July 2015 IEEE

Ackerman's Steering based Nonholonomic Parallel Parking System (N.P.P.S.)

Matthew Pragasam#1, Mr. Ooi Woi Seng*2

School of Computing, Engineering and Information Sciences,
KDU College Penang,
Anson Road, Georgetown, Penang, Malaysia- 10400
[1]contact@matthewpragasam.com
[2]ooiws14@gmail.com

Abstract— **Parallel parking presents one of the areas, which challenges a driver's capabilities in modern day driving. The area of an autonomous parking system attracts a great deal of attention from the research community and is seen to be one of the first steps to developing a fully automated vehicle. The technological and mathematically complexity of the task had inspired me to take on the approach to develop an accurate and consistent parking system. The system consist of firstly identifying, computing and finally executing the parking procedure which mimics biological solutions. These stages requires an internal computer generated map which grants greater accuracy over the conventional method in detecting a suitable parking bay. The degree of uncertainty over determining a parallel parking space are based on driving experience and varies from one person to the other. The information acquired from mapping the environment i.e., a suitable parking bay were used to generate a path for the model to perform its maneuver. It was found that the system is ideally suitable to be implemented realistically with the need of additional mechanical assist in identifying certain vehicle constraints, i.e. length of axle's track width. The feasibility of the prototype is measured in terms of performance, reliability and accuracy, which produced consistent results.**

Keywords— **Ackerman's steering, parallel parking system, Nonholonomic model, autonomous navigation, path-planning, parking trajectory algorithm**

I. INTRODUCTION

The first step to solving problem is recognizing there is one. Parallel parking is a method where a vehicle is aligned parallel to the road and performs a maneuver into an available parking space to the right or left of a designated area. However, parallel parking requires experiences and much practice to achieve and is one of the areas where the automotive industry still yet need to focus. Consequently, the duration for a parallel parking is proportional to the number of maneuvers the driver has to take. Therefore, the more number of maneuvers taken to complete a parallel park affects the overall duration of a parallel parking. To perform a parallel parking maneuver, the driver has to first identify a suitable parking bay proportional to the length of his or her vehicle. This identification of a suitable parking space is presently possible only by an estimate. The parallel parking maneuver requires motion planning and a specific set of controls to accurately position the vehicle consistently within the constraints of a parking bay. These motion-planning algorithms are derived from the implementation of Ackerman's Steering and Pythagoras theorem, circle identities and trigonometric functions. Various methods of control were used for the purpose of this research, namely the Bang-Bang and Time-Variant control.

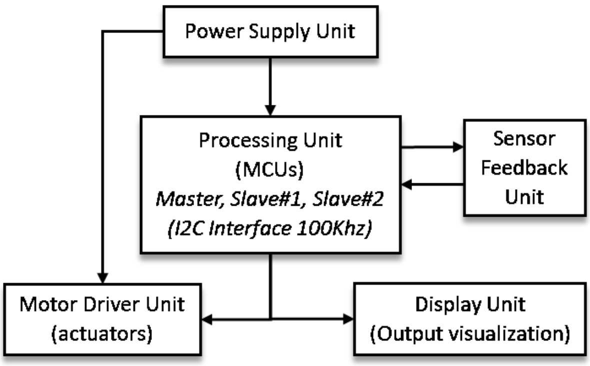
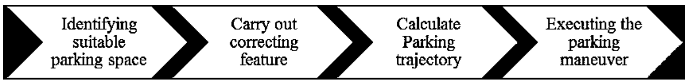


Figure 1: Block Diagram of N.P.P.S.

The design approach in carrying out a successful parallel parking procedure requires the following process shown in Figure 2.



II. PARKING BAY DETECTION

The parking bay detection system is a pre-parking procedure, where the model will search for a suitable parking bay to the right or left of the model. The system identifies a suitable parking bay according to the input dimensions of the vehicle. Proposed by Simon Blackburn [1], the minimum parking space required can be identified by acquiring the vehicle's wheelbase, track width and minimum turning radius. The model will then scan for a suitable parking bay by navigating forward maintaining constant velocity.

The method of detecting a suitable paring bay consist of using a Ultrasonic distance measuring sensor and an internal timer interrupt(timer0) of the Atmega328PU. The distance measuring sensor was used to identify the depth and the timer was used to identify the length of the parking bay. Since the model is travelling at a fixed velocity, the bay length can be identified from the time taken to travel from start of gap to end of gap. The information acquired from detecting the parking bay is used to generate an internal map and proceeded to producing a parking trajectory. The parking bay detection is also subjected to Blackburn's geometrical constraints [1].

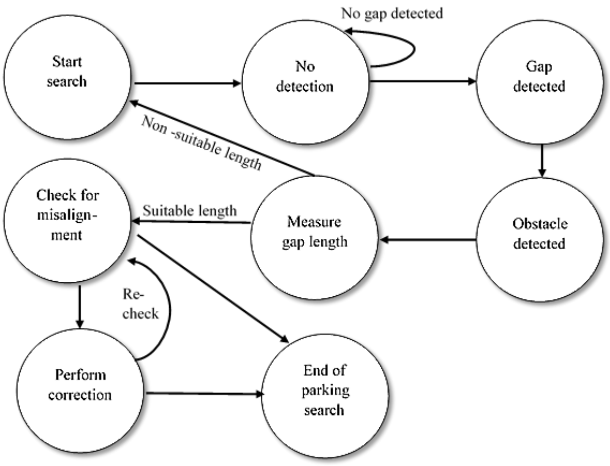


Figure 2: State diagram of parking bay search.

A. *Localisation (Map-learning)*

Proposed by Meyer and Filliat [2], the map-learning method for robotic navigation is a significant part of acquiring the position of the model in reference to its environment when no prior knowledge about its surrounding is given.

The model constructs a two-dimensional internal map of its surrounding by retrieving *x* and *y* values. This can be referred as to the length and depth its surrounding. However, the model has to travel pass it's surrounding for it to be able to collect a plot of its environment.

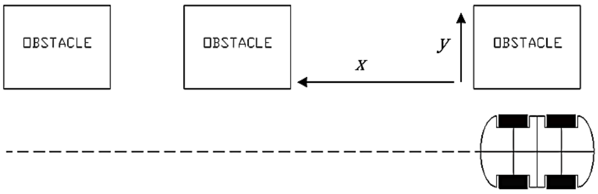


Figure 3: Environment of model.

Because the *y* value returns an infinite number limited by the distance sensor's capability, a filtered mapping response (red plot) was constructed when depth values surpassing minimum parking depth condition to filter out insignificant depth measurements.

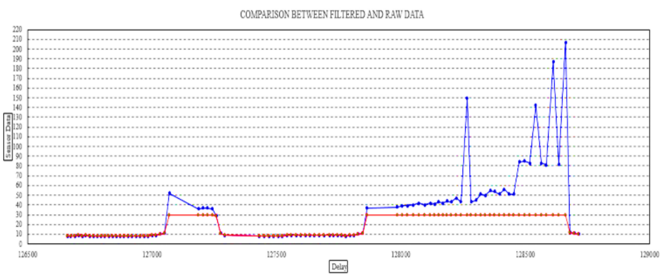


Figure 4: Example of environmental plot of model's environment based on Figure 3.

