A RULE-BASED CONTROLLER SIMULATION FOR AN AUTONOMOUS PARALLEL PARKING OF A CAR-LIKE ROBOT USING LASER SENSORS

Kleber de Oliveira Andrade, pdjkleber@gmail.com
André Carmona Hernandes, andre.hernandes@usp.br
Marcelo Becker, becker@sc.usp.br
Mobile Robotics Lab – Center for Robotics of São Carlos – CRob - SEM – EESC - USP, São Carlos, Brazil

Abstract. *In the last few years, a plenty of tools and methods developed in the Mobile Robotics field have shown a great potencial for the automotive area. Due to this, the interest of researchers for applications in this subject has increased. Therefore, many researches have addressed the problem to embed smart systems in touring vehicles to perform daily tasks, such as urban driving, obstacle avoidance, active cruise control in heavy traffic situation, automatic parallel parking, etc. This work aims to develop a rule-based control system responsible for driver handling assistance, in the special case of reaching a robust control capable of doing a perfect automatic parallel parking. The studies involving the kinematics of the vehicle and the sensors modelling were previously carried out. Then, a simulation environment has been designed in C# language with the Visual C#2008 IDE in order to validate the finite state automaton algorithm. Based on the data of the front and rear laser sensors, the automatic mode of the simulation environment controls the vehicle by generating acceleration and steering commands. The manual mode returns the control of the vehicle to the user. So far, the simulations results have shown that a finite state automaton is capable of parallel parking the car, based on two sensors data.*

Keywords*: Finite State Machine, Autonomous Vehicle, Autonomous Drive and Park, Expert System, Virtual Reality*

1. INTRODUCTION

Today it is quite easy to find articles and news about robotics applications that few years ago we could just dream about. The mobile robotics field has evolved quickly last decades due to more powerful processors and sensors variety and availability (Dudex e Jenkin, 2000). This area aims to develop devices (robots) capable of autonomous act and interact with the environment and, especially, with humans. This was shown at different researches results that have been published last years. For instance, we may cite some application areas like space mining, service robots, agricultural activities, military services, education, and entertainment (Murphy, 2000). Due to recent technology breakthroughs, a sort of methods and tools developed in the mobile robotics area have shown a great potential of use in other areas, gathering attention from researchers of all around the world, especially in the automobile field. This technology has been developed based on two distinct approaches, the first one, an assistive system which can help drivers during traffic and share some information with the driver, and the other approach, the autonomous system, that is able to take decisions without human intervention and can change its goal to satisfy one determined function (Vlacic *et al.* 2001). Both approaches must be able to receive stimulus from the environment provided by sensors (e.g.: infrared, ultrasound, LIDAR, cameras, IMU, etc.) and plan their actions according to an objective. However, there are several challenges in using autonomous vehicles. One of them is the need of endowing them with an artificial intelligence capable of interact with the environment where they are inserted (Medeiros, 1998).

Based on this, the automotive industry has put a lot of effort on these new technologies, designing and installing electronic components in touring vehicles with the purpose of increasing the vehicle safety and users (driver and passengers) comfort. Hence, one of the greatest targets of the industry and the scientific community is the development of assistive systems which are capable of fulfill the design requirements (Lo *et al.*, 2003). We may cite as examples of the use of these technologies the Toyota Prius IPS (Intelligent Parking Assist), the Sedan Lincoln MKS and Ford Lincoln MKT parking systems and the Citröen C4 Picasso system, which shows if the car fits the parking space or not.

According to DENATRAN 2010 (Brazilian National Traffic Department) the vehicle fleet is increasing around 10% every year in Brazil. One should take into account that others developing countries have a similar increase in their fleets every year. In this scenario, the parallel parking may be considered as a big issue in urban traffic management. Based on this, this work presents the development of a rule-based algorithm capable of perform the perfect parallel parking. A finite automaton is used to control the velocity and steering angle during the maneuver, receiving data from the vehicle odometry system, a compass and 2 LIDAR (Light Detection and Ranging) sensors fixed at the extremities of the vehicle.

The organization of this paper is as follows: in section 2 we present a brief description of the problem and some related works found in literature. In the third section, the developed simulator is introduced, as well as the kinematic model and the simulated sensors. In section 4, the finite automaton is described. Then in the fifth section, the hypothesis used during the experimental phase and the results obtained are shown. And, finally, in section 6 we present the conclusions and propose future works.

2. PROBLEM STATEMENT

Knowing how to park quickly a car, for instance, at a city´s center, demands a certain experience level from the driver and instantaneous decision making to avoid traffic congestion. Due to this, this maneuver becomes a big challenge for inexperienced drivers. An autonomous smart system, instead of a human driver, could ease the traffic handling and increase the drivers' safety. Therefore, a variety of works found in literature has used artificial vision techniques (Daxwanger and Schmidt, 1996; Xu and Chen, 2000) and sonar sensors (Laugier and Fraichard, 1998; Jiang and Seneviratne, 1999) to search for an adequate parking space. For safety purposes, the length of this parking space has to be compared with the theoretical length calculated by Eq. 1 (Hermann, 2003).

$$p = \sqrt{2RW + (L-b)^2} \tag{1}$$

Where L is the vehicle length, b is the distance between the rear- axle and the rear bumper, W is the car width; R is the minimum radius of an arc of circumference performed by the vehicle and p is the minimum length required to park perfectly the vehicle without collisions using an S-shaped trajectory (Figure 1).

