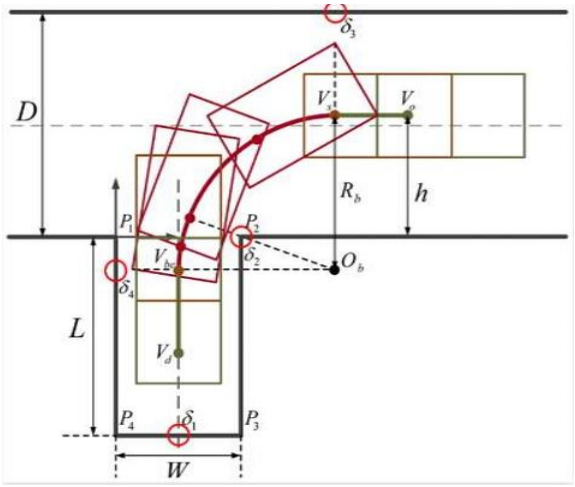
$$d_3 = \left(R_{\min} - W/2\right) - \sqrt{\left(R_{\min} - W/2\right)^2 - \left(S - L_s + L_b\right)^2} \quad (16)$$

Track Width (d4)

$$d_4 = R_b + (R_{min} + W/2)$$

5.4.2 Shortest Path Design for Perpendicular Parking

Circle Straight-Line (CS) is the chosen path,



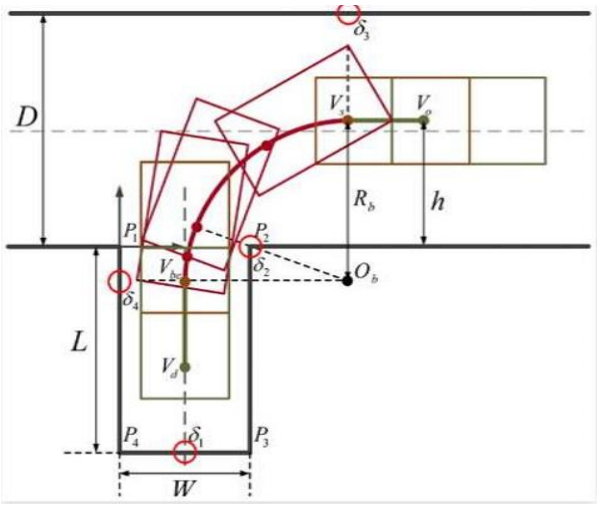
Path margins,

- From start point (Vs) to (Vbc)
- From (Vbc) to (Vd)

The path can be described as the equations:

$$\begin{cases} x_d = W / 2 \\ y_d = \max(L_r - L_a, \delta_1 + L_r - L) \\ \theta_d = \pi / 2 \end{cases}$$

Collision-free regions, regions that collision may be occur:



Minimum Parking Space Length & width

$$W = \sqrt{l_b^2 + \left(R_b + \frac{W_a}{2}\right)^2} - \sqrt{\left(R_b - \frac{W_a}{2}\right)^2 - s^2}$$

$$L = L_{car} + d_1$$

The collision of right side of rear axle with P2 set as

$$\sqrt{\left(R_b - \frac{W}{2}\right)^2 + (h - R_b)^2} < R_b - \frac{W_a}{2} - \delta_2$$

Track Width

$$D = \sqrt{(l + l_f)^2 + \left(R_b + \frac{W_a}{2}\right)^2}$$

5.5 Model Verification and Simulation

The car trajectory has been modeled & programed utilizing MATLAB, as a soft test bench before hardware realization. This design phase is very important as an initial qualification of the path planning & an easy tool for path planning debugging.

The actual dimensions of the vehicle should be considered, the parameters of the test vehicle are:

- The car's length (L_Car) = 41.5 cm (about 1.36 ft)
- The car's width (W_Car) = 18 cm (about 7.09 in)
- The wheelbase (L_axi) = 27 cm (about 10.63 in)
- The front overhang (Lf) = 6.25 cm (about 2.46 in)
- The rear overhang (Lb) = 8.25 cm (about 3.25 in)
- The maximum turning angle for steering wheel is within a range of (45:35) for each side, The maximum turning angle (phi_max) = 45
- The ratio between wheelbase and cross base, the coefficients average values are gained is (0.55, 0.56, 0.66), The ratio between wheelbase and cross base (kb) = W_Car / L_axi = 0.66
- The turning ratio (R_min) = L_axi / Phi_max = 27 cm (about 10.63 in)

