APS: An Automatic Parking System for autonomous vehicles

Eric Jimenez Barril - Aleix Sancho Jimenez - Arnau Perich
Iglesias - Misael Samir Malqui Cruz - Mauro Moises Ortega López
Universitat Autònoma de Barcelona
Taller de Modelització

(Dated: April 21, 2024)

This work is aimed to develop an automatic parking system to be implemented in autonomous vehicles. At the first stage, the vehicle detects the environment via a LIDAR sensor and stores it. This information is used to identify the feasible maneuvers, which are previously studied, and finally it follows one of these trajectories. The efficiency of this program is studied by testing the recognition and execution in different configurations, and finally a trailer is studied from an iterative approach.

CONTENTS

I. INTRODUCTION AND MOTIVATION

Nowadays, due to the massification of the number of vehicles at the big cities, having as a facility an *Automatic Parking System* implemented in the car is very well received, given that having an assistance for parking in a reduced space can save effort and time. These are becoming to be common on the newer generations cars, but yet, using them is not as fast as parking manually.

In this work is presented a mathematical model that assists the parking maneuvers, starting from the process of finding viable parking slots, and takes in account the limitations on space, making doable the most possible cases. The parking process in sufficiently large slots are done in the least number of maneuvers, optimizing the time took on the process.

II. VARIABLES AND NOTATION

A. Maneuver 1

- P_i Initial point of trajectory (diagonal parking).
- P_f Final point of trajectory (diagonal parking).
- Δx Horizontal distance between P_i and P_f.
- Δy Vertical distance between P_i and P_f.
- C_i Center of the circumference i.
- C Distance between the two circumfere.nces of the maneuver
- Cg Center of rotation of the diagonal maneuver.
- R_i Radius of the circumference i.
- θ Angle that describes the arc of circumference which is the vehicle's trajectory over both circumferences.
- hi Vertical distance from the initial reference point to the origin.
- li Horizontal distance from the initial reference point to the origin.
- α Angle defined by the lateral limits of the car park with respect to the horizontal axis.

In the following variables, all of them for the diagonal maneuver, $j \in \{i, c, e\}$ where:

- i:= inner wheel
- c:= ideal wheel
- e:= outer wheel

- P_{0i} Initial position of the "i" rear wheel of the car.
- P_{0id} Initial position of the "i" front wheel of the car.
- P_{fi} Final position of the "i" rear wheel of the car in the first turn.
- P_{fid} Final position of the "i" front wheel of the car in the first turn.
- P_{fai} Final position of the "i" rear wheel of the car.
- P_{faid} Final position of the "i" front wheel of the car.

B. Notation

- $\overline{XY}$ Distance between arbitrary points X and Y.

C. Constants

- w_l Vehicle width.
- w Vehicle length.
- a Parking width.
- b Parking length.
- R_{car} Minimum turning radius of the vehicle.

III. ASSUMPTIONS

It is impossible to take into account all the details of the parking process. For that reason it has been considered some assumptions for this mathematical model, to minimize the difficulty of all the calculations. However, the principal objective of this work is to park a car automatically, so it is important to choose this simplifications accurately to best approximate the model to reality.

The following assumptions has been made:

- Some simplifications of the car structure has been made: it is considered that the car is a rectangle, with the rear wheels axis in the back of this rectangle.
- Some properties of the car has not been considered: there are not considered differences between front-wheel drive and rear-wheel drive, the steering wheels are exclusively the front ones, and it has been assumed that the car does not drift.
- Is only studied the trajectory that the car follows, dynamics are not considered: is not decided which velocity the car needs to park and what forces are applied to the system.
- The environment is not given; the car is able to recognize it. But at this point, it has been supposed that the car goes straight.

IV. DESCRIPTION OF THE SYSTEM

A. Steering Geometry

This work focuses on front wheels steering vehicles. First, the following statement must be introduced:

Ackermann Principle. [1] The optimal turning path is given when every wheel has the same turning center.

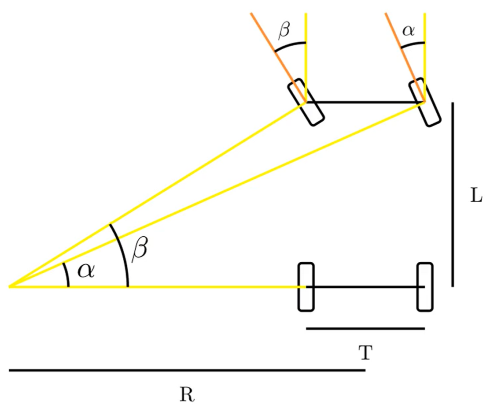

FIG. 1. Schema of the Ackermann principle.

Nowadays, most of conventional cars follow Ackermann's rule, or at least some variation of it. Due to the widespread use of this rule, in this study it is considered that the car that is going to park strictly follows Ackermann, this makes possible working with circumferences when calculating the meneuvers. There are some consequences from this principle related to the steering geometry:

- Given a turn path, the left and right front wheels have different angles regarding to the longitudinal axis of the vehicle.
- The center of turn stays on the prolongation of the line that connects the back wheels.
- The inner back wheel is tangent to the circumference the car describes.

Moreover, as the radius of turn is measured from the center of the car's axis, it is convenient to identify an ideal wheel placed in the middle of the front wheels from which the parameters related to the maximum angle the wheels can make will be calculated. Knowing the angle θ or turn radius of the circumference described by the ideal wheel it is easy to determine the angle of the two front wheels via the following expressions:

$$\alpha = \arctan \frac{L \tan\theta}{L - \frac{T}{2}\tan\theta}$$
$$\beta = \arctan \frac{L \tan\theta}{L + \frac{T}{2}\tan\theta}$$

Were both expressions make use of the notation used in the Fig 1.

All the maneuvers treated in this study are based on this principle, given that it makes working with circumferences a viable way for reach the objective: be able to park a car automatically, in a deterministic way when it is possible.

B. Kinematics

In this work the different maneuvers are all based kinematic models of the vehicle. Firstly, is important to know that the car position can be defined using only 3 parameters: two spacial coordinates and the vehicle's angle. The two spacial coordinates are the ones from the middle point between the back wheels, and the angle α described at *II* .

Using this variables, two magnitudes are added to them: velocity and turn radius. With them, the temporal evolution of the different vehicle's states are fully determined.

Considering x_k, y_k the spacial coordinates in the instant k and α_k the angle of the car at that same instant, naming the velocity and the turn radius v and θ, considering the simplifications, the following expression describes the system temporal evolution:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \alpha_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \alpha_k \end{bmatrix} + \begin{bmatrix} v\cos\alpha_k \\ v\sin\alpha_k \\ \frac{v}{L}\tan\theta \end{bmatrix} \mathrm{d}t\,.$$

This way of thinking about how the vehicle motion is defined, together with the Ackermann Principle, are the main tools required to describe the parking maneuvers treated in this work.

V. PARKING PROCESS

The parking system consists on three main stages. First, the vehicle must recognize the environment so as to detect obstacles. These observations are gathered into a global map, which stores the positions of the objects. With this information, the system has to identify the slots present in the environment. These types are described in the subsection V C. Finally, given the possible available configurations to park, the final maneuver is decided. The vehicle executes the maneuver and the program is considered complete when it reaches the desired position.

A. Recognition

In order to be aware of the possible obstacles, the vehicle has to recognize the environment. Besides, this detection must be carried out at each step of motion, so that its map can be updated with the latest objects. To reach that goal, the vehicle is equipped with a devise located at the center of the rear axle. The implemented sensor is called LIDAR, and it consists on measuring distances by emitting lasers in all directions [2]. In order to program the behavior of light, it was used a discrete version of its motion, traveling a distance $c\,dt$ at each step. When it detects an obstacle, its velocities is inverted and starts moving backwards until arriving the sensor. This way, the length of its trajectory is given by [3]

$$d = \frac{c\Delta t}{2}. \tag{1}$$

Since the system contains its position and the orientation of the emitted laser a priori, it can compute the last position of the emitted light. By repeating this process, it scans the environment through an iteration of laser emissions for several angles. An example of this process completed for one state is represented in Fig. 2. All positions obtained in one state are transformed in a polygon, which represents the available space that can be observed. With this information, the program computes the union of this and previous polygons. This algorithm results in a global map that takes into account current and past observations, as desired.