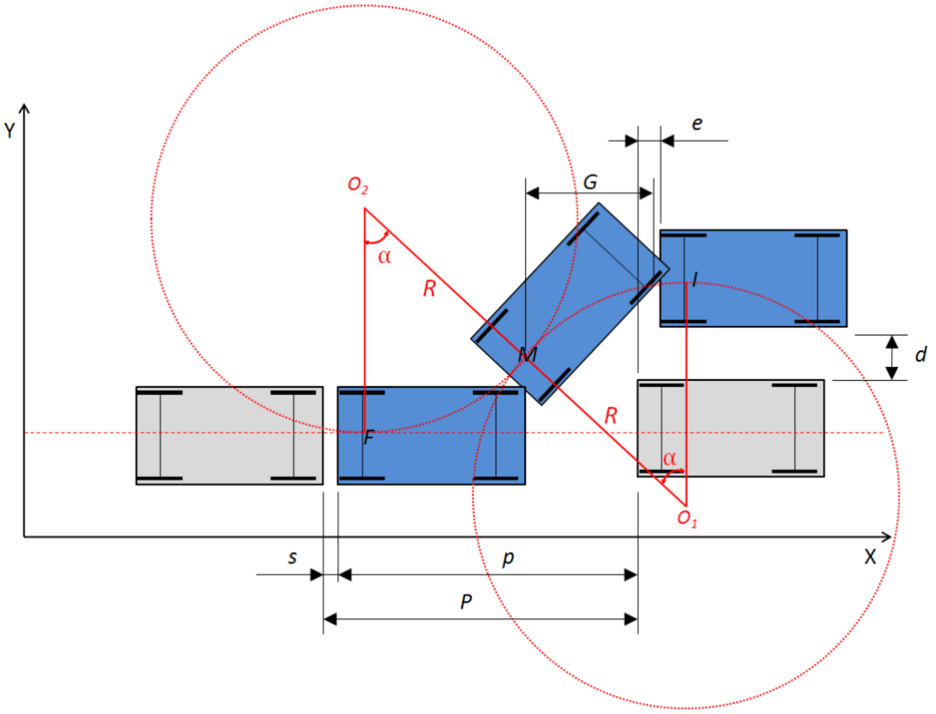


Figure 1. S-shaped trajectory of a parallel parking maneuver.

Once a parking space is found, the vehicle is positioned to perform the parking maneuver, which has two different approaches. The first one, the trajectory planning approach, needs a mapping (Hata and Wolf, 2009) and an auto-localization algorithms. In addition to this, it generates paths by using the dynamic and kinematics models of the vehicle and the vehicle constrains. Then, it controls the vehicle velocity and the steering angle to ensure a position in the mapped environment (Thrun *et al*. 2005). On the other hand, the second approach tries to mimic the skills of an experienced driver by using Artificial Intelligence techniques such as finite automaton (Heinen *et al.* 2006; Heinen *et al.* 2007; Osório *et al*. 2002), fuzzy logic (Ryu *et al.* 2008), artificial neural networks (Hernandes *et al*. 2010; Osório *et al*. 2002) or hybrid methods (Demirli and Khoshnejad, 2009) to solve the problem.

3. AUTONOMOUS VEHICLE SIMULATION

According to Law and Kelton (2000), the autonomous vehicles researches should be initiated in a virtual environment, due to the damages that a fail of the algorithm embedded in real vehicle could cause. The creation of a virtual simulation environment requires the development of sensors and actuators models as well as the proper physical behavior of the vehicle (Osorio, 2006).

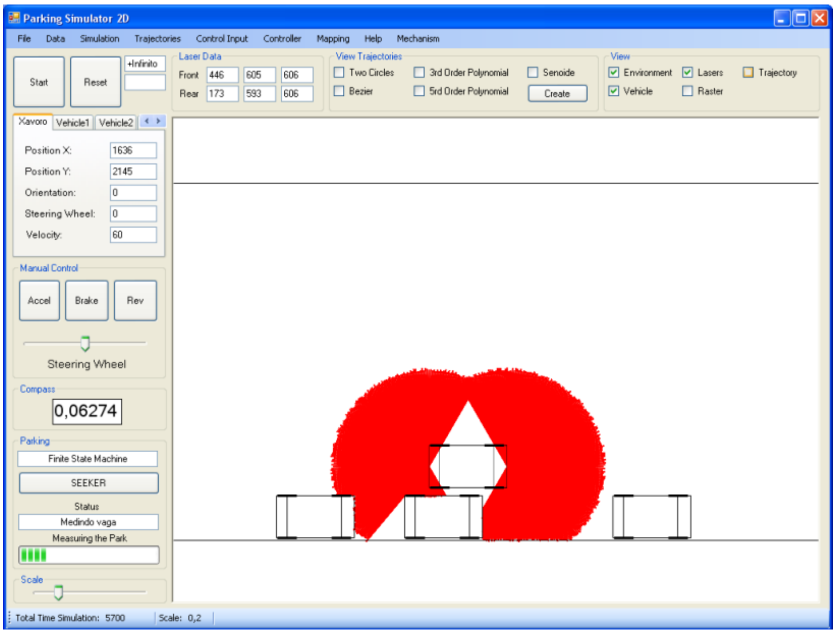
In the literature, one may found several proposed models that produce responses similar to real vehicle behaviors, including response errors to actuators commands (Dudek and Jenkin 2000; Siegwart and Nourbakhsh, 2004). Nowadays, it is possible to find in the web several simulation tools for mobile robots, some of them are users free, others you have to buy a license and a few that are sold along with the robot. Some examples of these simulators are: Player/Stage/Gazebo (Collett *et al.* 2005), CARMEN, ARIA/MobileSim, Microsoft Robotics Studio, Webots, SEVA 2D (Osório *et al.* 2002) and SEVA 3D (Heinen *et al.* 2007). In order to easily implement the algorithms developed, to use our vehicle models, and to use simulated sensor data as close as possible to the provided by the real sensors, we decided to develop our own simulator.