B. *Cyclical and Linear Time Variant System*

The parking bay detection system is developed upon cyclical and linear time variant control. Linear time is in respect to Newtonian theory of sequential time where a series of events that are leading to towards and end. On the contrary, circular time sees time as a circular and does not necessarily lead towards something but instead it repeats itself within a cycle of events [3].

As an example, a moment in circular time can represent any state but in linear time one state leads to another in sequential order. This means that circular time can be used to reset and linear time is used as a measurement. The circular and linear time-variant system is mainly implemented in the use of determining a suitable parking bay, where circular time is initiated when the gap detected is non-suitable measured by linear time. The parking bay detection system is time-dependant and thus known as a time-variant system.

III. Ackerman's Steering Parking Algorithm

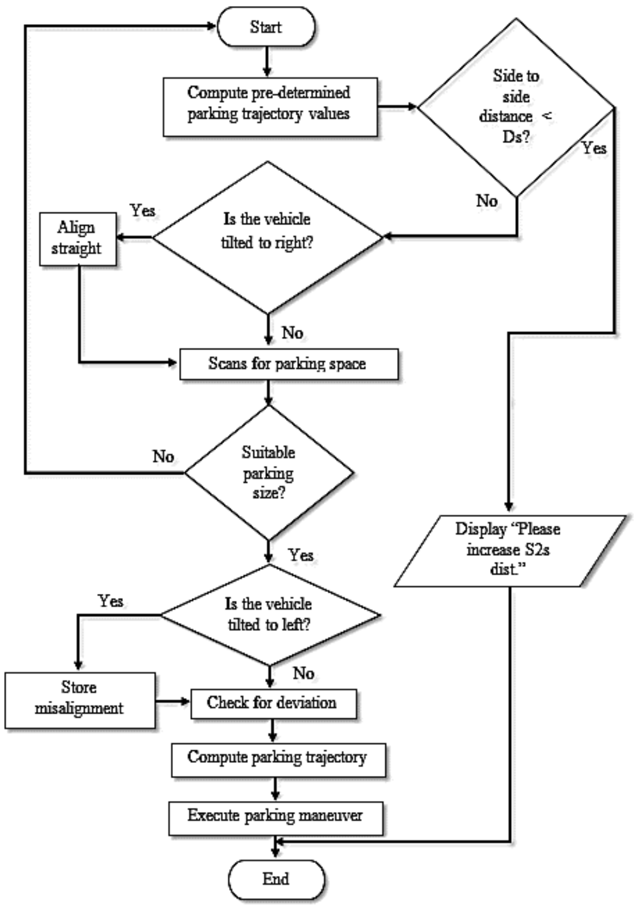


Figure 5: Overall Ackerman's Steering based Parallel Parking System.

The parking trajectory consist of two turning circles with only one steering change. The parking trajectory can be seen in Figure 6.

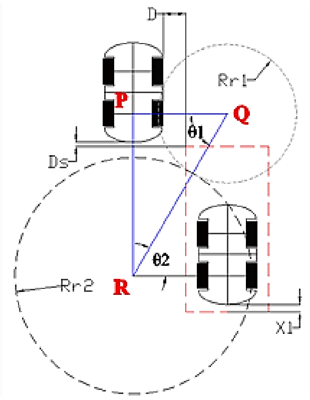


Figure 6: Computed parking trajectory.

The angle for which the front inner and outer wheels need to steer to turn along a turning radius is calculated by using Ackerman's Steering Theorem [4] shown in equation (1) and (2).

$$\theta i = \arctan\left[\frac{Wheelbase}{Min.\ Turning\ Radius - \left(\frac{Trackwidth}{2}\right)}\right] \tag{1}$$

$$\theta o = \arctan\left[\frac{Wheelbase}{Min.\ Turning\ Radius + \left(\frac{Trackwidth}{2}\right)}\right] \tag{2}$$

The model needs to first reverse the distance of D_s which is identified by,

$$D_s = PR - L_{tr} - \left[L_p - (L_{tr} + X_1)\right] \tag{3}$$

Where, L_{tr}, L_p and X_1 are the front track width to front edge length, length of parking space and distance between rear of vehicle to front of parked vehicle respectively.

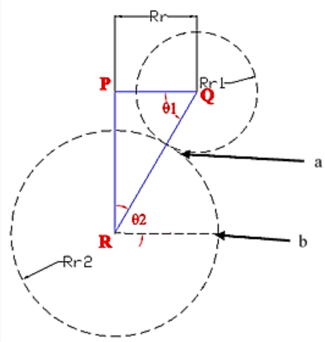


Figure 7: Trajectory of rear right wheel.

The turning radius of R_{r1} and R_{r2} is the turning radius of the rear right wheel which is defined as:

$$R_{r1} = R_r - \frac{W_{track}}{2} \tag{4}$$

$$R_{r2} = R_r + \frac{W_{track}}{2} \tag{5}$$

Where W_{track} is the vehicle's track width. This calculation enables the tires to intercept at point 'a' where the steering angle would change from maximum right steering to maximum left steering. The distance which the vehicle is required to travel along the both turning radius is determined by the angle of θ_1 and θ_2 . The angle of θ_1 and θ_2 are of equivalent values and is identified as follow:

$$\theta_1 = \cos^{-1}\left(\frac{PQ}{QR}\right) \tag{6}$$

The model when then finally rest at point 'b' where the rear right tire is positioned.

A. *Deviation Correction*

When a vehicle is deviated, it means that the side to side distance of the vehicle from the parked vehicles are above or below ideal distance. This will most likely happen in reality as the driver is not able to achieve the precise side to side distance. Therefore to solve this, Dx is considered to be the minimum side to side distance which the vehicle needs to maintain. As an example, if the vehicle is deviated to the right (<Dx closer to parked vehicles), the system will prompt the driver indicating that the required side to side distance is too narrow. Therefore the driver will have to deviate to the left which will increase the side to side distance; however if the vehicle is deviated to the left with side to side distance being larger than Dx, then the deviation correction algorithm will take place.

$$R_{r1new} = R_r - \frac{W_{track}}{2} + \Delta D \tag{7}$$

$$R_{r2new} = R_r + \frac{W_{track}}{2} + \Delta D \tag{8}$$

Where, ΔD is the deviated distance. The new turning radius will then replace the ideal turning radius of equation (4) and (5).

B. *Misalignment Correction*

Misalignment occurs when the vehicle is not positioned parallel to the parking space or vehicles parked at the side. Misalignment can be categorized into two directions where the vehicle can either be misaligned to the right or left direction as shown in Figure 8 below.

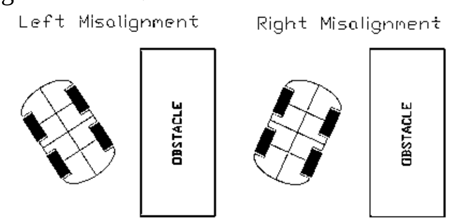


Figure 8: Misalignment direction.

If any forward maneuver assigned to a vehicle which is misaligned to the right; the vehicle may collide with the obstacle infront. Therefore to obey the no-collision constraint, a misalignment check at the initiation of the system is mandatory. If the vehicle is found misaligned to the left, it is known that the vehicle is not going to collide with parked vehicles. However, the system will proceed to measure the side to side distance to ensure that the vehicle is not deviated above the maximum allowable distance.