B. Decision

After computing a map with the obstacles positions, the vehicle has to identify what it is observing. On the one hand, the global polygon must be interpreted as the available space. Hence, it needs a condition to recognize whether it is inside of the available option or not. In this work, this polygon is saved as a `polygon` object from the library `shapely`, so this method was already created [4].

On the other hand, the program has to make a decision of the maneuver it will perform. For that, it has to identify the configuration of the parking and then complete its correspondent trajectory. This question was solved by studying some particular cases, presented in the next section, and saving them into the program. As presented in Alg. 1, at every stage of the vehicle's motion a new map is constructed. Subsequently, it checks all known

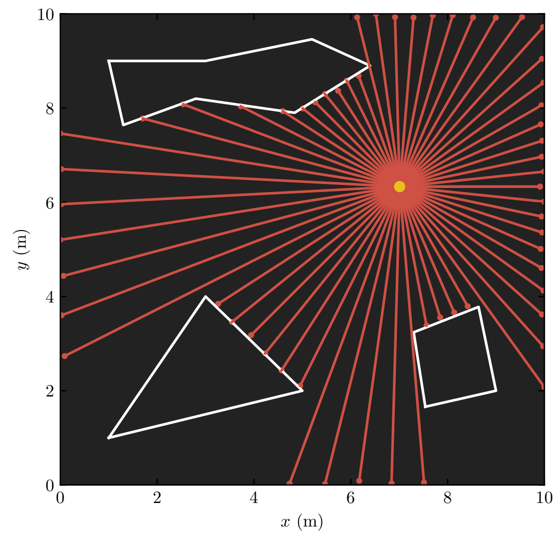

FIG. 2. LIDAR detector implemented in Python. Red lines represent the light, white objects the obstacles, and the yellow point the position of the sensor.

spaces to park, and if one of them exists, it starts the associated maneuver. This identification is executed by finding an intersection between the space and the global polygon, now with the method `contains` from the same library.

Algorithm 1 Decision

$\mathbf{q}_0 = [x_0, y_0, \theta_0, \phi_0]$ initial state
car = Car($\mathbf{q}_0$)
obstacles
environment = car.recognize(obstacles)
while car.within(environment) **do**
  $\mathbf{q}_{k+1}$ = motion.equations($\mathbf{q}_k, v, \alpha$)
  environment = car.recognize(environment, obstacles)
  for space in spaces **do**
    if environment.contains(space) **then**
      car.park(space)
      break
    end if
  end for
end while

C. Maneuvers

In this work there are three studied configurations for parking: parallel, diagonal in forward direction, and diagonal in reverse direction. Notice that battery configuration is just a particular case of the diagonal case but for right angles, so their results are already obtained in these sections. In general, all possible scenarios are studied with the same approach. First, a system of reference is created, and then a possible maneuver is selected. However, the trajectory to follow is constrained by the

physical properties of the car and the parking. For that reason, all bounds are searched, regarding initial position, displacement and rotation. This study concludes with a set of feasible regions, which determined the final maneuver.
During the rest of the paper it has been assumed to make the graphs that $a := 2.5$, $b := 5$,$w := 1.8$, $w_l := 4$.

VI. FIRST MANEUVER: PARALLEL PARKING

The first maneuver is designed to concatenate two turn paths given the initial and final point, P_i, P_f such that the vehicle is parallel to the sidewalk in both of them as it is shown in figure 3.

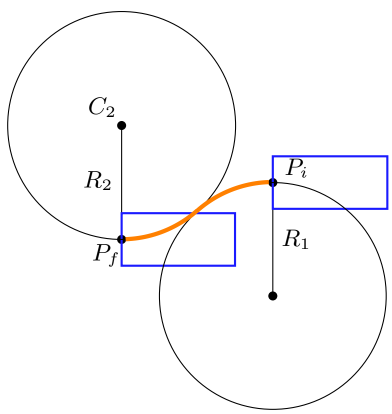

FIG. 3. Back ideal wheel trajectory in a reverse parallel parking.

First of all, it must be known how these two circumferences are placed in the plane. By the Ackermann Principle:

- P_i is the highest point of the first circumference and P_f is the lowest point of the second one.
- By the previous point, the centers of the circumferences, C_1 and C_2, are located in the line perpendicular to the sidewalk at P_i and P_f, respectively.
- At the point of change of circumference, C_1 and C_2 are both in the line made by the prolongation of the back axis of the car. And so, both circumferences are tangent to each other at this point.

Observe that given P_i, P_f and R_1 the first radius of turn, as the two circumferences are tangent, if $C = \overline{C_1C_2}$ then $R_2 = C - R_1$. And so, the following lemma:

Lemma 1. Given P_i and P_f, then C is invariant from the radius.

Proof. Consider the rectangular triangle shown in Figure 4. Let Δx and Δy be the horizontal and vertical distance between P_i and P_f. First notice that $R_1 = \overline{AP_F} + \Delta y$ and $\overline{AC_1} = \Delta x$. Then by the Pythagorean Theorem:

$$\begin{aligned} C^2 &= (R_2 + \overline{AP_F})^2 + \overline{AC_1}^2 \\ &= (R_2 + \overline{AP_F} + \Delta y - \Delta y)^2 + \Delta x^2 \\ &= (R_2 + R_1 - \Delta y)^2 + \Delta x^2 \\ &= (C - \Delta y)^2 + \Delta x^2 \\ &= C^2 - 2C\Delta y + \Delta y^2 + \Delta x^2 \end{aligned}$$

Thus:

$$C = \frac{\Delta x^2 + \Delta y^2}{2\Delta y} \tag{2}$$

And C is invariant. □

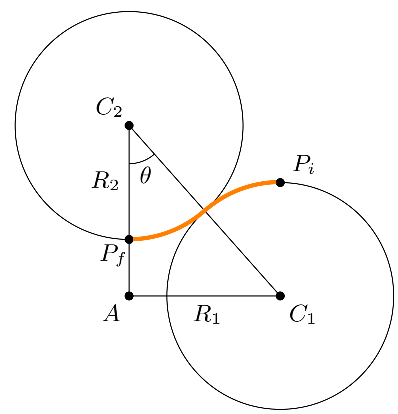

FIG. 4. Triangle to calculate θ and the distance between C_1 and C_2 given P_i and P_f.

Notice that this fact implies that given P_i and P_f is irrelevant which R_1 is choosed, C is a constant and R_2 is determined and, thus, unique. In fact, the following result is true:

Corollary 1. *Given P_i and P_f, then $\overrightarrow{C_1C_2}$ is invariant from the radius.*

Proof. As it is known that C is constant, then the slope of the vector must be calculated such that

$$\tan\theta = \frac{\overline{AC_1}}{R_2 + \overline{AP_f}} = \frac{\Delta x}{C - \Delta y}$$

and θ is determined. □

A. Reverse

The approximation to the maneuver focuses on the bounding of the defined parameters in order to produce an permissioned execution without crashing.

1. Reference system

To begin with the results of the reverse parallel parking, firs of all, it must be set a system of reference to properly define the trajectory of the car.

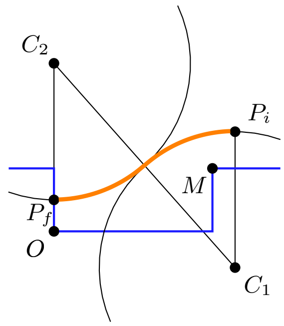

FIG. 5. System of reference and the points defined.
C

As shown in Figure **??**, the system's origin is choosed at the left bottom corner of the parking spot. Are identified the following points $C_i = (li, hi - R)$, C_f, $P_i = (li, hi)$, $P_f = (0, \frac{w}{2})$ and $M = (a, b)$ where R is the radius of turn of C_1. Every point has been presented except M, which is the unique point of crash because of the election of P_f. Furthermore, lets consider the constants R_{min} as the minimum radius of turn the car can execute and w as the width of the car.