3.1. Simulator developed

Our simulator was developed exclusively for the parallel parking problem and it was based on C# language, which is an object-oriented language created by Microsoft along with the *.Net* architecture. In addition to this, the Integrated Development Environment (IDE) used was the Visual C# 2008 Express Edition. Basically, the features of this simulator are:

- It is a virtual environment simulator (the user may add/remove and place vehicles);
- It uses a kinematic model (based on the well known bicycle model);
- It uses accurate models for the embedded sensors: Hokuyo LIDAR, compass, encoder sensors;
- It has a vehicle manual displacement and steering system (the user can move manually the vehicle forward/backward, change its velocity, and rotate the steering wheel);
- It allows the autonomous control of the vehicle (it uses a finite automaton);
- And, it shows the sensors data and the performed trajectory.

Figure 2 shows the graphic interface of the developed simulator. At the left border, there is information of vehicles from the virtual environment, an initialize/reset simulation button, a vehicle manual control, a compass-based orientation information and an init button to start the autonomous control of the vehicle based on a chosen approach. The sensor and trajectories data and display information are located at the top border. The simulation appears at the main part.



+Infinito

Laser Data
Front 446 605 606
Rear 173 593 606

View Trajectories
☐ Two Circles
☐ 3rd Order Polynomial
☐ Senoide
☐ Bezier
☐ 5rd Order Polynomial

View
☑ Environment
☑ Lasers
☐ Trajectory
☑ Vehicle
☐ Raster

Position X: 1636
Position Y: 2145
Orientation: 0
Steering Wheel: 0
Velocity: 60

Compass
0,06274

Parking
Finite State Machine
Status
Medindo vaga
Measuring the Park

Scale

Total Time Simulation: 5700
Scale: 0,2

Figure 2. Simulator developed.

3.2. Kinematic vehicle model

During the simulation, the vehicle behavior needs to be as much as accurate when compared with the real world. Due to this, it was necessary to model the vehicle kinematics. However, the whole system modeling is very complex and laborious. So, some simplifier hypotheses were adopted to obtain a satisfactory vehicle behavior and a faster processing time:

1. **Rolling constrain**: The vehicle movement in the wheel plane must be equal to the movement performed by the spinning wheel.
2. **Sliding constrain:** The lateral wheel movement was discarded.
3. **Low-speed vehicles:** The vehicle dynamics can be neglected (Zhu and Rajamani, 2006).

Based on these hypotheses, the vehicle kinematic can be analyzed. According to Siegwart and Nourbakhsh (2004), in order to examine the kinematic constrains of a mobile robot, we just have to verify its own wheel constrains. So, the following equations are achieved:

$$\begin{cases} J_1 R(\theta)\dot{\varepsilon}_I - J_2\dot{\varphi} = 0 \\ \quad C_1 R(\theta)\dot{\varepsilon}_I = 0 \end{cases} \tag{2}$$

In Eq. (2)above, R(θ) represents the rotation matrix from the ground axis to the mobile robot axis. $\dot{\varepsilon}_I$ represents the velocity vector $[\dot{x}\ \dot{y}\ \dot{\theta}]^T$ at the ground axis. J_1 is the matrix with the rolling constrain. This matrix has N x 3 dimension, where N is the quantity of wheels. J_2 is an N x N diagonal matrix with the wheel radius. The vector $\dot{\varphi}$ represents the wheels angular speed. At last, C_1 is the matrix with the sliding constrains. This matrix has N x 3 dimension, where N is the quantity of wheels.

In addition to this, the Instantaneous Center of Rotation (ICR), which represents the circumference center described by the vehicle, depends only on its own restrictions, apart from the number of the wheels (Siegwart and Nourbakhsh, 2004). This consideration allow us to simplify the vehicle model, therefore, considering the rolling and sliding constrains and the low-speed constrain, in which the vehicle dynamic is irrelevant (Zhu and Rajamani, 2006), the vehicle model can be simplified to the bicycle model (Pinheiro, 2009), whose kinematic equations are well known (Kochem *et al.* 2002):

$$\begin{cases} \dot{x} = V cos\phi \\ \dot{y} = V sin\phi \\ \dot{\phi} = \left(\frac{\tan\theta}{l}\right) V \end{cases} \tag{3}$$

Where V is the vehicle longitudinal velocity, the vector $[\dot{x}\ \dot{y}\ \dot{\phi}]^T$ contains the vehicle linear and angular velocities at the inertial reference system, ϕ is the vehicle orientation measured from X axis of the inertial system, l is the distance between axles and θ is the wheel steering angle, considering the bicycle model, as one can see in Figure 3Figure **3**. The parameters details are in Table 1 – Table containing the vehicle parameters and its descriptions