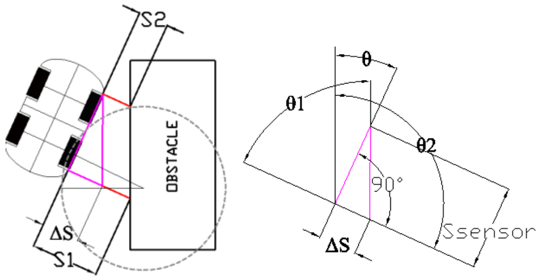


Figure 9: Geometry of right misalignment.

Figure 9 above shows a vehicle misaligned to the right with sensors S2 and S1 attached to the front and rear right side of the model. Using the formula below, the misalignment direction of the vehicle can be determined by finding ΔS.

$$\Delta S = S2 - S1 \tag{9}$$
$$if, \Delta S > 0 \ (vehicle\ misaligned\ to\ left) \tag{10}$$
$$if, \Delta S < 0 \ (vehicle\ misaligned\ to\ right) \tag{11}$$

The misaligned angle, θ of the vehicle can be identified as follow:

$$\theta_m = \left[180 - \left[\tan^{-1}\left(\frac{S_{sensor}}{\Delta S}\right)\right]\right] - 90° \tag{12}$$

Where, S_{sensor} is the distance between the front and rear right side sensor. Therefore, the vehicle will turn left with an angle of θ_m.

Apart from that, when a vehicle is misaligned to the left instead, the misalignment angle will be incorporated into the parking maneuver. The misalignment is added to the second turning angle θ_2 in equation (6) represented by 'β' in Figure 10.

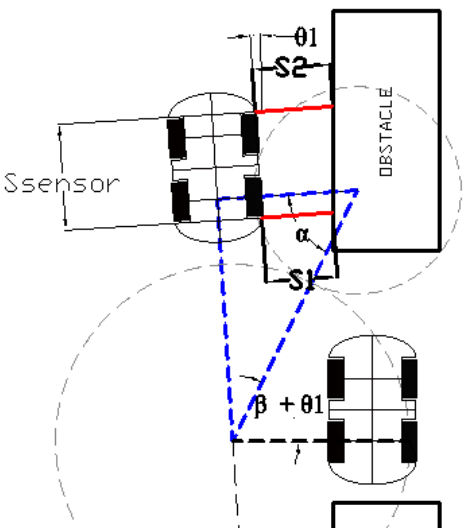


Figure 10: Incorporation of misalignment angle into parking trajectory.

The misalignment angle (θ_1) is integrated into the parking trajectory of the second turning radius. The total distance which the vehicle needs to travel across an arc angle of $\beta + \theta_1$ is required in the second turning circle to account for the misaligned angle; β is equivalent to α. The misalignment angle of θ_1 can be identified from the equation (13):

$$\theta_1 = \tan^{-1}\left(\frac{\Delta S}{S_{sensor}}\right) \tag{13}$$

Apart from that, the parking trajectory has to also be based upon if the vehicle is deviated. Therefore for a deviated and misaligned vehicle, the side to side distance (S_{2s}) has to also be identified and compared with the minimum required side to side distance to determine the deviated distance of ΔD. The methodology is specified in equation (14) by using parallelogram and trigonometric identities:

$$S_{2s} = \cos[90° - (90° - \theta_1)] \times S_1 \tag{14}$$

Where, S_1 is the sensor value of rear right side sensor.

IV. Nonholonomic Parking Algorithm based on Ackerman's Steering

The nonholonomic parking trajectory is based upon the parking trajectory developed in respect to Ackerman's Steering Theorem. This is because the model based car-like robot is of differential wheeled configurations and is limited due to *kinematics constraint* [5]. However, the parking bay detection system is of similar methodology and functions in the same way of map-learning.

A. Nonholonomic Parking Trajectory

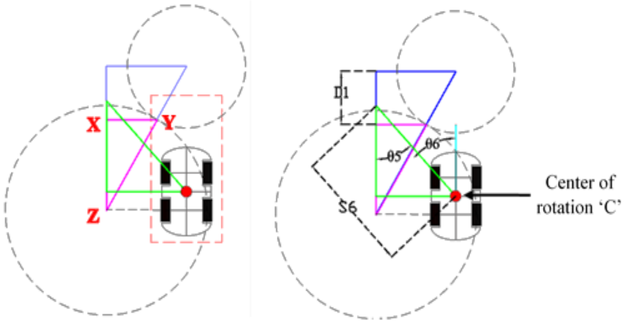


Figure 11: Nonholonomic parallel parking trajectory based on Ackerman's Steering.

Since the differential-wheeled model is limited to kinematic constraints. The model is not able to steer like a realistic vehicle. However, it is able to rotate to right and left with center of rotation known as 'C' shown in Figure 11. The distance of D_1 is the length which the model needs to reverse so that it can rotate the angle of θ_5 into the parking space with center point 'C' and length of S_6. To identify the reverse distance of D_1, the length 'XZ' of the purple right angle triangle is determined by equation (15).

$$XZ = cos[(90° - 60°)] \times \left(R_r + \frac{W_{track}}{2} + \Delta D\right) \tag{15}$$

Where R_r the adapted turning radius of the Ackerman's Steering based design. From retrieving the value of 'XZ', D_1 can be determined by $PR - XZ$. At this point the vehicle needs to rotate to left with an angle of θ_5 which is calculated as follow:

$$\theta_5 = \tan^{-1}\left(\frac{R_r + \Delta D}{XZ}\right) \tag{16}$$

The model will then need to reverse the distance of S_6, calculated as follow:

$$S_6 = \frac{XZ}{\cos\theta_5} \tag{17}$$

However at this stage, the model is not aligned straight. The model has to rotate right with an angle of θ_6 which is of the same value as θ_5.

V. Electronic And Control Approach

The programming is done in such that each stage is considered as a state machine. This method is used throughout the development of *NPPS*.

A. *Differential-Wheeled Design and Ultrasonic sensor placement.*

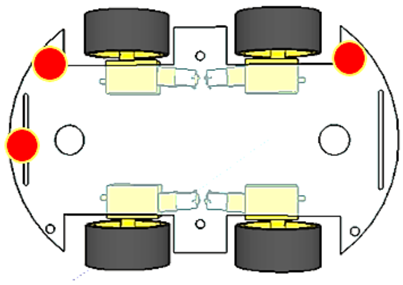


Figure 12: Differential-wheeled model and placement of sensors.

The differential wheeled design shown in Figure 12 enables the model to rotate instead of steer towards a right or left direction. This is because the model is incapable of turning right or left without accounting for tire slippage. Therefore, the best solution for the model to accurately maneuver across a plane is to rotate about the center point. To be able to rotate right or left, either right or left side wheels has to rotate in the opposite direction to enable the model to rotate.

Besides that, the ultrasonic sensor placement is extremely important in determining proper distance measurements. The ultrasonic sensors were placed marked in red dots shown in Figure 12. The front sensor is used to detect obstructing obstacles at the front of the vehicle whereas the front right and rear right sensor is used to check for misalignment. In addition, the rear right side sensor is also used to carry out map-learning. All the ultrasonic sensors are placed vertically as shown in Figure 13.

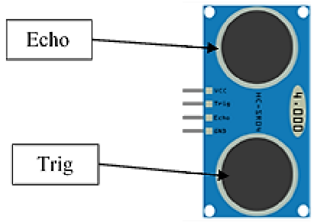


Figure 13: Ultrasonic sensor (HC-SR04) placed vertically.

