

Autonomous Parking Using A Sensor Based Approach*

David Pérez-Morales¹, Salvador Domínguez-Quijada¹, Olivier Kermorgant¹ and Philippe Martinet¹

Abstract—This paper considers the perpendicular parking problem of car-like vehicles for both forward and reverse maneuvers. A sensor based controller with a weighted control scheme is proposed and is compared with a state of the art path planning approach. The perception problem is threatened as well considering a Velodyne VLP-16 as the sensor providing the required exteroceptive information. A methodology to extract the necessary information for both approaches from the sensor data is presented. A fast prototyping environment has been used to develop the parking application, and also used as simulator in order to validate the approach. Preliminary results from simulation and real experimentation show the effectiveness of the proposed approach.

I. INTRODUCTION

Parking is often a difficult task, especially for unexperienced drivers. Starting with the problem of having to find a suitable parking spot, task that is often far from being trivial in popular places, to then start maneuvering into the, frequently tight, spot without colliding with anything (or anyone) while trying avoid disturbing the surrounding traffic. All of this hassle could be avoided with an Autonomous Parking system.

The most complete solution for the parking problem, and the dream of many, would be to be able to, once the driver arrives to the desired destination, leave the car at the entrance of the building or at the entrance of a dedicated parking lot (if any) and forget about it, letting the car park by itself safely and autonomously in an available (parallel, perpendicular or diagonal) parking spot and, whenever the user wants leave, be able to summon the car remotely to go to the driver's location and pick him up.

Currently available parking assistance systems, using mainly cameras and ultrasonic sensors to perceive the environment, have lead to a significant increase of driving comfort and provided cost savings from (avoidance of) accidents [1].

As it can be seen in [2], the literature related to automatic parking is quite extensive, having many different control approaches available. These different control approaches can

be divided into two categories [3]: one based on stabilizing the vehicle to a target point, with fuzzy control being the most widely investigated from this category and with one of the earliest works on automatic parking [4] using sinusoidal control functions; the other one based on path planning, where geometric collision-free path planning approaches based on the non-holonomic kinematic model of a vehicle have been of special interest in recent years [5], [6], [7].

It is also worth to notice that maneuvers contemplating parking with a forward motion are barely studied, with [7] for the parallel parking case and [8] for the perpendicular case being some of the few works on this regard.

Regarding the use of a sensor based approach for controlling autonomous vehicles, it has been proven to be valid for navigation [9] and dynamic obstacles avoidance [10].

The novelty of this paper is the implementation of a sensor based control with a weighted control scheme to perform perpendicular parking maneuvers. With the proposed technique it is possible to perform the parking maneuver with either reverse or forward motion with only some minor changes depending on the desired direction of motion. Furthermore, as far as we know, this is one of the firsts works that uses a sensor based control approach to solve an automatic parking problem.

The paper is organized as follows. In section II the model considered as well as the notations used are presented, followed by the sensor based control with a weighted control scheme. In section III the perception problems are described: how to extract the empty parking place and its localization in order to be able to perform the parking maneuver through a state of the art path planning approach and how to extract the current features for performing the maneuver using sensor based control approach. The fast prototyping environment and the results obtained under simulation for the state of the art and the sensor based proposed control approaches considered and from real experimentation for the sensor based approach are shown in IV. Finally, some conclusions are given in section V.

II. MODELING AND CONTROL

Since the parking maneuvers are always performed at low velocities, a kinematic model can be considered as accurate enough.

A. Model and notation

The kinematic model considered is the one used to represent a car with rear-wheel driving:

*This work was supported by the Mexican National Council for Science and Technology (CONACYT), Baja California State Council for Science and Technological Innovation (COCIT BC) and the Mexican Ministry of Public Education (SEP). This paper describes work carried out in the framework of the Valet project, reference ANR-15-CE22-0013-02.

¹ David Pérez-Morales, Salvador Domínguez-Quijada, Olivier Kermorgant and Philippe Martinet are with IRCCyN, Institut de Recherche en Communications et Cybernétique de Nantes, École Centrale de Nantes, 1 rue de la Noë, 44321 Nantes, France

¹ David.Perez-Morales@eleves.ec-nantes.fr

¹ Salvador.Dominguez@irccyn.ec-nantes.fr

¹ Olivier.Kermorgant@irccyn.ec-nantes.fr

¹ Philippe.Martinet@irccyn.ec-nantes.fr

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ \tan \phi / l_{wb} \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \dot{\phi}, \quad (1)$$

where v and $\dot{\phi}$ are the longitudinal and steering velocities. Table I presents the different parameters used in the paper.