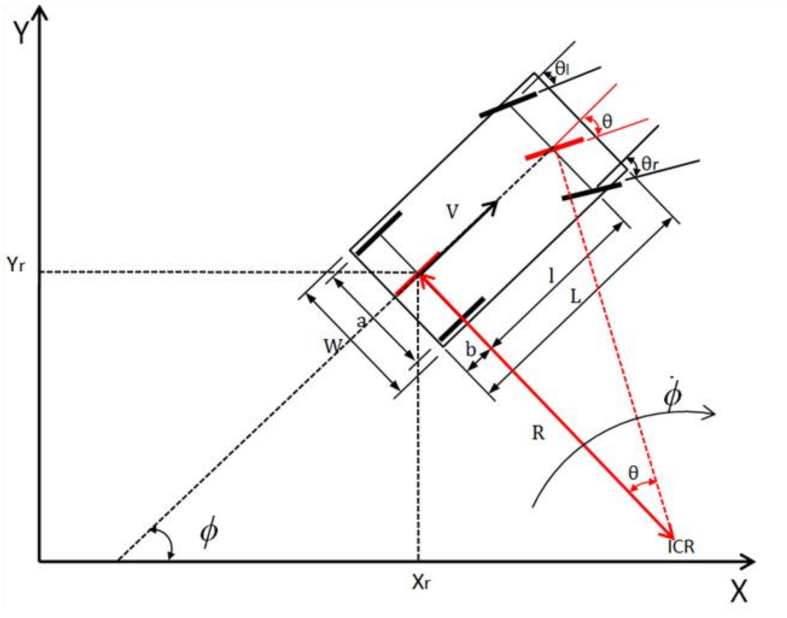


Figure 3. Kinematic model of the vehicle.

Table 1 – Table containing the vehicle parameters and its descriptions.

Symbol	Description	Symbol	Description
θ	Virtual front wheel steering angle	**A**	Distance between wheels
θ_l	Left wheel steering angle	**W**	Vehicle width
θ_r	Right wheel steering angle	**R**	Virtual rear wheel radius
l	Distance between axles	**ICR**	Instantaneous center of rotation
L	Vehicle length	**V**	Vehicle longitudinal velocity
b	Distance between the rear- axle and the rear bumper	**ϕ**	Vehicle orientation
$[x_r, y_r]$	Vehicle reference center at the inertial reference system	**[X, Y]**	Inertial reference system

Based on the kinematic equations shown in Eq. (3), we obtained the vehicle speed vector, which is a function of the actual position and the steering angle. Therefore, a following estimation can be carried out (Kochem *et al.* 2002):

$$\begin{cases} x(k+1) = x(k) + \dot{x}\Delta t \\ y(k+1) = y(k) + \dot{y}\Delta t \\ \phi(k+1) = \phi(k) + \dot{\phi}\Delta t \end{cases} \tag{4}$$

Where the vector $[x\ y\ \phi]^T$ represents the vehicle position at the ground reference system, k is the algorithm iteration counter, the vector $[\dot{x}\ \dot{y}\ \dot{\phi}]^T$ represents the vehicle velocities at the k instant, which is a function of the vehicle actual pose and its longitudinal velocity as seen before, at last, Δt represents the time interval between the k position and the $k + 1$ prediction. This time interval needs to be sufficiently small for a more similar update with the real environment, since the position algorithm presented in Eq. (4), which involves a multiplication between velocity and time interval, for great values of time, doesn't apply to the robot movements.

3.3. Sensors model

In the mobile robotics field, it is not enough only moving the vehicle in the environment. It is necessary to perform a complex set of tasks such as auto-localization, detection, classification, and obstacle avoidance, etc. Due to this, it is necessary to collect data from proprioceptive and exteroceptive sensors onboard the vehicle. A good choice for an exteroceptive sensor able to perceive objects around the vehicle are the use of Light Detection And Ranging (LIDAR) sensors. In this work, as we are using as test platform a scaled vehicle similar to a passenger car, two *Hokuyo*™ LIDAR sensors were used. These sensors have an angular range of 240° with 0.36° resolution and radial range from 20 mm to 4000mm with 1mm resolution and a 10Hz scan frequency. Moreover, the information of measure errors is important for a good sensor simulation. This information was obtained by checking the sensor datasheet provided by the manufacturer (ACRONAME, 2010). The graph presented in Fig. 4 shows the sensor standard deviation, which is a function of the measured distance.

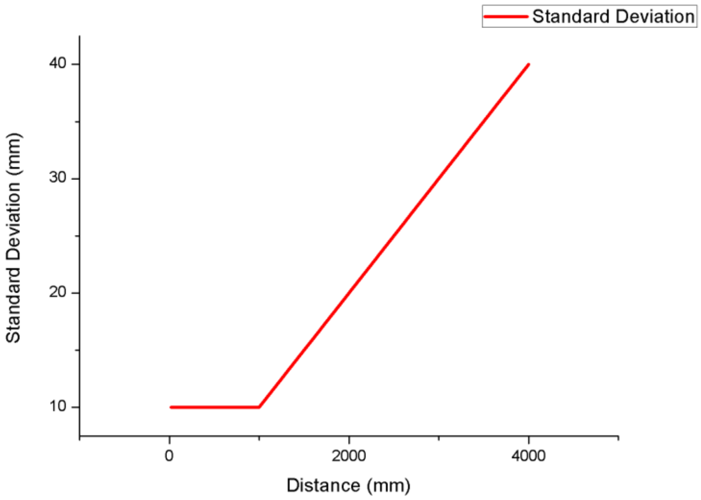
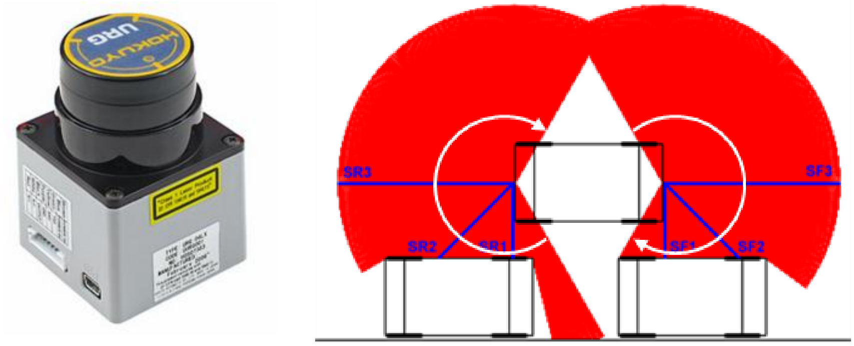


Figure 4. Standard Deviation x distance graph.