The ultrasonic sensor is placed vertically to improve distance measurements. This is because having the sensors placed vertically prevents loss of trigger or echo signal as compared to sensors placed horizontally. Ultrasonic sensors heavily rely on receiving echo waves hence, it may result in delay when the model is positioned at an edge of a gap.

B. *Magnetometer Angle measurements*

The angle measurements are made possible by using the magnetometer sensor named HMC5883L by Honeywell. If the model is required to rotate an angle of left 45°, the system will have to go through the following steps to achieve its angular position using definition of quadrant in a circle shown in Figure 14:

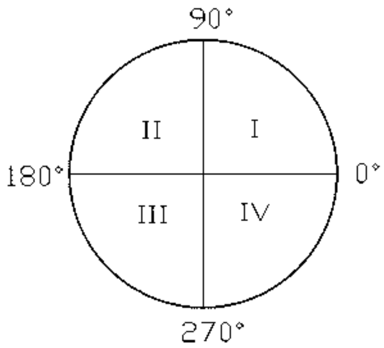


Figure 14: Circle separated in quadrants.

Step 1: Knowing that the left direction gives angular increment values; the system then adds the current heading value with the required rotating angle.

Step 2: Check if the heading value would exceed 360 degrees by using subtraction method. This means that if the total heading value is more than 360, the rotational angle would fall into quadrant I starting from 0 degrees.

Step 3: If the heading value is identified to be more than 360 degrees, then the system would first rotate the model to reach value $\geqslant 0^\circ$ entering quadrant I. This is done so that the system can distinctively differentiate between two conditions.

Step 4: At this stage, the system would then rotate until it reaches its intended position.

C. *Bang-bang control*

The method of control used in the angular positioning is by using the concept of bang-bang control theory. This control is suitable for the system in providing controllable degree of cycles to achieve desired angular position. This means that each step of the model in terms of rotation is controlled by supplied power to the actuators in on and off sequence. Each step of switching 'on and off' by supplying power to the actuators gives a degree of angular maneuver which can be determined by the duration when the actuators are switched on. As an example, each cycle (on then off) gives an angle maneuver of 5° change, therefore to reach 45° the model has to go through atleast 9 cycles to reach it's intended position.

However if in non-ideal cases, it is wise to use bang-bang control for correcting its value more precisely. If the rotational angle is suddenly larger than 45° due to slippage or external factors; bang-bang control can also be used to precisely position the model back to 45° only by changing the polarity control to the DC motors.

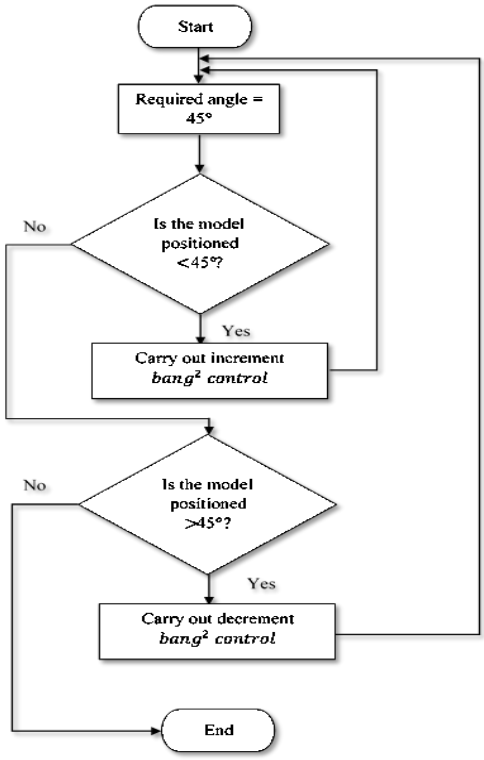


Figure 15: Flow chart for Bang-bang control.

VI. Experiment and Simulation Results

The experiment constructed mainly identifies and discusses the reliability of the parking bay detection and reliability of the parking procedure.

A. Reliability of Parking Bay Detection System

This experiment studies the pattern and response of the system's identification of a suitable parking bay. Since the parking bay detection system is of time-variant control, the method for conducting the experiment is to relate the rear right side ultrasonic sensor value with time. Considering the fact that the model has to be mobile while data collection are retrieved, information are instead collected through Bluetooth serial communication which enables wireless transmission and control over the entire system. Sensor data are sent to slave device2 (system debug MCU) through I2C interface to be stamped with current time and sent through serial communication.

To be able to accurately identify the dimensions of the parking bay, the system has to collect sufficient samples to construct an internal-map as such in Figure 4. By increasing the PWM speed, less samples are collected within a given parking bay area.

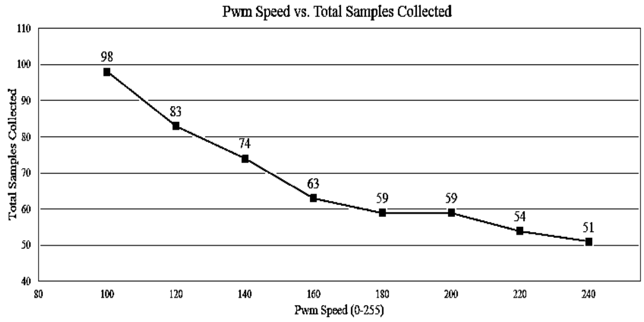


Figure 16: PWM speed vs Total samples collected.

Form the observation, the lower the sample count, the lower the resolution of the mapping system. The samples collected at maximum PWM configuration reduced to almost half of its number as compared to PWM speed of 100.

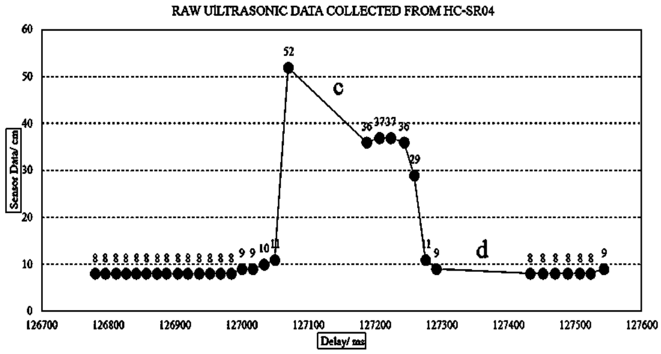


Figure 17: Conditional delay labelled 'c' and 'd'.

Besides that, conditional delay labelled '*c*' and '*d*' were identified as the system requires such delay to check through the conditions to start and stop timer0 interrupt. The results shows that conditional delay 'c' and 'd' requires an average of 4.8cm distance before another sample can be collected. This means that to be able to accurately map out an entire parking bay, the gap length and gap to gap distance must be atleast 4.8cm long to maintain experimental consistency.

B. Reliability of Parking Bay Detection System

In the parking reliability test, the system's constraints as well as the duration required for parking are put to test. This will produce the reliability results of the system in accordance to the objectives of this study.

To be able to identify the range of constraints, the system has to first identify the width of the parking space. The width of the parking space will affect these sets of constraints. The width of the parking space is manipulated and the allowable range of side to side distance is shown in Figure 18 below.