5.5.1 Parking Space Environment for Parallel Parking

To specify the parking space environment, the following consideration parameters should be taken:

- Minimum side clearance (d2) = 0.87 cm (about 0.34 in)
- Minimum front clearance (Rb) = 35.81 cm (about 1.17 ft)
- Minimum length of parking space (Ls) = 39 cm (about 1.28 ft)

- Minimum Width of parking space (Hs) = 16 cm (about 6.3 in)

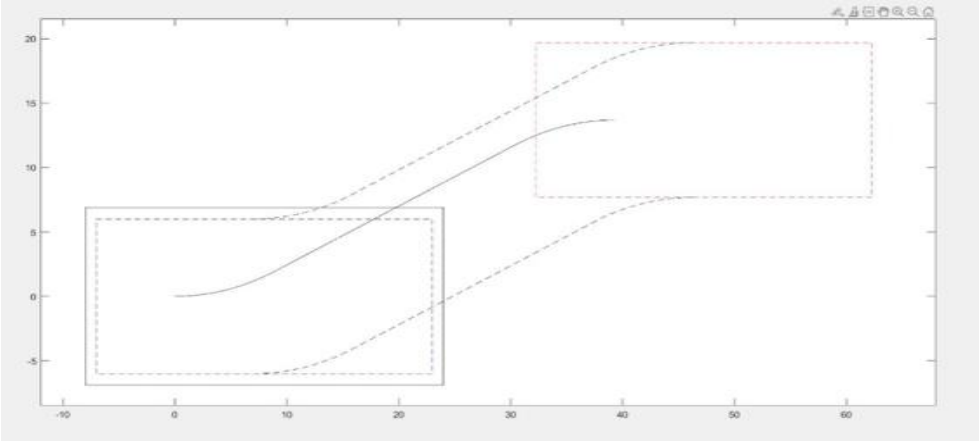
Initial position and orientation of the vehicle:

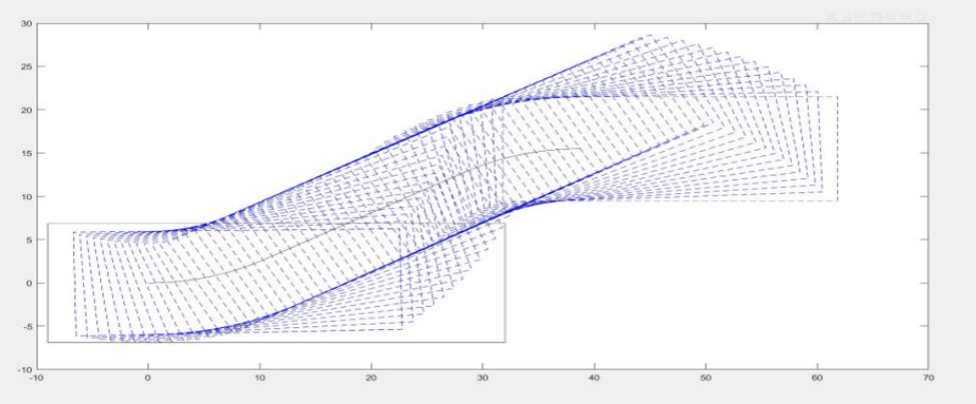
- The horizontal coordinate of start position (S) = Ls
- The minimum parallel distance between the car and the obstacle (d3)
- The vertical coordinate of start position (H) = W_Car + d3

We have three phases of movement,

1. From start position r to f1
2. From f1 to f2
3. From f2 to end position O

With this information, we can create a path plan for our robot, create the environment surrounding it, and know whether these plans are useful and serve their purpose or not.





5.5.2 Parking Space Environment for Perpendicular Parking

To evaluate the proposed geometric path planning and steering controls for perpendicular reverse parking in one trial, simulation results using MATLAB were conducted.