Therefore, defining $\mu_{measured}$ as the sensor theoretical distance and $\sigma(\mu_{measured})^2$ the sensor variance which is a function of the measured value, as shown in Fig. 4, the sensor noise could be interpreted as a Gaussian noise defined as $\mathcal{N}(\mu_{measured}, \sigma(\mu_{measured})^2)$. Combining all the information collected, we can incorporate the *Hokuyo*™ sensor in the simulator. The Fig. 5 shows the sensor image and its simulation.

The sensor data have a dimension of approximately 6700 distance values per second, however, handling a data set as long as this one just for the parallel parking problem increases the processing effort. Hence, significant angles for the parallel parking maneuver were chosen (Heinen *et al.* 2006) as shown in Figure **5** 5 (b). The angles represented by SR_1, SR_2, and SR_3 are related to the rear sensor, respectively, 30°, 75° and 120°. On the other hand, the angles SF_1, SF_2, and SF_3 are related to the front sensor, respectively, 210°, 165° and 120°. The senses of measures are represented by the white arrows. For proprioceptive data acquisition, an IMU sensor (Inertial Measurement Unit) is an interesting choice. It is not the cheapest one, but this sensor is one of the most complete sensors. In this work we used a *SBG IG-500*™ (Fig. 6) sensor which has 3 accelerometers, 3 gyroscopes, one compass and a GPS interface. For the parallel parking problem, only the compass data were used, for that matter, it was simulated. The compass inner standard deviation is

0.5°, and like the LIDAR sensor, a Gaussian noise model described by $\mathcal{N}(\theta_{measured}, \sigma^2)$ was applied, where $\theta_{measured}$ is the vehicle orientation and σ^2 the measured variance.



(a) (b)

Figure 5. Hokuyo URG -04LX sensor. (a) Sensor image. (b) Sensor positioning in the vehicle, inside the simulation environment.

Compass
0.06274

(a) (b)

Figure 6. IMU SBG IG-500 sensor. (a) Sensor image. (b) Virtual compass representation

4. RULE-BASED CONTROLLER

Parking a car is not easy, especially when you are a beginner level driver. Basically, in the parallel parking maneuver the driver needs to align the car between another two cars, making a parallel line with the curb. Despite the difficulties of young drivers, this maneuver is well-defined in 5 steps as shown at Fig. 7.

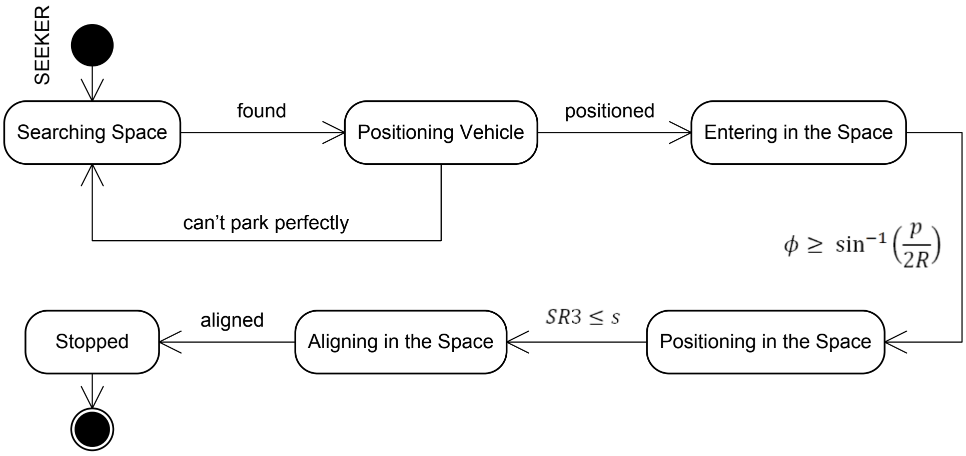


Figure 7. Finite Automaton for parallel parking maneuvers.

The autonomous parking control starts when the SEEKER button is pressed in the simulator. The first step is the search for a space large enough so the car can park. In this phase, its actions are a forward controlled speed and the verification of the SF_1 sensor values. If this value is bigger than $d_{min} + W$, it triggers a counter using the car odometry data trying to identify if a valid parking space has been found. When the measured distance is smaller than $d_{min} + W$, it means that an object was found. Than the counter stops and its value is compared with Eq. (1) in order to decide if the length is sufficiently large to park the vehicle. When that space is found, the second step is triggered. It acts on the vehicle to keep a forward speed until the SR_1 detects the obstacle responsible for stopping the counter. When that happens, the vehicle stops. Now the vehicle is fully positioned and it is assumed that the vehicle reference system is at position (0, 0). Based on this, the radius R can be estimated as follows:

$$\begin{cases} I = (0,0) \\ F = (-p, -(W+d)) \\ M = \left(-\frac{p}{2}, -\frac{W+d}{2}\right) \\ O_I = \left(0, -\frac{(W+d)^2 + p^2}{4(W+d)}\right) \\ O_F = \left(-p, -\frac{3(W+d)^2 - p^2}{4(W+d)}\right) \\ R = R_I = R_F = \left(\frac{(W+d)^2 + p^2}{4(W+d)}\right) \end{cases} \tag{5}$$

Where I is the vehicle initial position, F is the vehicle final position inside the parking space, O_I e O_F are the center of its respective circumferences, M is the tangent point of both circumferences and R is the vehicle turning radius to get in the parking space. At this point, if R is smaller than R_{min} performed by the vehicle, so it cannot park or even if it can, a free obstacle collision path must be done. These analyses have to be carried out before the vehicle starts the maneuver, so it can decide if it is allowed to get in the parking space or if it will search for another place to park.