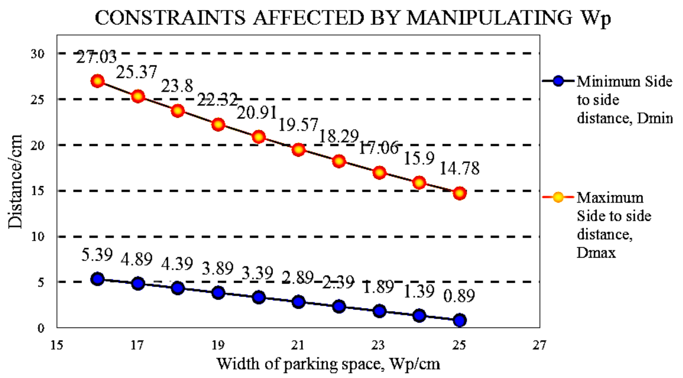


Figure 18: Side to side distance range affected by parking width.

The results shows that when the parking width is narrower, the model's minimum and maximum side to side distance increases. This is because the algorithm designed only utilizes two turning maneuvers to reach its final resting position. Hence, the turning radius is used as reference to place the vehicle at its position. Therefore if the parking width is larger, the model has to move closer in parallel to the parking spot to perform a successful parallel parking. However, if the system is given a fixed parking space size with a bay width of 25cm, the range of constraint which the model has to be placed side to side is at minimum of 0.89cm to 14.78cm apart. The maximum side to side distance is sufficient enough because comparing the range to the model's width, the maximum side to side distance is almost the length of the model's width.

Apart from that, the duration of which the model requires to Parallel Park is also being discussed. In this experiment, the model is required to perform the parking maneuver which consist of reversing and rotating. These maneuvers will contribute to the total duration required to park the model successfully. The focus of this study is to identify the factors that affect the difference in parking duration. The speed of the motors were kept at PWM100 for reversing motion and PWM255 for bang-bang control cycles in rotating the model.

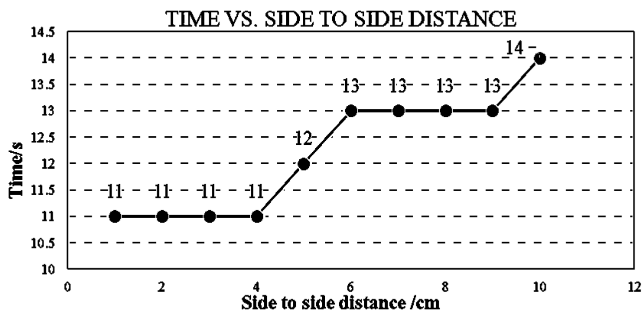


Figure 19: Time vs. Side to side distance.

From the results obtained, it is clearly seen that manipulating the side to side distance of the model will affect the change in parking duration. It can be observed that when the side to side distance increases, the parking duration increases as well. However the change in parking duration stayed constant until it reaches 5cm of side to side distance. This can be explained by small delay increments of milliseconds throughout side to side distance of 1 to 4cm. This data reading can also be observed from 6cm to 9cm where the model is placed. The longest duration taken by the model to complete a parallel parking is 14 seconds with a side to side distance of 10cm. Although the graph does not show a linear graph, but the increment of side to side distance is directly proportional to the time taken to Parallel Park. This can be explained as the model requires a longer distance of S_6 when the side to side distance increases shown in Figure 20.

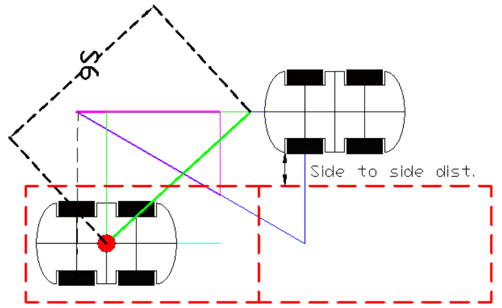


Figure 20: Depicting length of S6.

The parking duration can be further reduced by increasing the speed of the motors during maneuver. However, this will have a total separate effect on the positioning of the model as the DC brushed motors do not have features to brake. Therefore, PWM speed of 100 was used and is sufficient to carry the model back and forth with minimal momentum.

VII. Conclusions

The completion of this research and the development of the nonholonomic parallel parking model has contributed to many surprising findings and greater knowledge learnt. From the experiments conducted, results shown presents a satisfactory solution to the parallel parking problem as opposed by many motorist. The parking maneuvers developed and performed by the model is reliable, sufficiently fast, accurate, reliable and consistent. The parking procedure involves minimal maneuvers, thus producing better results in overall duration of performing the parallel parking. System constraints were clearly stated and tested in par with the algorithm developed for the parallel parking procedure. The kinematic constraints on the nonholonomic robot were taken into consideration, and a mimic system was successfully implemented to accommodate its maneuvering constraints.

The main contribution to this research study are the implementation of identifying, positioning and maneuvering which helped to evaluate and simplify the problems faced when developing an autonomous parking model. These stages are practical solving skills when a driver is faced with performing a parallel parking; Hence, being adapted into the basics of developing each stage to autonomously carried out on a car-like robot. The strategy to carry out a successful parallel parking procedure is made possible by constructing the map of the goal. This map is then fitted with parking algorithms developed specifically for that task which is ideally collision-free. This is because, the algorithms are derived from the information retrieved from mapping the environment, which in this case is the parking bay with all constraints and measurements. These methods are inspired by biometric robotic research which helps in carrying out specific tasks autonomously.

Mapping the environment of the model is one of the most important stage throughout the research. Although the technology and sensors used in NPPS are very rudimentary, the results obtained still does provide adequate information and necessary accuracy of its environment sufficient for it to proceed with a collision-free path planning. In fact, only one ultrasonic sensor were used in mapping its environment which helped to minimize cost and still maintaining sufficient data collection. Apart from that, the largest contribution of this research is its distinctive difference of two separate complex program where one is to autonomously find a suitable parking bay and the other is to autonomously perform parallel parking maneuvers. The system is by no means recommending the substitution of a physical driver but instead incorporates its autonomous capabilities with the driver itself. This is because the system is not capable of handling unsuspected traffic or to autonomously drive on a street full of traffic. The significance of this system can be enhanced when allowing a human driver to drive the vehicle manually until a parallel parking space along the street is required and therefore initiating the system. Challenges such as traffic control are being handled manually by the driver and the task of parallel parking is assigned to the system performing what it does best.

Future Work

Improvements on real time sensor reading and capabilities of sensors can help in debugging and identifying maneuvering issues. Besides that, coupling the use of different measuring sensors such as ultrasonic and infrared sensors can improve distance measurements by comparing both sensor data. The use of incremental rotary encoders can help in accurately determining distance travelled as well as travelling velocity.

Acknowledgment

I would like to express my deepest gratitude to my academic and research supervisor Mr. Ooi Woi Seng for his guidance and constant support in helping me to conduct and complete this work during and after working hours. I would also like to thank Ms. Yap Suk Han for her ever-ready support in consulting me throughout completing my final year project. In addition, I want to thank Mr. Koay Fong Thai and my peer Mr. Ong Yuen Chiek in providing me the necessary assist and insights while conducting my research.