2. Bounding the allowed vertical distances

Let's consider that the car is not allowed to surpass a maximum vertical distance, h_{max}. This distance will be touched at first by the front-left wheel, where the circumference that describes has a radius of $R' = \sqrt{(r + \frac{w}{2})^2 + w^2}$. So a restriction of the radius is given when:

$$h_{max} = hi - R + R'$$

i.e. when the highest point of the circumference that describes the front-left wheel equals the highest point available. Thus, the minimum radius by this restriction, R_h, is given by:

$$R_h = \frac{h_{max}\,(1 - 2hi) + hi^2 - w^2}{2\,(hi - h_{max})} - \frac{w}{2}$$

3. Bounding the allowed radius

A first boundary of the first radius of turn is:

$$C - R_{min} \geq R_1 \geq R_{min}$$

But, this one doesn't consider the point of crash M. It will be defined the minimus radius of turn of the first circumference such that it verifies $\overline{P_i\,C_1} = \overline{M\,C_1}$, so:

$$\begin{aligned} R - \frac{w}{2} &= \sqrt{(li - a)^2 + (hi - R - b)^2} \\ R^2 - R\,w + \frac{w^2}{4} &= (li - a)^2 + R^2 \\ &\quad -2\,R\,(hi - b) + (hi - b)^2 \\ R &= \frac{(li - a)^2 + (hi - b)^2 - \frac{w^2}{4}}{2(hi - b) - w} \end{aligned}$$

Thus, the boundary of the first radius is:

$$C - R_{min} \geq R_1 \geq \max\left(R_{min}, R_h, R\right) \tag{3}$$

And given R_1, R_2 is determined.

4. Bounding the allowed horizontal distances

Now, a boundary for the horizontal component of P_i can be stablished, as its extrems will occur when there is only one possible radius to execute. So lets consider the equalities:

$$\begin{aligned} C - R_{min} &= \frac{(li - a)^2 + (hi - b)^2 - \frac{w^2}{4}}{2(hi - b) - w} \\ C - R_{min} &= R_{min} \end{aligned}$$

where $C = \frac{li^2 + \Delta y^2}{2\Delta y}$ and $\Delta y = hi - \frac{w}{2}$. And get the equations:

$$\begin{aligned} 0 = &\left(\frac{1}{2\,\Delta y} - \frac{1}{2\,hi - 2\,b}\right) li^2 + \frac{1}{2\,hi - 3\,b}\,li + \\ &+ \left(\frac{a^2 + (hi - b)^2 - \frac{b^2}{4}}{2\,hi - b} - \frac{\Delta y}{2} - R_{min}\right) \\ li &= \sqrt{\Delta y\;(4\,R_{min} - \Delta y)} \end{aligned}$$

For the first equation, the quadratic one, the two solutions correspond to the maximum and minimum value allowed to li. Let s_1, s_2 be those solutions such that $s_1 \leq s_2$. Thus:

$$s_2 \geq a \geq \max\left(s_1, \sqrt{\Delta y\,(4\,R_{min} - \Delta y)}\right) \quad (4)$$

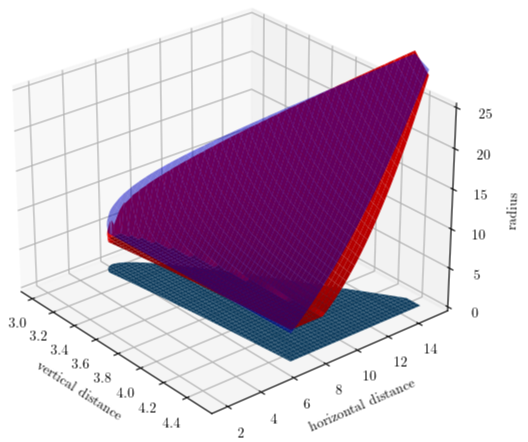

FIG. 6. Allowed radius in function of the horizontal and vertical distance from the origin of the system of reference.

Combining both functions it is plotable the allowed radius given li and hi shown in Figure 6.

5. *Bounding the dimensions of the parking spot*

Consider the minimum value of li given by the minimum radius the car can execute, R_{min}, is given by;

$$li_{min} = \sqrt{\Delta y\,(4\,R_{min} - \Delta y)}$$

Now, given that the point of crash M gives the minimum radius based on the back - right wheel, which equation is:

$$(x - li)^2 + (y - (hi - R))^2 = (R - \tfrac{w}{2})^2$$

As, $M = (m_x, m_y)$, the minimum value of m_x, $m_{x\,min}$, is given when $R = R_{min}$ and M belongs to the previous circumference equation:

$$(a - li_{min})^2 + (b - (hi - R_{min}))^2 = (R_{min} - \tfrac{w}{2})^2$$
$$a_{min} = li_{min} - \sqrt{(R_{min} - \tfrac{w}{2})^2 - (b - (hi - R_{min}))^2}$$

6. *Lack of optimization*

A first intuition to choose a initial radius to execute the maneuver would be to select the on which produces the least distance the car travels. This distances, d, is:

$$d = R_1\,\theta_1 + R_2\,\theta_2$$

where R_i are the radius of the two circumferences and θ_i is the angle that the car travels in each circumference. By Corollary 1 the vector $\vec{C_1 C_2}$ is constant. Therefore $\theta = \theta_1 = \theta_2$ and:

$$d = R_1\,\theta + R_2\,\theta = (R_1 + R_2)\,\theta = C\,\theta$$

where $C = \overline{C_1 C_2}$. Thus, there is no optimal radius to choose, the distance travelled is constant given P_i and P_f.

VII. SECOND MANEUVER: DIAGONAL PARKING

The objective of this maneuver is to park the vehicle in a parking slot perpendicular to the direction in which it is circulating, or at an angle less than 90° and greater than 45° relative to its direction.

The idea is to trace a circumference with the vehicle, with the proper angle until it is parallel to the angle already mentioned, and once it is in this position, go back in the case of the reverse maneuver (go forward in the case of the forward maneuver) in a straight line until the vehicle is in the desired position.

It should be noted that this maneuver will not always be possible, it will depend on the minimum turn angle of the car, the angle between the parking slot and the direction of circulation, the width of both: the car park and the car, and the vertical and horizontal distance that the vehicle is away from the car park.

This section is focused on finding all the possible maneuver options that can be chosen to carry out the parking. It will be enough to choose one of them to park the vehicle.

A. Reverse

1. *Reference system*

First, it is set a reference system.

As the reference origin, is chosen the intersection between the imaginary line that defines the outer side of the car park and the line parallel to the vehicle that passes through the point of the car park furthest from said vehicle, thinking that is being drive in a direction parallel to the axis of the abscissas and in positive sense in an imaginary Cartesian plane. It is assumed that is being drive in a right lane and it will be parked on the right side.

The right rear corner is taken as the vehicle's initial reference system.

In the Figure 7 can be seen an idea of the system, with the already chosen variables.

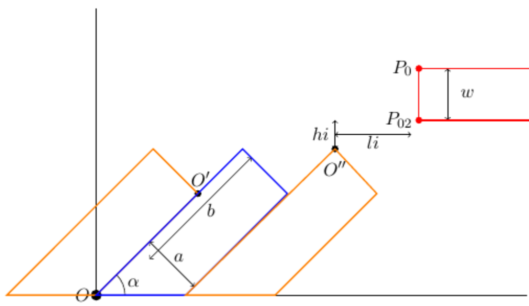

FIG. 7. Representation of the reference system for reverse maneuver.

As can be seen in the Figure 7, to facilitate the calculations of this maneuver, will be convenient to choose the origin for the horizontal (li) and vertical (hi) distance in O''.

In order for the vehicle to perform automatic parking, it will stop in a suitable position and perform certain wheel turns automatically. To do this, it is assumed that the vertical distance (that is, the one that would correspond to the ordinate axis in the reference system) between the car and the parking slot is fixed (this is initially chosen based on a maximum height -which can be infinite in the case in which there are not vertical obstacles-), and the car makes a choice of the turning angle and the frontal distance at which the maneuver begins (understanding this distance as the distance between the origin of coordinates and the vehicle).

To ease the calculations, the following constants are fixed:

$$A = \frac{a}{\sin\alpha} + b\cos\alpha \qquad B = b\sin\alpha + hi + w$$

$$a' = (A, B\sin\alpha)$$

$$P_0 = (A+li, B) \qquad P_{01} = P_0 - \left(0, \frac{w}{2}\right) \qquad P_{02} = P_0 - (0, w)$$

2. Bounding the allowed vertical distances

It should be noted that, in most car parkings, there exists a vertical limitation, so the vehicle cannot be or go "high" as need. As it is assumed that is going to be parked in a single maneuver, it will suffice to choose an appropriate hi.