When the vehicle is allowed to start the maneuver, the steering wheel has to be completely turned towards the curb until $\theta = \tan^{-1}\left(\frac{l}{R}\right)$ and then the vehicle begins to slowly move backwards. When $\phi \geq \sin^{-1}\left(\frac{p}{2R}\right)$, the vehicle stops near M, and so, the finite automaton goes to the fourth step in which the steering wheel is turned completely to the other side and keep the vehicle moving backwards until SR_3 read a distance value equal or smaller than a safety distance s. Next, the steering wheel is aligned and the vehicle move forward until it reaches the center of the parking space, verified by the difference between the SF_3 and SR_3 sensors. When the vehicle has reached the center of the parking space, it stops and the perfect parallel parking maneuver was finished.

5. EXPERIMENTS AND RESULTS

Some experiments were designed and performed in the simulation environment in order to validate the developed control strategy. As we are using a scaled vehicle to carries out the real tests, Table 2 shows the vehicle parameters and Fig. 8 shows pictures of the vehicle.

Table 2. Scaled vehicle specification parameters.

Symbol	Value
θ	-30º / 30º
W	260 mm
R_{min}	580 mm
l	335 mm
L	480 mm
b	65 mm

During the simulations, the parameters d, distance between vehicles, and p, size of parking space, were modified to test the control. Parameter d range was between 15% and 40% of vehicle width W, while p was between the minimum required length as shown at Eq. (1) and $2L$, where L is the length of the vehicle. To ensure no damage on the vehicle, we used a safety margin s of 10% of L. This procedure allowed us to generate all the curves and check if they were a collision free path. So, it allowed the vehicle to perform the perfect parallel parking using an S-shaped trajectory. Figure 9 shows the graph with the results of this test. As presented in Fig. 9, the scaled-vehicle cannot perform the perfect parallel parking maneuver considering only the minimum size required demonstrated by Eq. (1). We noticed that only from $p = p_{min} + 110\ mm$ it is possible to park the vehicle. If we have bigger parking spaces, farther you can get from the other vehicle to start the maneuver. Figure 10 shows an example of parallel parking maneuver; during this maneuver the wheel trajectories, the sensor data and even the two circles responsible to form the S-shaped path can be visualized.

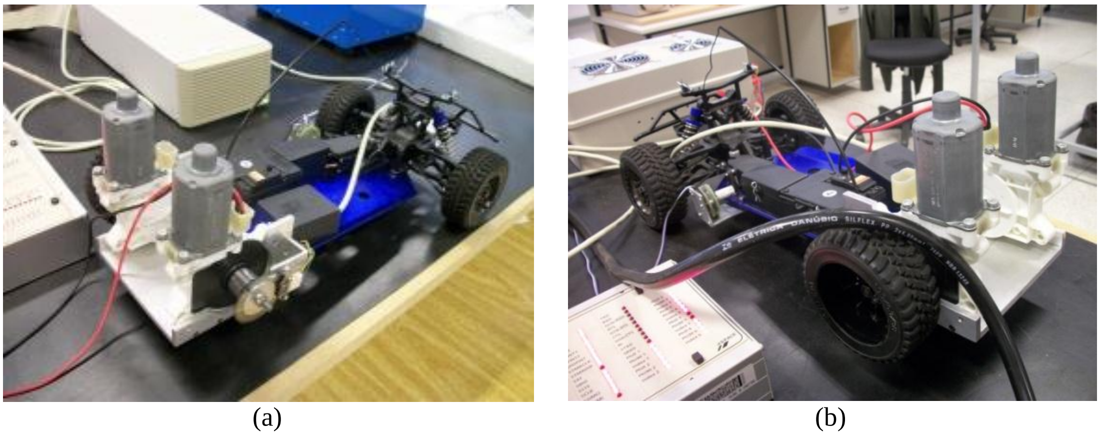


Figure 8. Scaled vehicle. In (a) detail of the encoders and in (b), other car view