I am grateful for the facilities and workspace provided by KDU College Penang, which had enabled me to carry out most of my experimental work there. Many thanks to all the people I have come to know during my studies in KDU College, whose friendship I cherish and experiences I will always remember. I owe my sincere appreciation to my family who have provided me constant emotional and financial support and had encouraged me throughout my years of studies. I want to especially thank KDU College Penang for offering me this opportunity to study the programme of my greatest interest and a scholarship to reduce my financial burden. Finally, I would like to extend my profound appreciation to my beloved parents for their love, affection and invaluable support throughout many late nights completing my work.

References

[1] Blackburn, Simon R. 'The Geometry of Perfect Parking'. (2009).

[2] Meyer, Jean-Arcady, and David Filliat. 'Map-Based Navigation In Mobile Robots:' Cognitive Systems Research 4.4 (2003).

[3] Sundaram, Dr. Mark. 'Linear And Cyclical Time: Time's Arrow Or Boomerang?'. *The Endless Knot*. N.p., 2013. Web. 12 July 2015.

[4] RcTek,. 'Ackerman's Steering Principle'. N.p., 2015. Web. 22 June 2015.

[5] Murray R. & Sastry S. (1993). Nonholonomic motion planning: steering using sinusoids. IEEE Transactions on Automatic Control, 5, 700–716.

[6] My.element14.com,. 'Incremental Encoders | Element14 Malaysia'. N.p., 2015. Web. 3 July 2015.

[7] Storr, Wayne. 'Pulse Width Modulation Used For Motor Control'. *Basic Electronics Tutorials*. N.p., 2013. Web. 7 July 2015.

[8] Laumond, J., Jacobs, P., Taix M. & Murray, R. (1994). A motion planner for nonholonomic mobile robots. IEEE Transactions on Robotics and Automation, 5, 577–593.

[9] Lo, Y., Rad, A., Wong, C. & Ho, M. (2003). Automatic Parallel Parking. Proceedings of the 2003 IEEE Intelligent Transportation Systems, 2, 1190–1193.

[10] Jarzebowska, Elzbieta. *Model-Based Tracking Control Of Nonlinear Systems*. Boca Raton, FL: CRC Press, 2012. Print.

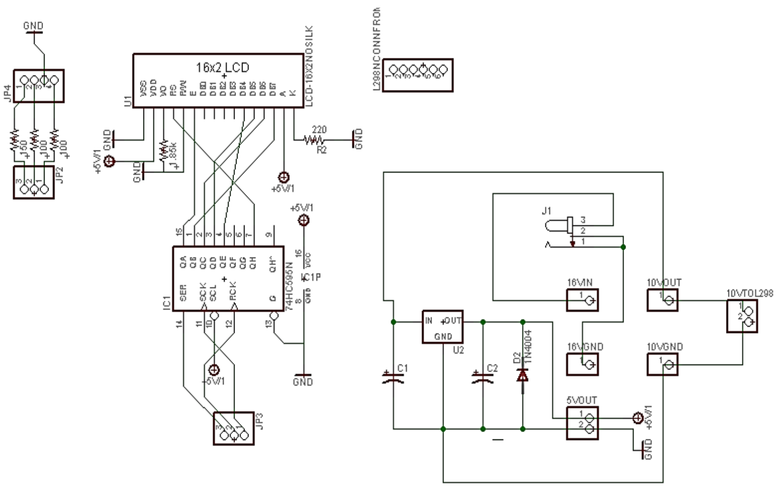
[11] Jernström, Johan. 'AUTONOMOUS PARALLEL PARKING'. (2015).

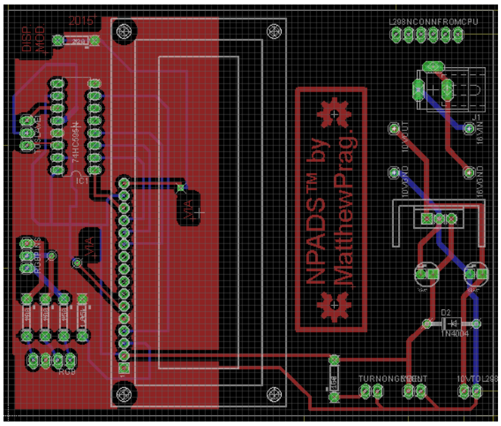
[12] Von der Hardt, Hans-J. et al. 'A Method Of Mobile Robot Localisation By Fusion Of Odometric And Magnetometric Data'. *The International Journal of Advanced Manufacturing Technology* 9.1 (1994): 65-69.

[13] Mallot, Hanspeter A, and Sabine Gillner. 'Route Navigating Without Place Recognition: What Is Recognised In Recognition-Triggered Responses?'. *Perception* 29.1 (2000).

APPENDIX I: SCHEMATIC CIRCUITS OF 3 BOARD LAYERS

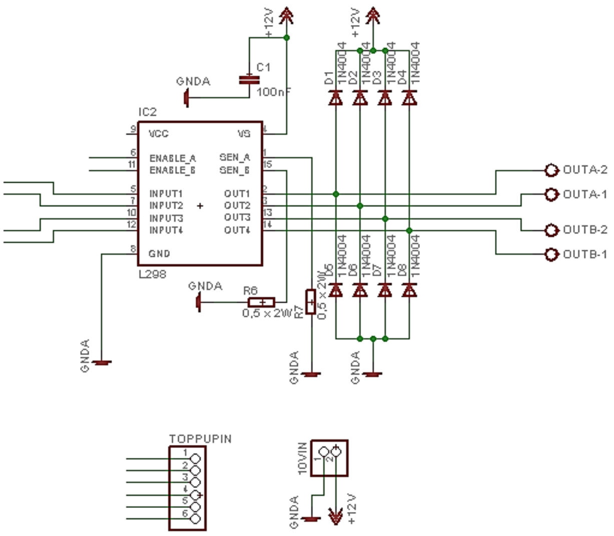
Display output layer design in which it consist of a display unit and Power unit