First of all, it should be noted that the highest point of the vehicle in the maneuver is reached by the front outer wheel, so by limiting this height, hi can be determined.

This maximum height will be reached when the vehicle is performing the maneuver with the minimum radius, since, in that case, the arc of circumference that it describes has a greater turning angle and therefore the front wheel will pass through a higher point.

The highest point of said circumference is reached when the outer front wheel passes through the component x of the center of rotation, since the horizontal component of the highest point of a circumference coincides with the horizontal component of its center.

Furthermore, if the radius with respect to the ideal wheel is denoted as r_h, we can relate the radius of the circumference around the front outer wheel to the minimum turning radius as

$$R' = \sqrt{\left(R_h + \frac{w}{2}\right)^2 + wl^2}$$

Now, h_{max} is denoted as the maximum height that the vehicle can reach. Therefore, the following equation needs to be solved:

$$h_{\max} = Cg_x + R'$$

were from

$$h_{\max} = (B - w - R_h) + \sqrt{\left(R_h + \frac{w}{2}\right)^2 + wl^2}$$

we have that

$$R_h = \frac{\frac{w^2}{4} + wl^2 - (h_{\max} + w - B)^2)}{2h_{\max} + w - 2B}$$

The positive solution gives a first restriction in order to find the minimum radius.

3. Bounding the allowed radius

To find the maximum radius, it is useful to find the maximum turning radius such that the outer part of the vehicle in the final position (p_f) of the turn (the one where the vehicle is parallel to the angle of the parking slot) intersects with the outer side of the parking slot (that is, the left line that delimits the parking slot is tangent to the turning circumference at the point turn end) as we can see in the Figure 8.

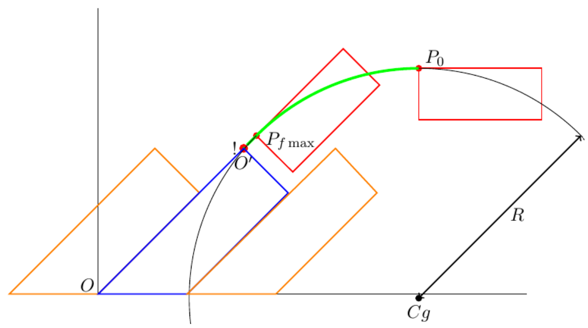

FIG. 8. Situation in which the car intersects with the outside obstacle

Now, we don't know the radius of gyration and the final position. By Ackermann's principle, we know that when performing the maneuver setting a turning angle, the circumference will always be perpendicular to the rear of the vehicle, therefore, we can know the turning center (Cg) equaling the initial and final position of the vehicle. vehicle. Fixing the following notation:

$$P_{f\,\max} = \left(\frac{y_{f\,\max}}{\tan\alpha}, y_{f\,\max}\right)$$

By Ackermann

$$Cg = (A + li, B - R)$$

$$Cg = \left(\frac{y_{f\,\max}}{\tan\alpha} + R\sin\alpha, y_{f\,\max} - R\cos\alpha\right)$$

where we get:

$$y_{f\,\max} = B(1+\cos\alpha) - (A+li)\sin\alpha$$

$$R = \frac{(A+li)\sin\alpha - B\cos\alpha}{1-\cos\alpha} \tag{5}$$

Now, to find this minimum radius, is required to find the radius such that the inner part of the vehicle intersects with the upper left corner of a supposed parking slot on the right (in the plane), therefore, in case there is another vehicle (well parked) on the right, the vehicle is not going to collide with it.

Is already known that

$$Cg = (A + li, B - w - r)$$

Also, the starting point (with respect to the inner wheel) is

$$P_{02} = (A + li, B - w)$$

and the end point

$$O'' = (A, B\sin\alpha)$$

Therefore, equating the distances to the center of rotation

$$\sqrt{(A - (A+li))^2 + (hi - r)^2} = r$$

where

$$r = \frac{li^2 + hi^2}{2hi} \tag{6}$$

So, relative to the ideal wheel, the radius allowed according to equations 5 and 6 are

$$\max\left(R_{car}, R_h, r + \frac{w}{2}\right) < r_{id} < R - \frac{w}{2} \tag{7}$$

where R_{car} is the turn radius of the car (relative to the ideal wheel).

4. *Bounding the allowed horizontal distances*

From the previous steps, the interval of distances was determined by making the left hand side and the right hand side of the equation (7) equal. This gives two solutions.

At these points, there is only one possible radius to choose. The frontal distance will be maximum or minimum. If not, there is not possible way to find any affordable radius.

These distances are obtained by solving

$$\begin{cases} r + \frac{w}{2} = R - \frac{w}{2} \\ R_{car} = R - \frac{w}{2} \end{cases}$$

then

$$\begin{cases} \frac{li^2+hi^2}{2hi} + w = \frac{(A+li)\sin\alpha - B\cos\alpha}{1-\cos\alpha} \\ R_{car} = \frac{\sin\alpha(A+li) - B\cos\alpha}{1-\cos\alpha} - \frac{w}{2} \end{cases} \tag{8}$$

whence, using the first expression of (8):

$$li^2\left(\frac{1-\cos\alpha}{2hi}\right) + li(-\sin\alpha) + \Big[\left(\frac{hi}{2} + w\right)(1-\cos\alpha)$$

$$- A\sin\alpha + B\cos\alpha\Big] = 0 \tag{9}$$

The expression (9) can be simplified, agrouping the terms considering the expressions as a second degree polynomial on li:

$$L_2 li^2 + L_1 li + L_0 = 0$$

and given the width of the car and the dimensions of the parking slot, the minimum and maximum frontal distances are

$$li_{\min 1} = \frac{-L_1 - \sqrt{L_1^2 - 4L_2L_0}}{2L_2}$$

$$li_{\max} = \frac{-L_1 + \sqrt{L_1^2 - 4L_2L_0}}{2L_2}$$

Now, from the second expression of (8) follows:

$$li_{\min 2} = \frac{(R_{car} + \frac{w}{2})(1 - \cos\alpha + B\cos\alpha)}{\sin\alpha} - A$$

Therefore the final expression is:

$$\max(li_{\min 1}, li_{\min 2}) < li < li_{\max} \tag{10}$$

Now, it is possible to bound the possible radius given hi and li. Shown in Figure 9.

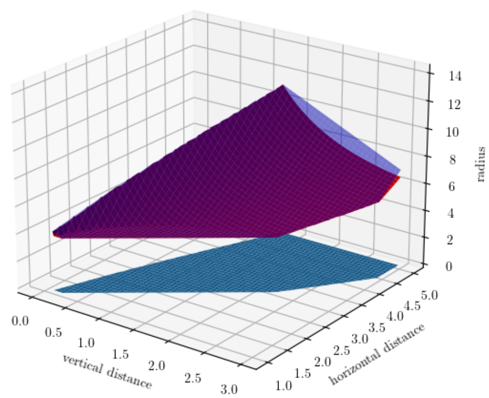

FIG. 9. Allowed radius in function of the horizontal and vertical distance from the origin of the system of reference

5. *Completion of the maneuver*

Once the initial positions and radii are decided, the vehicle starts turning. It is at this stage when the stop condition has to be determined.

Firstly is needed to find an expression for the final position. With some trigonometry it is suffice for archieving this. The Figure 10 can help understanding the intuition behind.

According to Ackermann, the tangent line to the rear of the vehicle passes through the center of rotation and is perpendicular to the two segments that delimit the sides of the parking slot. Hence, since the distance to the center of rotation (r_{id}) is known, relative to the ideal rear wheel, the final position (P_f) is

$$P_f = \left(A + li - r_{id}\sin\alpha, B - \tfrac{w}{2} + r_{id}(\cos\alpha - 1)\right) \tag{11}$$

and relative to the inner rear (P_{fi}) and outer wheels (P_{fe}), it follows:

$$P_{fi} = P_f - \frac{w}{2}(-\sin\alpha, \cos\alpha)$$
$$P_{fe} = P_f + \frac{w}{2}(-\sin\alpha, \cos\alpha)$$

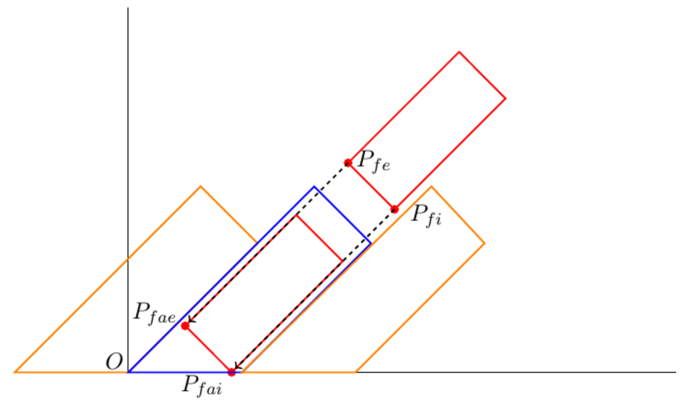

FIG. 10. Completion of the diagonal reverse maneuver.