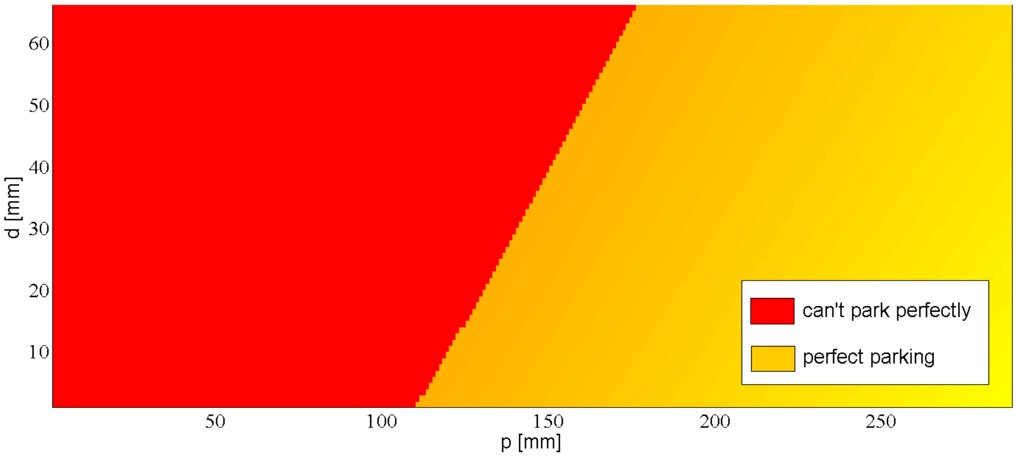
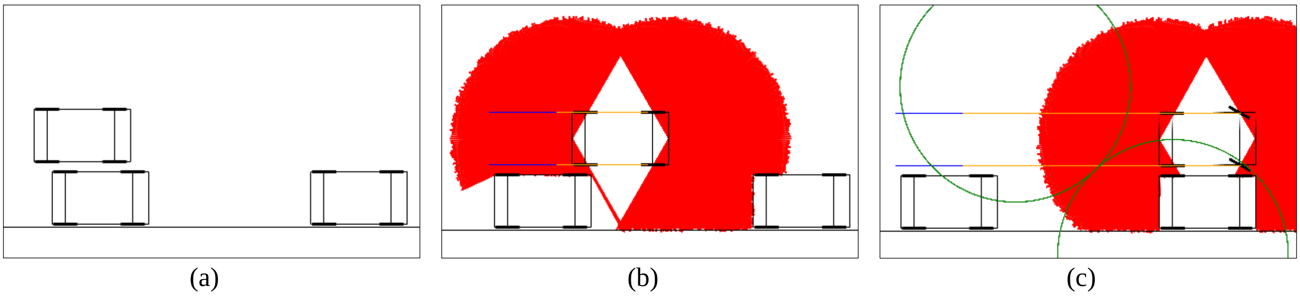


Figure 9. Graph showing the influence of parameters *p* and *d* on the perfect parking maneuver of our scaled vehicle.

In Fig. 10-a one may see that the vehicle is in Step 1, searching for a parking space, in Fig. 10-b, the LIDAR beams and the performed path are shown for the same procedure as in Fig. 10-a. In Fig 10-c, one may see the end of the Step 2, where the vehicle is fully positioned to perform the parking maneuver; one also can see the S-shaped path for the parking space. In Fig. 10-d, one may see the end of Step 3, where the vehicle approaches the M point. In Fig. 10-e, one can see the beginning of the Step 4, the images from the vehicle entering the parking space and the end of Step 4 can be seen in Fig. 10-f and Fig. 10-g, respectively. In Fig. 10-h, one may see the final position of the vehicle, as it can be noticed; it is fully positioned inside the parking space, and in Fig. 10-i one may see the path performed by the vehicle.



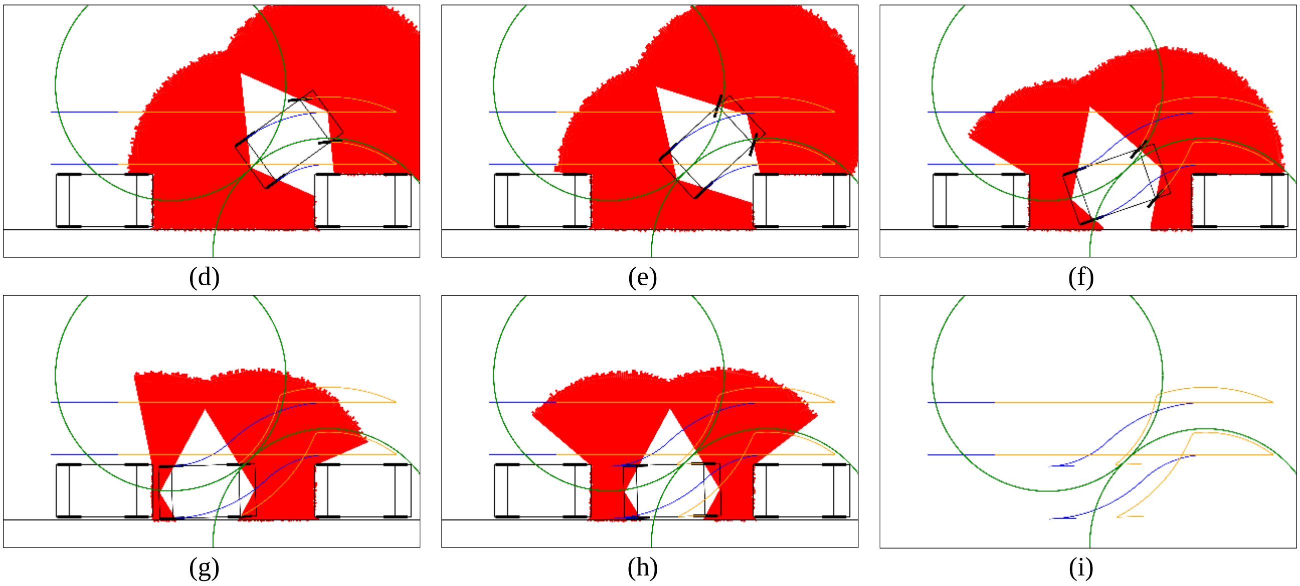

(d) (e) (f)
(g) (h) (i)

Figure 10. Sequence of pictures presenting an example of the maneuver generated by the finite automaton.

6. CONCLUSIONS AND FUTURE WORKS

This work aimed to develop a rule-based control so a vehicle is able to perform autonomously the parallel parking maneuver. The simulated results showed that the finite automaton control system developed can correctly control the vehicle, performing a perfect parallel parking without object collisions.

Nowadays, the real scaled vehicle, which has been used as a model for the simulated vehicle, has been equipped with the sensors. So, the developed control will be validated in the real experiment next months. Others techniques also have been studied, such as Artificial Neural Networks (ANN), fuzzy logic and hybrid methods, and their tests are still in progress. New tests on the developed system will be carried out as soon as these new approaches have been implemented. The trajectory planning approach will be studied so the control can become even more robust.