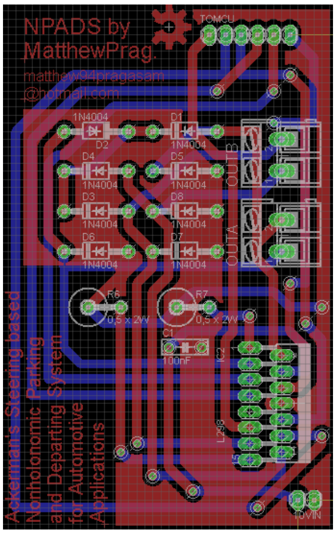




Display Module

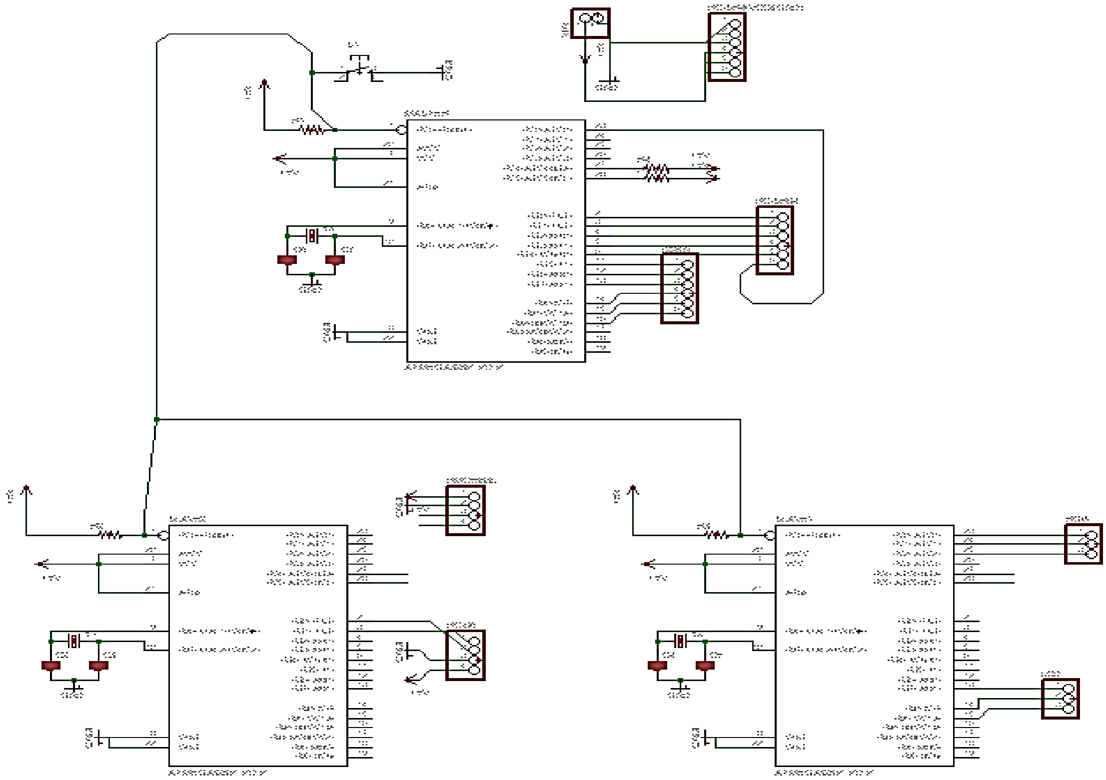
Schematic of constructed L298N motor driver





Motor driver with L298N

Schematic of the main processing unit layer using I2C interface.



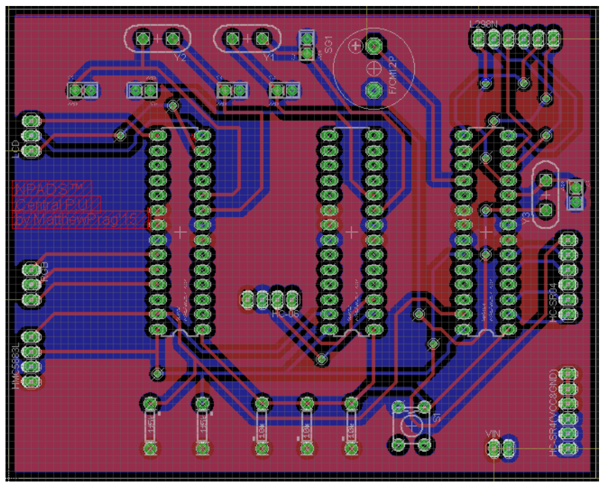
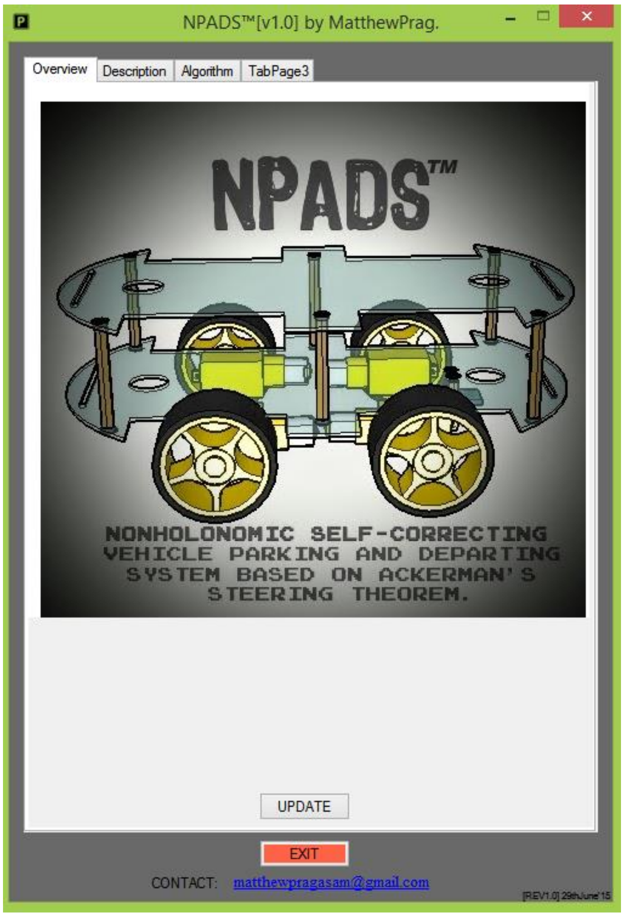
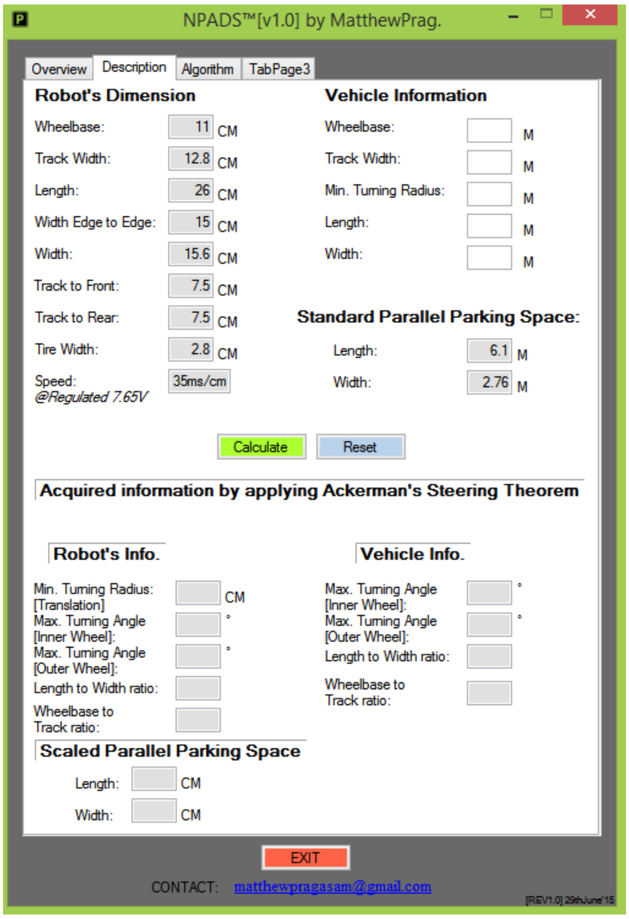