To specify the parking space environment, the following consideration parameters should be taken:

- Minimum side clearance = 1.6 cm (about 0.63 in)
- Minimum track width (D) = 20 cm (about 7.87 in)
- Minimum length of parking space (Ls) = 20 cm (about 7.87 in)
- Minimum Width of parking space (Hs) = 40 cm (about 1.31 ft)

Initial position and orientation of the vehicle:

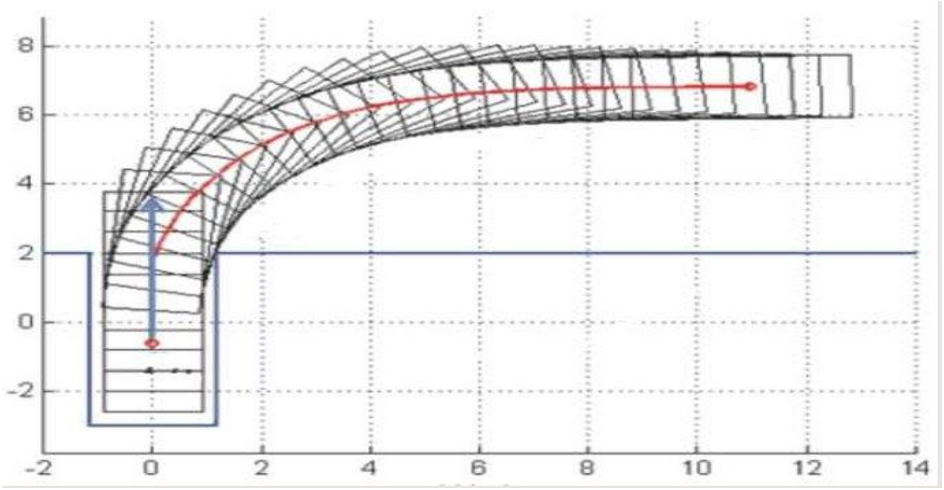
- The horizontal coordinate of start position (S) = Ls
- The minimum parallel distance between the car and the obstacle (d3)
- The vertical coordinate of start position (H) = Hs

We have two phases of movement,

1. From start position Vs to Vbe

2. From Vbe to Vd

With this information, we can create a path plan for our robot, create the environment surrounding it, and know whether these plans are useful and serve their purpose or not.



6. Experimental Methods and Results

6.1 Overview

we will talk about an important part and stage of completing any project, which is testing and approval, as it cannot be said that any product fulfills its purpose unless it is tested and known whether it has defects or is free of them. We will first discuss the ways and methods used for testing and how to synchronize them with the development and analysis of the project, how to test and ensure the readiness of each of the software, components, and the conceptual path for parking the car through special programs for that, and the actual testing of them. The preliminary results and final results will be explained, including how to detect and address errors. In the end, a summary of all the results and analyzes achieved will be presented.

6.2 SW Configuration Implement

The embedded system software for the auto parking car project aims to enhance the parking experience by efficiently managing and controlling multiple systems within the parking facility. Its main tasks include identifying available parking spots and automatically guiding cars to them, thus improving efficiency and saving time and effort for users.

The software program is a set of layers, to connect with the microcontroller and the application.

- Microcontroller Abstraction Layer (MCAL)
- Hardware Abstraction Layer (HAL)
- Real-Time Operating System Layer
- Application Layer

To program the controller and use its peripherals, you must read its Technical Reference Manual (TRM) and all its features, as well as the data sheet for the used board.

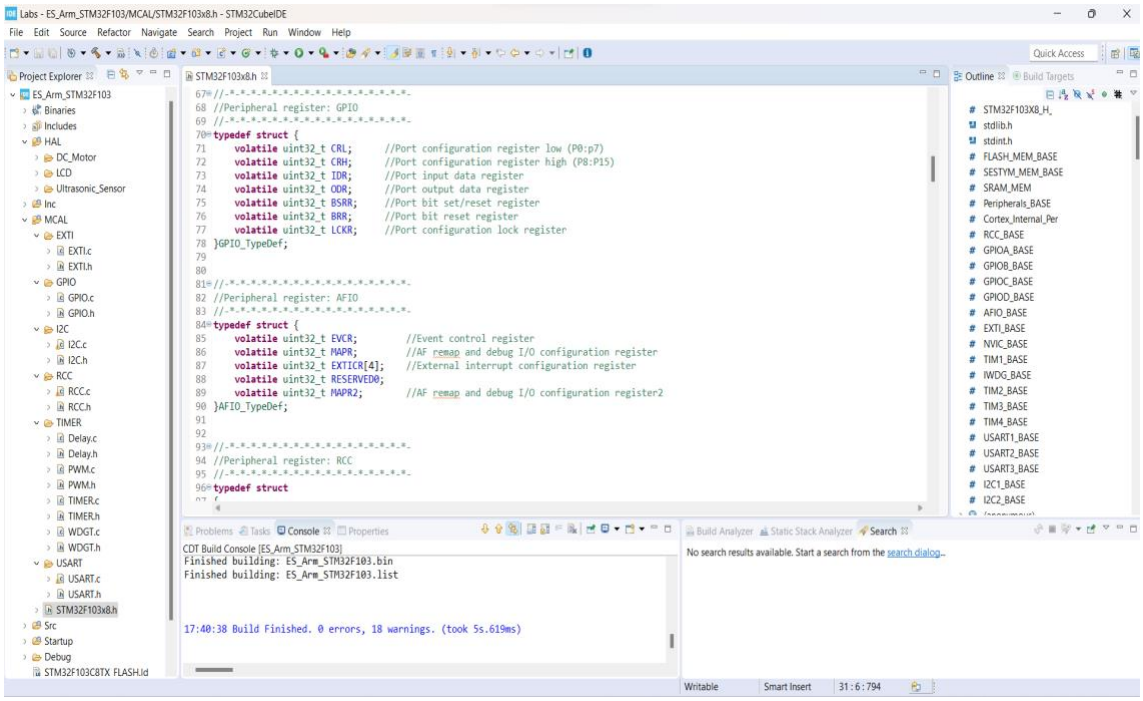
Technical Reference Manual (TRM) shows the following, this provides that how the peripheral work and how to use it:

- Peripheral Introduction.
- Peripheral Function Description.
- Peripheral Registers.

Data sheet of the used board shows the following, this provides that the steps need for configuration the peripheral:

- Block Diagram.
- Clock Tree.
- Memory Mapping.
- Pin Definitions.

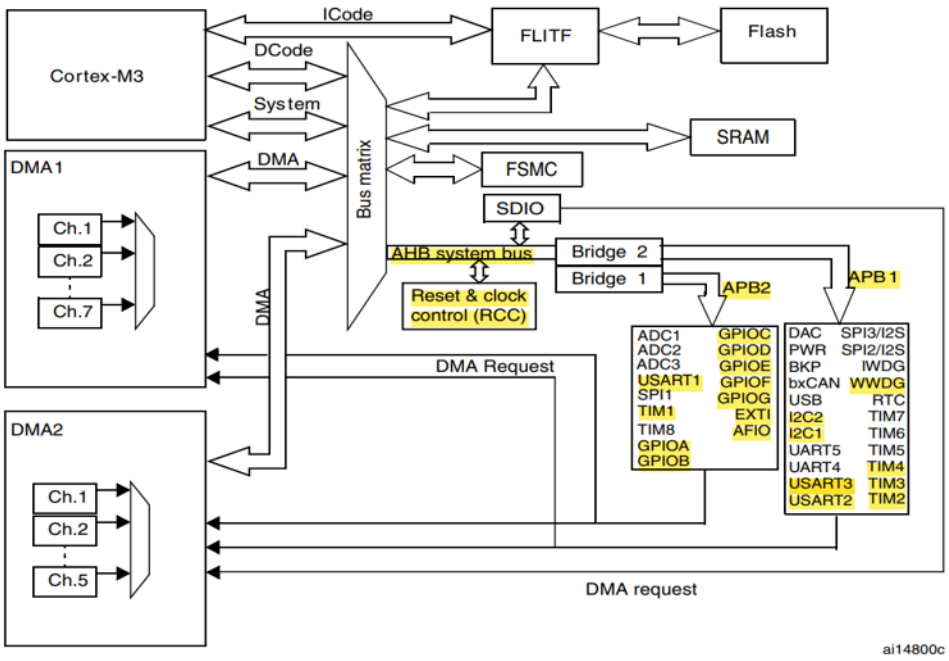
For each peripheral two configuration files. Header file (file.h) that includes peripheral configurations, user definitions, and APIs functions, Source file (file.c) that includes the implementation of APIs functions that meet our needs from the peripheral.



6.2 System Features and Functionalities

Features used by the controller:

- General Purpose Input/Output (GPIO). It is used to configure the controller's pins to interface with sensors and actuators by setting them as either input or output, can be routed to other peripheral to serve some alternate functions.
- Interrupt Service Routen (ISR) and Nested vectored interrupt controller (NVIC).
- Timers and Counters. They used to make action at a specific time, work by counting clk cycles from the system clock.
- Universal Asynchronous Serial Receiver and Transmitter (UART). Is a standard serial communication protocol that can transmit data between devices.
- Inter Integrated Circuit (I2C). Is a serial communication protocol used to connect multiple low speed devices to an MCU.