7. ACKNOWLEDGEMENTS

The authors would like to acknowledge the financial support of the Brazilian agencies CNPq, CAPES, and FAPESP and the INCT – INOF (National Institute for Optics and Photonics).

8. REFERENCES

ACRONOME, 2010. “Hokuyi YRG-04LX Laser”, Acroname Robotics. 20 November 2010. http://www.acroname.com/robotics/parts/R283-HOKUYO-LASER1.html

Collett, T.H.J., MacDonald, B.A., Gerkey, B.P. 2005. "Player 2.0: Toward a Practical Robot Programming Framework". In Proceedings of the Australasian Conference on Robotics and Automation (ACRA 2005), Sydney, Australia.

Daxwanger, W.A., Schmidt, G.K., 1996. “Skillbased visual parking control using neural and fuzzy networks,” Proc. of IEEE Systems, Man and Cybernetics, Vol. 2, pp. 1659-1664.

Demirli, K., Khoshnejad, M., 2009. “Autonomous parallel parking of a car-like mobile robot by a neuro-fuzzy sensor-based controller, Fuzzy Sets and Systems”, Vol.160, n.19, pp.2876-2891.

DENATRAN, 2010. “Frota de veículos”. Departamento Nacional de Transito. 10 February 2010 http://www.denatran.gov.br/frota.htm

Dudek, G., Jenkin, M., 2000. “Computational principles of mobile robotics”. Cambridge University Press, New York, NY, USA.

Hata, A.Y., Wolf, D.F., 2009. “Mapeamento de terrenos utilizando robôs móveis e sensor lase”. In: Latin American Informatics Conference (CLEI), Pelotas – RS: Universidade Federal de Pelotas (UFPel), pp. 49.

Heinen, F.J., Heinen, M.R., Osório, F.S., Kelber, C., 2006. “Uso de Realidade Virtual no Desenvolvimento de um Sistema de Controle do Estacionamento de Veículos Autônomos”. [SVR] Anais do Simpósio Brasileiro de Realidade Virrtual.

Heinen, M.R., Osório, F.S., Heinen, F.J., Kelber, C., 2007. “SEVA3D: Autonomous Vehicles Parking Simulator in a Three-Dimensional Environment”. In: INFOCOMP Journal of Computer Science, Vol. 6, pp. 63-70.

Hernandes, A.C., Mendonça, A.S.B., Andrade, K.O., Becker, M., 2010. “Redes neurais artificiais aplicadas à baliza paralela de veículo de passeio em ambiente simulado”. In: CONEM 2010 - Congresso Nacional de Engenharia Mecânica, Campina Grande - PB. Anais do CONEM 2010, 2010. Vol. 1. pp. 1-10.

Herrmann, N., 2003. “Ein mathematisches Modell zum Parallelparken”.

Jiang, K., Seneviratne, L.D., 1999. “A sensor guided autonomous parking system for nonholonomic mobile robot,” Proc. of IEEE Int. Conf. Robotics & Automation, Vol. 1, pp. 311-316.

Kochem, M., Neddenriep, R., Isermann, R. 2002. “Accurate local vehicle dead-reckoning for a parking assistance system”. In: Proceedings of the American Control Conference, pp. 6. Anchorage.

Laugier, C., Fraichard, T., 1998. “Sensor-based control architecture for a car-like vehicle,” Intelligent Robots and Systems, Vol. 1, pp. 216-222.

Law, A. Kelton, W., 2000. “Simulation Modeling and Analysis”. McGraw-Hill, USA.

Lo, Y.K., Rad, A.B., Wong, C.W. and Ho, M.L., 2003. “Automatic Parallel Parking”. In: Proceedings of the 2003 IEEE Intelligent Transportation System, Vol. 2, pp. 1190-1193

Medeiros, A., 1998. “Introdução a robótica”. In: Anais do XVII Encontro Nacional de Automática (50ª reunião annual da Sociedade Brasileira para o Progresso da Ciência – SBPC), Vol.1, Natal, RN, Brazil, pp. 56-65.

Murphy, R.R., 2000. “Introduction to AI Robotics”. MIT Press, Cambridge, MA, USA.

Osório, F.S., Heinen, F.J., Fortes, L., 2002. “Autonomous vehicle parking using finite state automata learned by J-CC artificial neural nets”. In: Proceedings of the VI Brazilian Symposium on Neural Networks (SBRN), Vol. 1, Porto de Galinhas, PE, Brazil. IEEE Press.

Pinheiro, B. C. 2009. “Sistema de controle tempo real embarcado para automação de manobra de estacionamento”. Florianópolis.

Ryu, Y.-W., Oh, S.-Y., Kim, S.-Y., 2008. “Robust automatic parking without odometry using an evolutionary fuzzy logic controller”. Int. J. of Control, Automation, and Systems, Vol. 6, No. 3, pp.434-443.

Siegwart, R., Nourbakhsh, I. R. 2004. “Introduction to Autonomous Mobile Robotics”. The MIT Press.

Thrun, S., Burgard, W., Fox, D., 2005. “Probabilistic robotics”. MIT Press.

Vlacic, L., Parent, M., Harashima, F., 2001. “Intelligent Vehicle Technologies: Theory and Applications”. Butterworth-Heinemann, Oxford, United Kingdom.

Xu, J., Chen, G., 2000. “Vision-Guided Automatic Parking for Smart car,” Proc. of IEEE Intelligent Vehicle Symposium, pp. 725-730.

Zhu, C., and Rajamani, R. 2006. “Global positioning system-based vehicle control for automated parking”.

9. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.