Once these three points are determined, it is possible to calculate the end point of the maneuver for each point, which will be appointed as P_{fa},P_{fai}, P_{fae}. The line that joins these points and the one that joins the points P_f,P_{fi}, P_{fe} are parallel, therefore perpendicular to the segments that delimit the car park, thus its slope is known (is $-\frac{1}{\tan\alpha}$), furthermore, the point P_{fai} is on the x-axis (since is wanted to be as close as possible), therefore, the line cuts this axis.

Thus, by trigonometry the point P_{fi} is

$$P_{fai} = \left(\cot\alpha(\tfrac{w}{2} + r_{id} - B) + \csc\alpha(\tfrac{w}{2} - r_{id}) + A + li, 0\right)$$

And since the other points are part of the line

$$y = -\cot\alpha(x - P_{faix})$$

where P_{faix} is the x component of P_{fai}, they can be deduced, because they are at a distance $\frac{w}{2}$ and w respectively.

$$P_{fa} = P_{fi} + \frac{w}{2}(-\sin\alpha, \cos\alpha)$$

At this point, the maneuver is fully determined.

B. Forward

1. *Reference System*

The same reference as in the reverse maneuver is set. However, now the whole car is considered (i.e. now all four wheels are taken into account) as it is necessary to avoid collisions with obstacles. In this case, to facilitate the calculations, is considered that the car is circulating in a supposed left lane in the opposite direction to the previous one, thus, the car park will have the same shape, but the vehicle is in the opposite direction.

2. *Bounding the allowed vertical distance*

In this maneuver, by the Ackermann principle, there is no need to consider an explicit expression of the minimum radius because this is already determined by the initial position due to the back wheels does not make a bigger circumference compared to the front wheels ones.

3. *Bounding the allowed radius*

There are considered the same assumptions as in the reverse maneuver, and, in the same way, it is required to look for the angle of rotation and the frontal distance in which the maneuver begins.

In this case, the radius that needs to be found is the one such that the circumference that defines the front outer wheel intersects with the obstacle located outside it. In this way, is prevented that any part of the car from collides with the obstacle (as long as it follows the turn and finishes the maneuver) as can be seen in Figure 11.

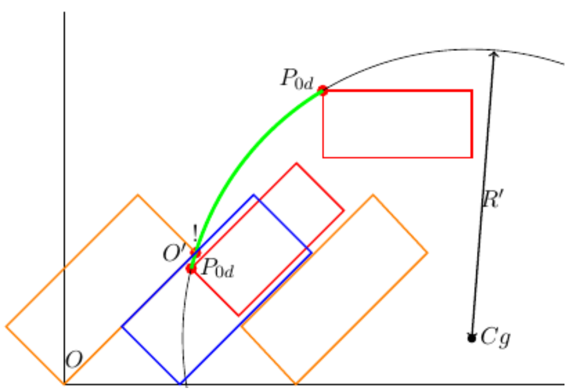

FIG. 11. Situation in which the car intersects with the outside obstacle.

This maximum radius is useful as long as the position of the front wheels at the point where it stops turning, does not have "overcome" the straight line that defines the external obstacle.

Otherwise, the maximum radius will be the radius such that at the final point of the turn, the line that defines the outside of the car coincides with the line defined by the inner part of the obstacle.

Now, the first case is studied with the same idea as in the previous maneuver. The distance from the center of rotation to the initial position (in this case from the front outer wheel) and the final point is set (which is denoted by O') is the same, since both points are part of the circumference. In this case, by Ackermann's principle the radius to determine ($R_{\max 1}$) is

$$R_{\max 1} = \sqrt{R'^2 - wl^2} \tag{12}$$

where R' denotes the radius of the circumference described by the front outer wheel and wl stands for the length of the car.

Thus, working on R', $R_{\max 1}$ will be determined. Again by Ackermann, the center of rotation is:

$$(A + li, B - R' \cos\gamma)$$

where

$$\gamma = \arcsin \frac{wl}{R'}$$

and the start and end points are:

$$P_{od} = (A + li - wl, B)$$

$$O' = (O'_x, O'_y) = (b\cos\alpha - a\tfrac{\cos^2\alpha}{\sin\alpha}, b\sin\alpha - a\cos\alpha)$$

Therefore, the objective is to solve

$$|P_{od} - Cg| = |O' - Cg| = R'$$

that equals to the following equation of R'

$$\begin{aligned} &\sqrt{wl^2 + R'^2\cos^2\gamma} = \\ &\sqrt{(O'_x - A - li)^2 + (O'_y - B + R'\cos\gamma)^2} \end{aligned} \tag{13}$$

To simplify the calculations, the following constants can be fixed:

$$P = a\frac{(\cos^2\alpha + 1)}{\sin\alpha} + li$$

$$Q = a\cos\alpha + hi + w$$

By squaring both sides from (13) :

$$R'^2 = P^2 + (R'^2\cos\gamma - Q)^2$$

and by doing some algebra, R' can be found with the following expression

$$R' = \left| \sqrt{\left(\frac{P^2 + Q^2 - wl^2}{2Q}\right)^2 + wl} \right| \tag{14}$$

Where the absolute value is taken because R' corresponds to a distance, and using (12), R_{max1} can be fully determined.

Now the other case needs to be studied

The objective is to find the line defined by the points P_fe and P_{fi}. It has slope $\tan(\alpha)$ since at the end of the turn the vehicle is parallel. Thus, assuming that the radius is $R_{\max 2}$, by Ackermann:

$$P_{fe} = Cg + R_{\max 2}(-\cos\tfrac{\pi}{2} - \alpha, \sin\tfrac{\pi}{2} - \alpha)$$

namely:

$$P_{fe} = (b\cos\alpha + li + \tfrac{a}{\sin(\alpha)}, b\sin\alpha + hi + w) + R_{\max 2}(-\sin\alpha, \cos\alpha)$$

Therefore, the line passing through this point and parallel to the parking slot is:

$$y = \tan(\alpha)(x - b\cos\alpha - li - \tfrac{a}{\sin\alpha} + R_{\max 1}\sin\alpha)$$
$$b\sin\alpha + hi + w + R_{\max 1}(\cos\alpha - 1)$$

And, since is required this straight line to be coincident with the one that defines the external obstacle, it has to pass through the origin O, therefore, the following expressions must hold

$$0 = \tan(\alpha)(-b\cos\alpha - li - \tfrac{a}{\sin\alpha} + R_{\max 2}\sin\alpha)$$
$$b\sin\alpha + hi + w + R_{\max 2}(\cos\alpha - 1)$$

and isolating R_{max2}

$$R_{\max 2} = \frac{\frac{a}{\cos\alpha} + li\tan\alpha - hi - w}{\tan\alpha\sin\alpha + \cos\alpha - 1} \tag{15}$$

Thus, from equations (12) and (15):

$$R_{\max} = \min(R_{\max 1}, R_{\max 2})$$

Now, it is also required to find the radius such that the inner rear wheel collides with the inner obstacle as can be seen in Figure 12.

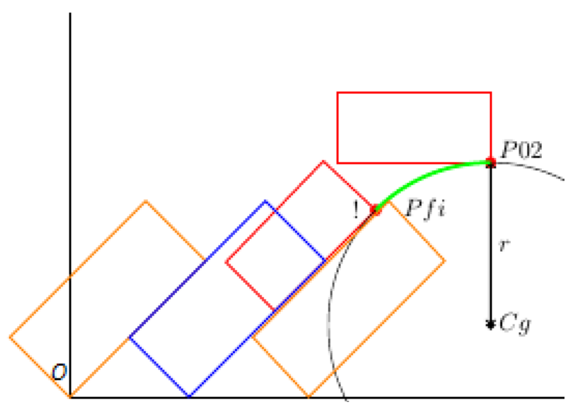

FIG. 12. Situation in which the inner rear wheel collides with the inner obstacle.