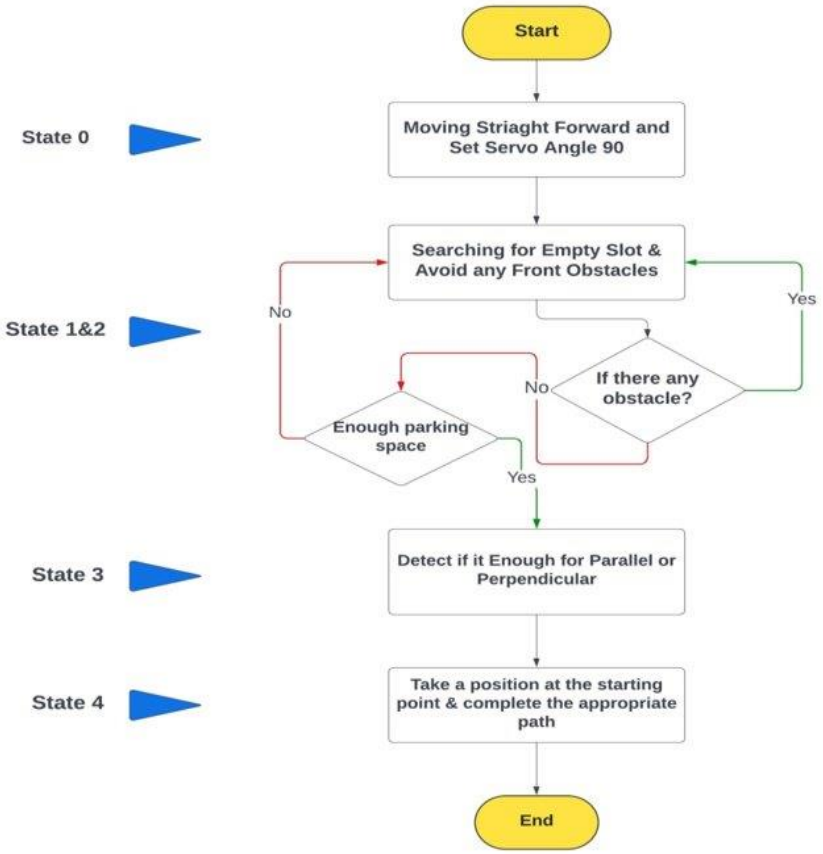


In HAL Layer, the ultrasonic sensor driver is essential for obstacle detection, safety, navigation, parking assistance, and user experience enhancement in the self-parking car project. By integrating ultrasonic sensors into the system and leveraging the driver effectively, the project can achieve reliable obstacle detection, precise navigation, and safe parking maneuvers, ultimately contributing to its success in autonomous parking applications.

RTOS is essential for managing multitasking, scheduling, and resource allocation in the self-parking car project. By leveraging RTOS effectively, the project can achieve real-time responsiveness, modularity, scalability, and reliability, ultimately contributing to its success in autonomous parking applications.

To create an algorithm that matches the project requirements with some other requirements to reach the goal in the best way and most accurate results, there are four states in the general parking procedure. The change of each state depends on the differences in sensor data. Two ultrasonic sensors were mounted on the right side of the robot and the anterior one was

identified as US1 and the posterior one was identified as US2. Any front obstacles are also avoided through the ultrasonic sensor installed in the front of the car UF1. we will assume valid width of parallel or perpendicular width is w.



State 1: In this case, the parking robot detects the environment around it and begins to calculate the length of the first car or wall parallel to it. If any obstacles appear in front of it, the robot will stop moving and activate the alarm. Assuming that the robot moves straight, the robot moves at a constant steering angle.

If (US1 and US2 < w) state=1;

State 2: The robot detects the end of the first car and begins to calculate the length of the parking lot. After that, the robot will make the decision to stop the car or not. There is also a calibration in Case 2 based

on the length of the parking lot. First, the US1 difference will be greater than the width of the parked car. The parking robot was supposed to be on top of the first car. If the difference US1 is less than the width of the parking car again before the difference US2 is greater than the width of the parking car, the robot will decide that there is not enough parking space and go to state 1. If rear sensor US2 is on top of the first vehicle and front sensor US1 is still in the parking space at the time of calibration. The robot will again decide to stop the car or not when the front sensor US1 detects the second car, and at that time, the calibration will stop. Whether or not there is enough parking space can be known by comparing the actual detected length of the parking space and the minimum length of the parking space. If the actual parking length is greater than the minimum parking length, the robot will park the car. If not, the robot will find the next parking spot and go to state 1. During this stage, if any obstacles appear ahead, the robot will stop moving and trigger an alarm.