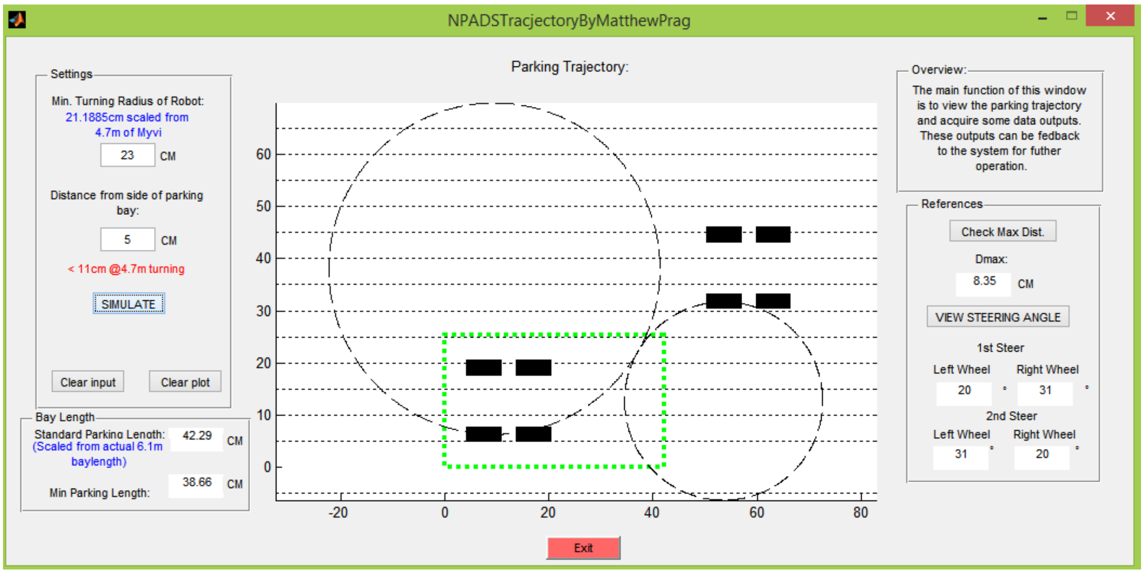
Figure 21: Central PU with I2C com. With exteral Pull-up resistors.

APPENDIX II: GUIS AND SCRIPTING



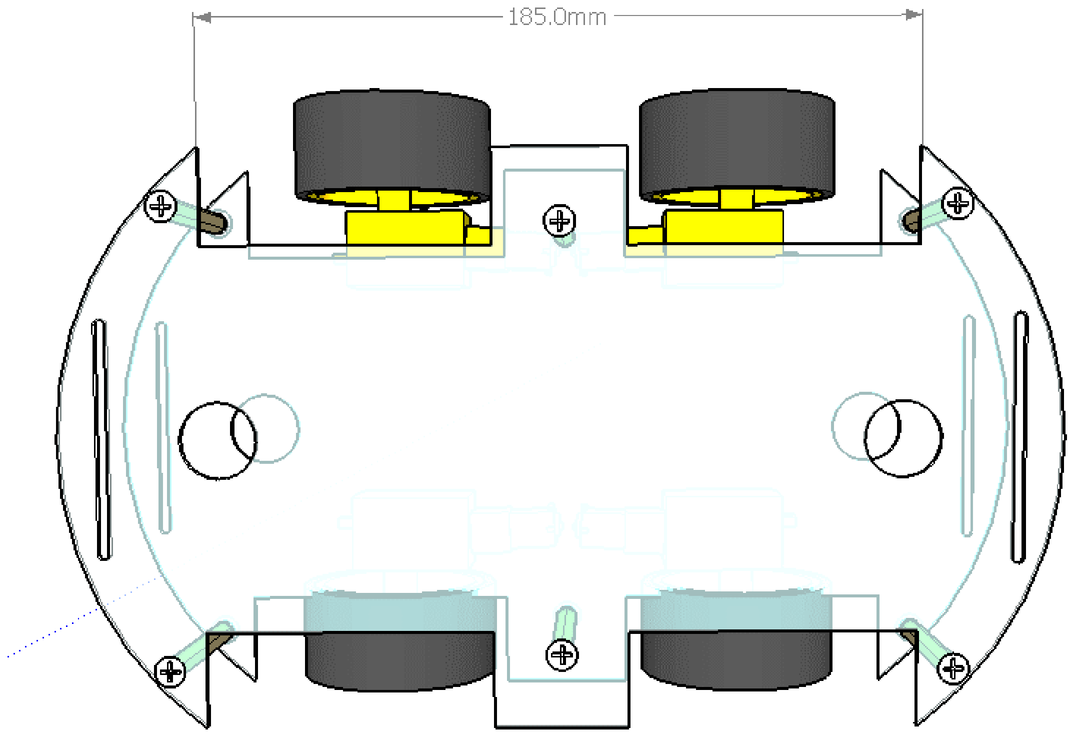


Developed **Visual studio GUI (.net scripting)**

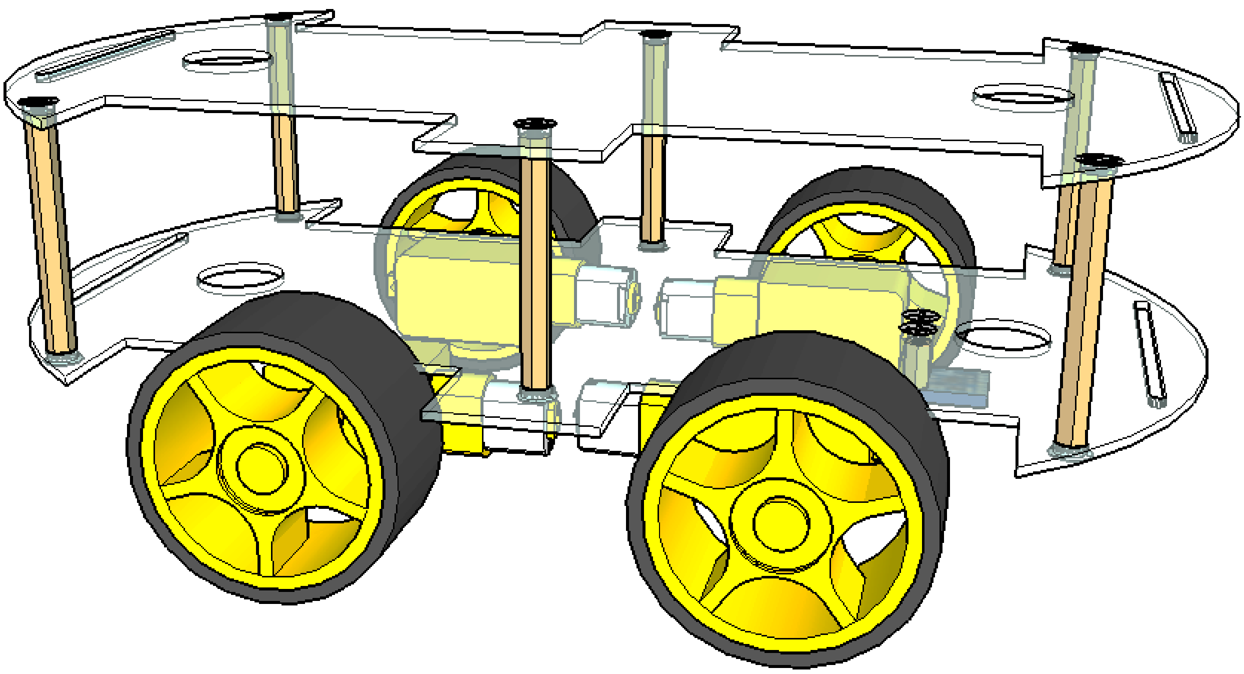


MATLAB simulator for parking trajectory

APPENDIX III: 3D MODEL DESIGN (CAD) DEVELOPED ON GOOGLE SKETCHUP.



Top View



Side View