The radius corresponding to that inner rear's circumference is denoted as r

The following lines need to be defined

$$\begin{cases} t := y = \tan(\alpha)\left(x - \frac{a}{\sin\alpha}\right) \\ s := y = -\frac{1}{\tan\alpha}(x - A - li) + B - w - r \end{cases} \tag{16}$$

The intersection of these lines correspond to the extreme case where the side of the car at the end of the turn coincides with the inner obstacle, in case of having any radius greater than this and less than the maximum, the car does not collide at any time with the obstacle. Note that this point to determine is also the point where the vehicle will end the turn, since it is parallel to the parking slot.

Fixing:

$$Cg = (A + li, B - w - r)$$

By equating t and s,

$$x = \frac{\tan\alpha(B - w - r) + A + li + \tan^2\alpha\left(\frac{a}{\sin\alpha}\right)}{\tan^2\alpha + 1} \tag{17}$$

where, denoting $Cg = (Cg_x, Cg_y)$, and then substituting (17) in (16)

$$\begin{cases} x = \cos\alpha\sin\alpha Cg_y + \cos^2\alpha Cg_x + a\sin\alpha \\ y = \sin^2\alpha Cg_y + \sin\alpha\cos\alpha Cg_x - a\cos\alpha \end{cases} \tag{18}$$

Now, in order to find r, the following equation has to be solved

$$|(x, y) - Cg| = r$$

In order to archieve it, it is useful to agroup the constants as follows

$$\begin{aligned} P_1 &= \sin^2\alpha Cg_x - a\sin\alpha \\ Q_1 &= \sin\alpha\cos\alpha Cg_x - a\cos\alpha \\ P_2 &= (2\cos\alpha)(\cos\alpha(B - w) + (P_1\sin\alpha + Q_1\cos\alpha)) \\ Q_2 &= (2\cos\alpha(B - w)(P_1\sin\alpha + Q_1\cos\alpha)) \\ &\quad - \left(P_1^2 + Q_1^2 + (B - w)^2\cos^2\alpha\right) \end{aligned}$$

Doing some algebra, is possible to reach the following expression for r, where again the absolute value is taken because a distance must be positive.

$$r = \frac{-P_2 + \left|\sqrt{P_2^2 - 4Q_2\sin^2\alpha}\right|}{2\sin^2\alpha} \tag{19}$$

Thus, with respect to the ideal wheel

$$r_{id} \in \left(\max\left(R_{car}, r + \frac{w}{2}\right), R_{max} - \frac{w}{2}\right) \tag{20}$$

The graphic from the Fig13 can be useful for acquiring an idea of the restrictions found.

4. Bounding the allowed horizontal distances

As in the previous maneuver, in order to find the minimum and maximum horizontal distances, the maximum

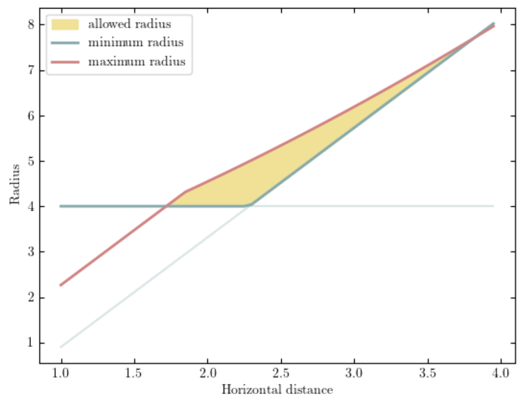

FIG. 13. Allowed radius in function of the horizontal distance from $O''(li)$.

and minimum radius are matched. So

$$\max\left(R_{car}, r+\frac{w}{2}\right) = R_{\max} - \frac{w}{2}$$

Namely

$$\max\left(R_{car}, r+\frac{w}{2}\right) = \min\left(R_{\max 1} - \frac{w}{2}, R_{\max 2} - \frac{w}{2}\right)$$

This implies solving a system of 4 non-linear equations that depend on li, but solving this equations analytically would be very complicated, and if possible, would end in huge unintelligible expressions. Therefore, is assumed that the equations system can be approximated with an arbitrary precision with some numerical method.

5. *Completion of the maneuver*

This maneuver finishes in the same way as the previous one, taking into account that, in the reference system, P_{fe} and P_{fi} are P_{fed} and P_{fid} respectively.

C. Iterative diagonal parking

In this last section of the diagonal parking is developed an algorithm that makes possible parking in slots where the two steps diagonal maneuver is not possible. The algorithm takes in account firstly that the horizontal position of the car is one such that the turn radius is not bigger than certain maximum radius. Then is needed that the vertical position makes possible that the trajectory does not intersect an obstacle at the vertical axis, such as a wall or the opposite lane of the road. Once in a good position, an iterative process is done until the vehicle is parallel to the parking slot, then is only left to go straight.

Algorithm 2 Iterative diagonal parking

$P_o = O'' + (li, hi)$
while $R - \frac{w}{2} > R_{car}$ **do**
  $li -=$ increment
end while
if $R_{car} < R_h$ **then**
  $P = P_o$
  $h'_i =$choose(hi)
  $P'_o = P_o + (0, w/2)$
  $P'_f = P'_o - (\sqrt{4 \cdot R_{car} \cdot (hi - hi') - (hi - hi')^2}, hi - hi')$
  $P =$parallel.maneuver(P_o, P'_f, R_{car})
  $P = P + (\sqrt{(}4 \cdot R_{car} \cdot (hi - hi') - (hi - hi')^2, 0)$
end if
$\beta = 0$
while $\beta < \alpha$ **do**
  while constraint1 **do**
    $(P,\ \beta)$= turn$(P,\ \beta,$ 'reverse')
  end while
  while constraint2 **do**
    $(P,\ \beta)$= turn$(P,\ \beta,$ 'forward')
  end while
end while
finish.maneuver(P, β)

where:

- *choose* is a function that, through the environment recognition sensor, finds the closest point to the initial one where $R_{car} > R_h$.
- *parallel.maneuver*(P_o, P_f, R) is a function that executes the maneuver in parallel with the given data
- *turn* is a function that makes a turn following the minimum turning radius of the vehicle and returns the final state and the angle of the car with respect to the horizontal component.
- *finish.maneuver* is a function that realize a straight trajectory until arrives to the final position.
- *constraint*1 := $(A \wedge B \wedge C)$, where
  - A(P) = $(P_y < \tan(\alpha)P_x) \vee (P_y > -(1/\tan(\alpha))$ $(P_x - b\cos(\alpha) - a\frac{\cos(\alpha)}{\tan(\alpha)}) + b\sin(\alpha) - a\cos(\alpha))$
  - P'= $P + w(-\sin(\alpha), \cos(\alpha))$ and B= A(P')
  - C = $(O'_y < -\frac{(O'_x - Px)}{\tan(\beta)} + P_y) \vee (P'_y < \tan(\alpha)P'_x$
- *constraint*2 :=

  $$\left(Pd'_y < h_{max}\right) \wedge (P_y > (\tan\alpha P_x - a\cos\alpha))$$

  where

  $$Pd' := P' + wl(\cos(\beta), \sin(\beta))$$

VIII. TRAILER

In this work an attempt has been made to implement a single trailer system with little success, but with some results that open the way for new work proposals on this subject. The following lines show the implemented idea.

Given the car-trailer system shown in figure (14)

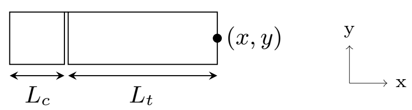

FIG. 14. Scheme of the considered car-trailer system.

The parameters L_c and L_t are the car and trailer length, respectively. Let $q = [x, y, \theta, \phi]^T$ the actual state of the system, where (x,y) is the point shown in figure (14) and θ, ϕ are the angle between trailer axis and horizontal axis, and the angle between trailer axis and car axis, respectively. Finally, let $[v, \alpha]^T$ the inputs, where v is the system velocity and α is the steering wheel angle. At this point, the trailer dynamics equations can be presented [5]:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \\ \theta_{n+1} \\ \phi_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \\ \theta_n \\ \phi_n \end{bmatrix} + v \begin{bmatrix} \cos(\theta_n) \\ \sin(\theta_n) \\ 0 \\ -\frac{\sin(\phi_n)}{L_t} \end{bmatrix} + \alpha \begin{bmatrix} 0 \\ 0 \\ 1 \\ -L_c \frac{\cos(\alpha_n)}{L_t} - 1 \end{bmatrix}$$

Note that these equations are described discreetly because is the best way to schedule them.