If (US1 > w && state==1) state=2;

If (US1 > w && US2 > w) Calibrate on;

If (actual parking length < minimum parking length) not parking, state==1;

If (actual parking length > minimum parking length) state3;

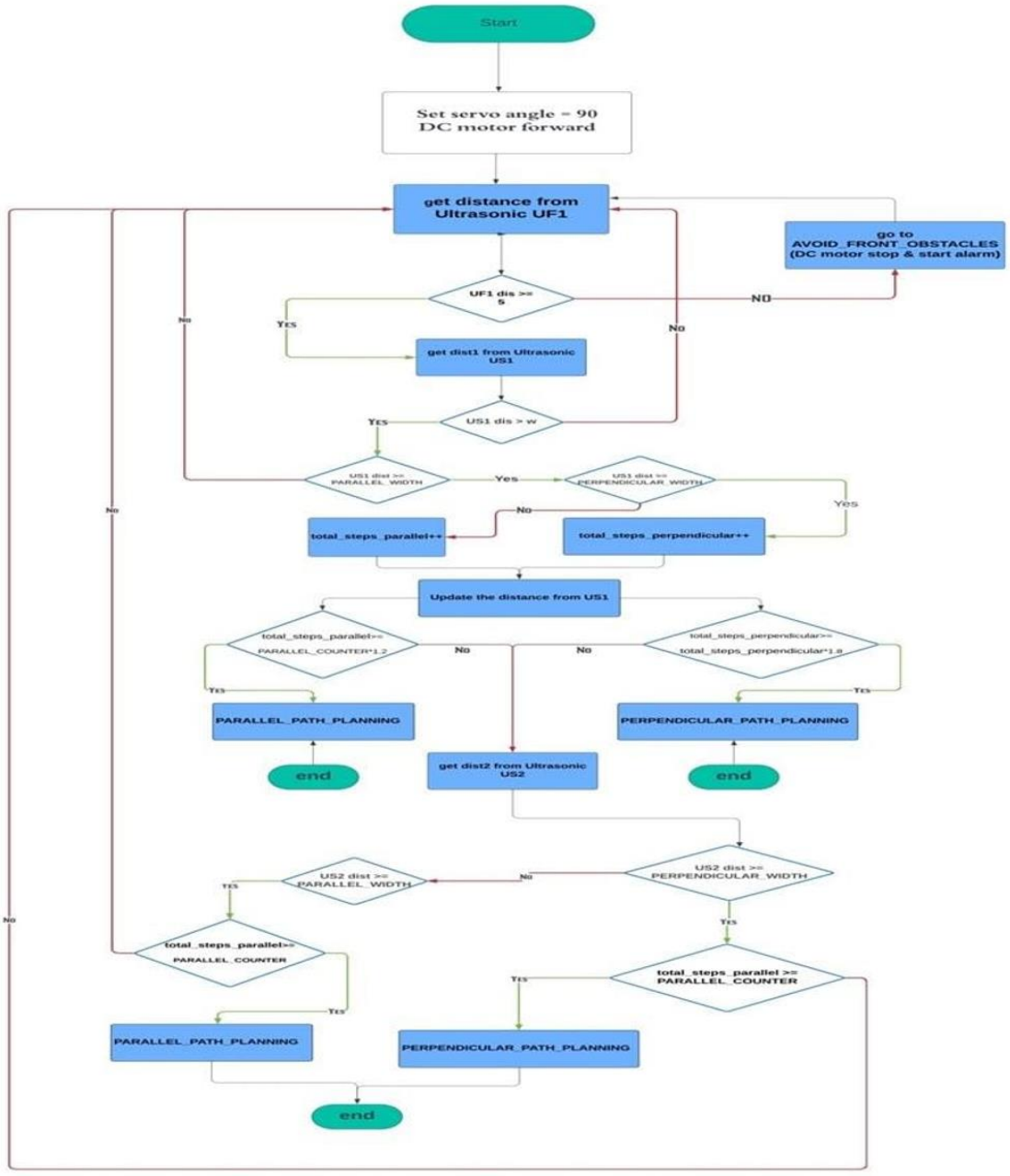
State 3: The car will stop when the rear sensor detects the second car, and the parking system has been introduced in the event that if there is no car or other side obstacle, the car can park (park in the space).