The first objective is to transport the initial point (x_0, y_0) to some final state, representing the final position, following the motion equations described. To implement this idea, the following algorithm has been proposed:

Algorithm 3 Trailer System Distance Minimization

$q_{ini} = [x_0, y_0, \theta_0, \phi_0]$ initial state
$q_{goal} = [x_f, y_f, \theta_f, \phi_f]$ goal state
while $q_{ini} \neq q_{goal}$ **do**
  $x(v, \alpha) = \text{motion.equations}(v, \alpha)_{1,1}$
  $y(v, \alpha) = \text{motion.equations}(v, \alpha)_{2,1}$
  $\text{Cont.Func}(v, \alpha) = \text{e.distance}((x(v, \alpha), y(v, \alpha)), (x_f, y_f))$
  $(v^*, \alpha^*) = \text{minimize}(\text{Dist}(v, \alpha))$
  $q_{i+1} = \text{motion.equations}(v^*, \alpha^*)$
end while

The basic idea is to find the (v, α) input that minimizes the distance between next position and final position. Being $q_{ini} = [40, 20, 0, 0]$ and $q_{goal} = [0, 0, 0, 0]$ this method generate the following path (figure (15)).

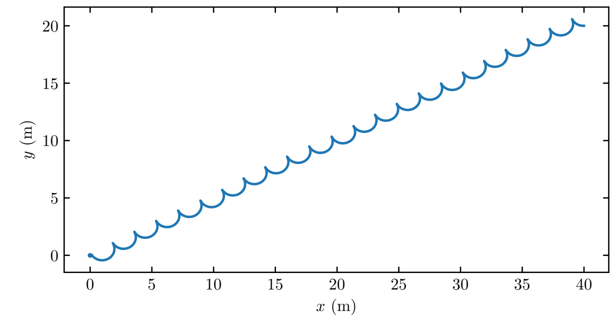

FIG. 15. Path generated using Algorithm 3: x-axis shows horizontal position and y-axis shows vertical position of the trailer.

From this example, it can be observed a few things to improve:

- The path generated is not the optimal one. It is necessary to add more restriction to the control function; in the Algorithm 1 has been considered only the euclidean distance.
- The goal state θ and ϕ angles has not been considered.

Considering the two points mentioned and testing with some new control functions, the generated path has been improved. This new control function involves θ and ϕ angles to take it into account the final orientation of the system. Also, it has been changed the scalar product and it has been adjusted manually. Finally, it obtains the following control function:

$$\text{Cont.Func}_2 = \langle q - q_{goal}, q - q_{goal} \rangle_A^{\frac{1}{2}} \tag{21}$$

where A is the matrix adjusted manually given a initial position.

IX. RESULTS

A. Possible configuration spaces

For each maneuver, it has been bounded the radii, vertical and horizontal distances. The final program needs this information to check if there exists a path without crashing.

From the parallel maneuver it has been obtained the following results:

$$\max\left(R_{min}, R_h, R\right) \leq R_1 \leq C - R_{min}$$

$$\max\left(s_1, \sqrt{\Delta y \, (4\, R_{min} - \Delta y)}\right) \leq a \leq s_2$$

where the first expression shows the bounded radius (extract from equation 3), and second one shows the bounded horizontal distance (extract from equation 4).

From the backward diagonal maneuver it has been obtained:

$$\max\left(R_{car}, R_h, r+\frac{w}{2}\right) \leq r_{id} \leq R-\frac{w}{2}$$

$$\max(li_{\min 1}, li_{\min 2}) \leq li \leq li_{\max}$$

extract from equations 7 and 10.

And finally, from the forward diagonal maneuver, similar results has been obtained:

$$\max\left(R_{car}, r+\frac{w}{2}\right) \leq r_{id} \leq R_{max}-\frac{w}{2}$$

extract from equation 20. It has not been calculated an analytical expression for the horizontal distance allowed.

B. LIDAR

The first parameter to test was the implemented LIDAR sensor. The map obtained from that program gives the information to the vehicle, so it requires a high precision for two reasons. Firstly, the vehicle is able to continue a given trajectory from the user as long as there are no immediate obstacle. Therefore, if the program does not detect some obstacle, it could end in a car accident. The second reason relates to the parking execution. The program needs to recognize the real slot in order to execute the correct maneuver.

The accuracy of the sensor was tested by making it perform a detection of a complex scenario for different cases. In particular, the studied magnitude is the number of laser emissions per millisecond, denoted by η. The experimental environment is generated by the function

$$f(x) = \frac{1.0}{\left|\cos\left(\frac{3.0\pi}{10.0}x\right)+0.3\right|} - \frac{1.0}{1.3}, \tag{22}$$

and it was created with the purpose of adding sharp regions in the obstacles. Even though it is not a common system, it is an academic exercise to test the performance of the sensor, since these shapes need a high precision to be detected. A sample of three possible values for η are represented in figure (16) with the original obstacle profile.

As one would expect, the higher η the better the precision of the sensor. A density of 20 laser emissions per millisecond leads to an inefficient performance of the program. The final polygon does not resemble the original obstacle and it presents a relative error 81.89 %. For that reason, this values could not be used for this program. The second value, $\eta = 80$, is able to generate most of the original shape, except for the spikes. Instead, each part has a rounded end that introduces an error of 13.74 %.

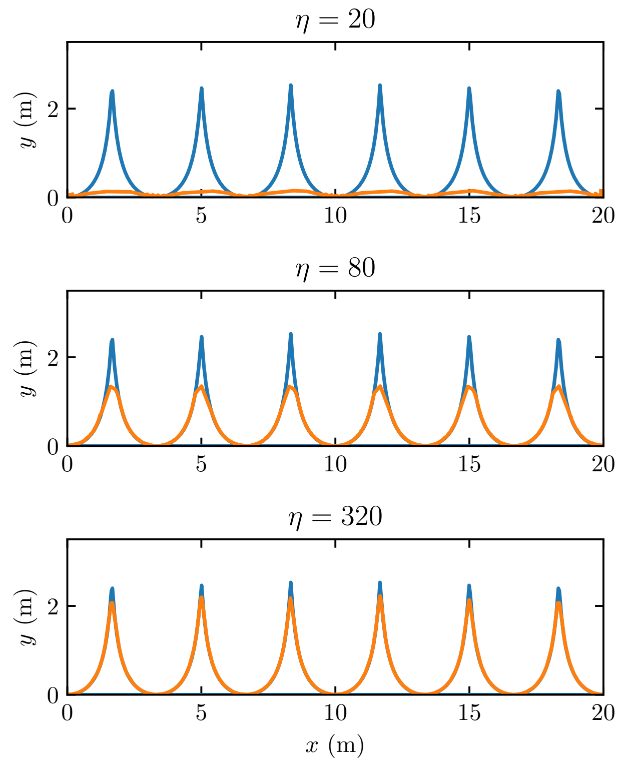

FIG. 16. Representation of the reproduced map from the original object, determined by the equation 22. The blue plot represents the environment and the orange plot the reconstruction, for values of $\eta = 20, 80, 320$.

Finally, the case of $\eta = 320$ reconstructs almost all the shape of the original obstacle. There is a slight difference in the peaks, but the error is reduced drastically to 1.20 %.

For the purpose of a global description of the LIDAR accuracy, the previous observations were completed with the figure 17. Now, the same procedure was completed for values of η from 20 to 400. As before, the error tends to decrease for higher densities. The pace of change is the greatest at the beginning, and later it converges more slowly to zero, with a minimum value of 0.62 %.

Apart from the precision, the computational efficiency should be considered too. A greater value of η not only implies a better resolution, but also implies a higher number of computations per iteration. After using the program for several densities of lasers, the velocity of execution slowed down highly for great values of η. In particular, when it reaches a density of 100, the program is so slow that it is no longer feasible to run at real time. As a result, it was decided to use a density of $\eta =$ lasers/ms.

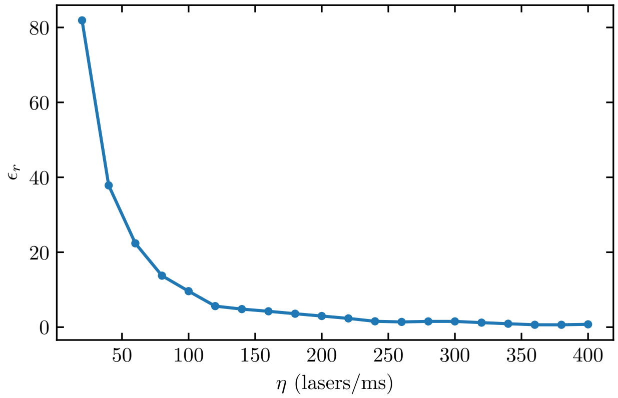

FIG. 17. Relative error of the LIDAR sensor as a function of the density of lasers per millisecond.

C. Car parking

With the density of the lasers decided, the vehicle was tested to park in a complete simulation. Starting from the left hand side, it moved forward at a constant velocity, which was introduced manually. For different parking configurations, it was able to generate a global map of the environment and recognize the type of slot available to park. Then, it executed the corresponding maneuver and stopped when it was completed. With that, it was concluded that the program performed as expected. The polygon representing the available space presented some discrepancy with the actual obstacles, but it was not enough to cause an error in the program. An example of these results is shown in the following link.

automatic-parking-system

D. Trailer

Finally, it has implemented an algorithm to park a trailer system with some positive results. This method is not the optimal one but it approaches nicely the real path of a trailer in the particular initial case $[40, 40, 0, 0]$. Applying (21) to the algorithm 3 it obtains the following figure (18), that it shows the comparison between the path generated with this method and the real optimal path extract from a MATLAB project. [5].

The difference between two paths can be compared in many ways and there are different methods to calculate this error. On this work it has been calculated the error between the two paths comparing the vertical position in each horizontal position (figure 19). It can be observed that the maximum value reaches 4.515 m.

However, there is some points to discuss:

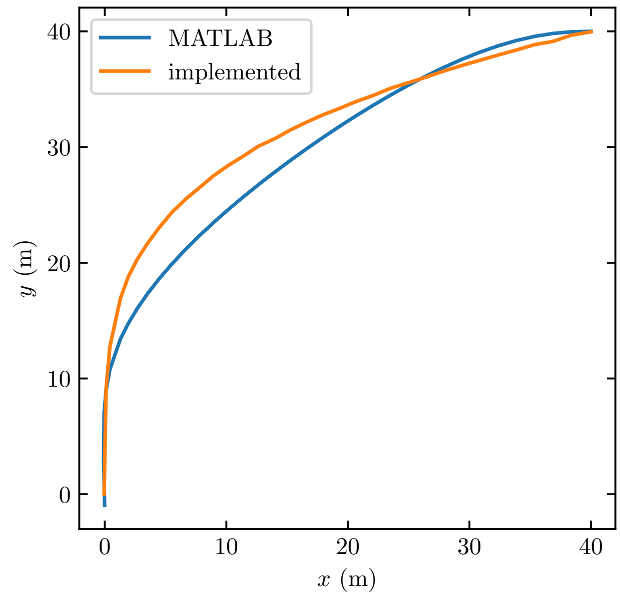

FIG. 18. Comparison between the implemented method (in orange) and the optimal real trailer path (in blue): x-axis shows horizontal position and y-axis shows vertical position of the trailer.

- This method requires to readjust manually the scalar product matrix in each q_{ini}, and it is not proved that there are always a matrix that approaches the path well.
- The final position seen in the generated path is not exact the q_{goal}.
- It has been found the way to transport the trailer system from q_{ini} to q_{goal} following the equations of motion. But the many obstacles that can appear in the environment, such as two more trailers delimiting the parking slot, has not been taken into account. It is needed some improved algorithm that can generate an alternative path avoiding this obstacles.

X. FURTHER WORK

This mathematical model can be improved in some aspects:

- It has been proved that there is no optimal radius to park in parallel, the traveled distance is always the same. But there is not a similar result to the diagonal maneuver. It is interesting to find some results about the optimal radius to minimize this distance.

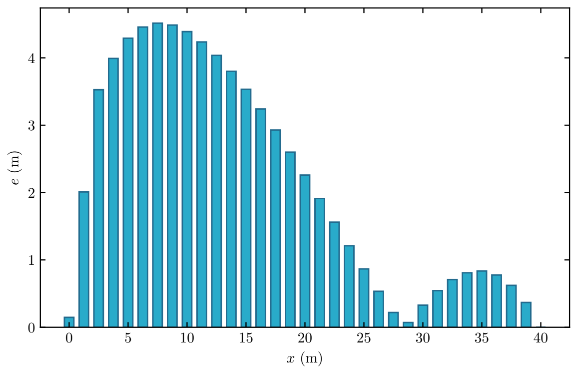

FIG. 19. Error between the second implemented method and the optimal real trailer path: x-axis shows horizontal position and y-axis shows the error.

- Apart from the distance optimization, other factors are involved. The time to complete the maneuver, the velocity of the car during the parking process and the oil consume are some aspects that could be taken into account. One should not only try to find the most efficient solution, but also consider for instance that people cannot tolerate all accelerations.
- This model has implemented the basic maneuvers to park and some iterative algorithms to expand the possible parking scenarios. But there exists a few cases that these maneuvers does are not feasible. A way to improve this model is designing new maneuvers or algorithms to solve these non-solved problems. For instance, when it exists a obstacle in the middle of the path and it is necessary to dodge it or surround it.
- The precision of the recognition program varies depending on the number of emitted lasers per millisecond. Higher densities of laser resulted in better resolution, but their computational cost made them inefficient. In order to solve this question, a different method to obtain the map could be implemented. Two approaches are proposed. on the one hand, instead of multiplying the density by 4, the vehicle could use for sensors, each one at a different corner of the vehicle. On the other hand, without increasing the number of laser emissions, the generation could be performed using a polynomial interpolation, since it could complete the shape more naturally.
- This work can be improved implementing a trailer in the car system. The algorithm that it has been used does not work well. It is believed that is necessary to implement a more complex method using sophisticated mathematics and programming techniques.

XI. CONCLUSIONS

In this work it has been successfully implemented a mathematical model to park a car totally automatic. To carry out this program, the three phases of the parking process have been studied: environment recognition, maneuver decision and maneuver design.

Firstly, an algorithm has been implemented so that the device is able to recognize its environment in each moment. To make this possible, a LIDAR sensor has been implemented. This allows to store an updated map of the real scenario in all times. Among all possible values of laser emission per millisecond, it was finally decided to use 80 laser/ms. Although the relative error of 13.74 % in the test, the final polygon reconstructed most part of the original obstacle.

Regarding the maneuver decision, as commented, the used density for lasers worked properly. From the gather information, the algorithm interpreted the polygon as the available space and compared it with all known configurations. In all cases it managed to find the corresponding maneuver of a particular slot and executed it as expected.

Finally, all the possible maneuvers that the program can execute has been designed. It has been studied the parallel (backward) and diagonal (backward and frontward) parking spots. In each one, given an initial a final position, it has bounded the allowed radius and horizontal and vertical distances. Also, it has been proved that there is not an optimal radius to choose in the parallel maneuver. And it has improved the diagonal maneuver applying a recursive algorithm, minimizing the horizontal distance mentioned.

To try to improve this model, an attempt to implement a trailer has been done. But with no successfully results developing a generic method, it has studied a particular case. Giving an initial position, it has changed manually the function cost to finally park it. And it has been compared with an optimal path to quantify the error made.

It can be concluded that the *APS: Automatic Parking System* is a device able to park completely automatic in multiple scenarios with good precision.

[1] Wikipedia, Ackermann steering geometry — Wikipedia, the free encyclopedia (2022).

[2] U. Stilla and B. Jutzi, Topographic laser ranging and scanning: Principles and processing (2008) pp. 215–234.

[3] F. Rohrbach, An introduction to lidar (2015), last accessed 20 May 2022.

[4] S. Gillies *et al.*, Shapely: manipulation and analysis of geometric objects (2007–).

[5] MATLAB, *Truck and Trailer Automatic Parking Using Multistage Nonlinear MPC* (The MathWorks Inc., Natick, Massachusetts, 2019-2020).