If (US2 difference > w && state==3) state=4;

If (US2 > Parallel width and length) state4(Parallel_Parking);

If (US2 > perpendicular width and length) state4(perpendicular_Parking);

State 4: path shape is formed based on ultrasonic sensor values, is it suit for parallel parking or perpendicular parking



6.3 Discussion of Effectiveness and Limitations

In this section, we will discuss the effectiveness of the application and what limitations we encountered.

The effectiveness of the application is centered on identifying whether the empty area suitable for parking is perpendicular, parallel, or inclined, and

then implementing the appropriate path for the parking, which is what is implemented correctly through the application by comparing the width and length of the empty area through the ultrasonic sensor. In the event that the garage is empty of cars or there are no side obstacles, the application must determine the appropriate area for it, the best path to implement it, and park the robot; which is what has been implemented to avoid the robot continuing to move and its inability to recognize the environment surrounding it. One of the basics of safety is to avoid colliding with any obstacle. The application identifies obstacles that appear in front of the robot, then stops and activates the alarm. During the implementation of the parking path, work was done to create a collision-free path and ensure its effectiveness.

Among the limitations encountered are the limitations of the ultrasonic sensor. This type of sensor has many weak points as it cannot measure a distance of less than 3 cm and can only measure the distance of only one point at a time, and its efficiency is not very good. So, when we started working on mapping, we were trying to do full autonomous navigation to have the CPU build a complete picture of the environment but while studying this method we found that the data we had from the sensors was not enough to use this complex method. But in this method, we can use LIDAR or camera to provide enough information for this algorithm. It is also not possible to use the sensor located on the back side of the car because while parking the car, the car will be very close to the objects surrounding it, especially the back side, which makes the sensor give random values, which may spoil the parking process. When we try to implement the parking method (corner parking), the sensors cannot give enough information to determine whether there is a suitable place or not because there is always a blind spot and this may lead to damage to the car while parking and for this reason we dispense with the use of this one.

Conclusion

This paper explores the realm of self-parking automobiles, which assist drivers in locating and parking spaces through embedded systems technology. The devices are designed to lessen driver fatigue, increase parking efficiency, and lower the chance of collisions. We created car design, parts, and algorithms. To reach the true experience of self-parking and come closer to reality, we used actual automobile measurements. We used the SOLIDWORKS program to design every part, and 3D printing was used to make every item. The project needed thorough component evaluation and selection based on speed, accuracy, cost, interface compatibility, and other considerations. In order to accomplish the goal in constrained settings, the research used a low-speed approach with non-holonomic limitations for vehicles to accomplish the aim in confined locations. MATLAB was used for system modeling and verification. The dimensions of the parking area and the automobile itself determined the length and width of the parking environment, which was built using sensor data. Equations were used to characterize the parking path, and the car's width, wheelbase, turning angle, and clearance requirements were taken into consideration for determining the minimum size of the parking space. We are developing a software system that uses a servo motor, DC motor, and ultrasonic sensor to identify an empty parking space.

References

[1] Zhenji Lv, Linhui Zhao, Zhiyuan Liu, “A path-planning algorithm for parallel automatic parking”, Third International Conference on Instrumentation, Measurement, Computer, Communication and Control 2013

[2] Dainis Berjova, “RESEARCH IN KINEMATICS OF TURN FOR VEHICLES AND SEMITRAILERS”, Latvia University of Agriculture, Faculty of Engineering 2008

[3] American International Journal of Sciences and Engineering Research, “Autonomous 4WD Smart Car Parallel Self-Parking System by Using Fuzzy Logic Controller”, American Center of Science and Education 2019

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

[5] 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.

[6] 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.

[7] 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.

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

[9] Soe Yu Maung Maunga , Yin Yin Ayeb , Nu Nu Winc, “Autonomous Parallel Parking of a Car-Like Mobile Robot with Geometric Path Planning”, American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)

[10] STM32F10Xxx Technical Reference Manual

[11] STM32F103c8 Datasheet

[12] HCSR04(Ultrasonic sensor) Datasheet

[13] User guide L298N Dual H-Bridge Motor Driver

[14]https://www.embitel.com/blog/embedded-blog/what-is-autosar-mcal-software-architecture