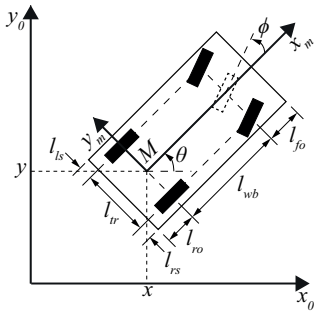
TABLE I: Parameters definition

Parameters	Notation	Value
Wheelbase: Distance between the front and rear wheel axles	l_{wb}	1.87 m
Track: Distance between right and left wheels	l_{tr}	1.03 m
Front overhang: Distance between the front wheel axle and the front bumper	l_{fo}	0.413 m
Rear overhang: Distance between the rear wheel axle and the rear bumper	l_{ro}	0.657 m
Distance from the left wheel to the left side of the vehicle	l_{ls}	0.115 m
Distance from the right wheel to the right side of the vehicle	l_{rs}	0.115 m
Maximum steering angle	ϕ_{max}	28°
Total length of the vehicle	l_{ve}	2.94 m
Total width of the vehicle	w_{ve}	1.26 m
Maximum (desired) longitudinal velocity	$ v_{max} $	2 km/h

The point M is located at the mid-distance between the passive fixed wheels (rear) axle and the distance between the rear and the front axle is described by l_{wb} . The generalized coordinates are $q = [x, y, \theta, \phi]^T$ where x and y are the cartesian coordinates of the point M, θ is the orientation of the platform with respect to the x_0 axis and the steering angle of the steerable wheel(s) is described by ϕ (Fig. 1a).

From the kinematic model it is possible to extract the following relation between ϕ and $\dot{\theta}$:

$$\phi = \text{atan}\left(\frac{\dot{\theta} l_{wb}}{v}\right) \quad (2)$$



(a) Kinematic model diagram



(b) HDK DEL2030DUB

Fig. 1: Kinematic model diagram for a car-like rear-wheel driving robot and vehicle used for experimentation and simulation

The vehicle used for experimentation and simulation is a HDK DEL2030DUB (Fig. 1b). It is represented by its bounding rectangle. The front track has been approximated to have the same length as the rear one.

B. Sensor based control with a weighted control scheme

A novel sensor based control technique is proposed. It is based on the framework described in [11] with the main difference being the fact that we consider $\mathbf{v}_m = [v \ \dot{\theta}]^T$ as the control signal instead of the robot joints velocities denoted by $\dot{\mathbf{q}}$. Taking this modification in consideration, we now proceed to briefly present the sensor based control with a weighted control scheme used.

1) *Kinematic model*: Let us consider a robotic system equipped with k sensors that provide data about the robot pose in its environment. Each sensor S_i gives a signal (sensor feature) s_i of dimension d_i with $\sum_{i=1}^k d_i = d$.

We consider a reference frame \mathcal{F}_m in which the robot velocity can be controlled. In our case the frame \mathcal{F}_m would be attached to vehicle's rear axis with origin at the M point. The sensor measurements are already on \mathcal{F}_m .

Considering the assumption that the vehicle to which the sensors are attached to evolves in a plane and that, given the kinematic model of a car-like robot, it does not exist a velocity along the y_m on the vehicle's frame, the sensor feature derivative can be expressed as follows:

$$\dot{\mathbf{s}}_i = \mathbf{L}_i \mathbf{v}_i = \mathbf{L}_i \mathbf{v}_m, \quad (3)$$

where \mathbf{L}_i is the interaction matrix of s_i and has a dimension $d_i \times 2$ and v_i represents the sensor velocity.

Denoting $\mathbf{s} = (s_1, \dots, s_k)$ the d -dimensional signal of the multi-sensor system, the signal variation over time can be linked to the moving vehicle velocity:

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v}_m \quad (4)$$

where \mathbf{L}_s is obtained by vertically concatenating the interaction matrices $(\mathbf{L}_1, \dots, \mathbf{L}_k)$

2) *Weighted control scheme*: Having the sensor signal error defined as $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$, where \mathbf{s}^* is the desired value at equilibrium of the signal \mathbf{s} , we consider the weighted error technique as described in [11] that allows to ensure specific constraints by establishing a safe interval for s_i .

The control law takes the following form:

$$\mathbf{v}_m = -\lambda \mathbf{C} \mathbf{e}, \quad (5)$$

where $\mathbf{C} = (\mathbf{H} \hat{\mathbf{L}}_s)^+ \mathbf{H}$, λ is a diagonal positive semi-definite gain matrix and \mathbf{H} is a diagonal positive semi-definite weighting matrix that depends on the current value of \mathbf{s} .

Since we cannot control directly $\dot{\theta}$, we use (2) to calculate the steering angle that would allow the system to have the desired $\dot{\theta}$. Then, since we want to have a smooth acceleration at the beginning of the parking maneuver, we impose the following velocity profile:

$$\begin{aligned} &\text{if } v < -|v_{max}|(1 - \exp(-t\tau)) \\ &\quad v = -|v_{max}|(1 - \exp(-t\tau)) \\ &\text{else if } v > |v_{max}|(1 - \exp(-t\tau)) \\ &\quad v = |v_{max}|(1 - \exp(-t\tau)) \\ &\text{else} \\ &\quad v = v \end{aligned} \quad (6)$$

where τ is a positive constant.

III. PERCEPTION

The vehicle used (Fig. 1b) has been equipped with many sensors (Velodyne VLP-16, GPS, cameras in the front, etc.) to observe its environment, a computer to process the data and actuators that can be computer controlled. Since our application requires exteroceptive information from all around the vehicle, the VLP-16 was the sensor chosen to work with.

The point cloud provided by the sensor is first filtered with a crop box in order to keep only the data that is close enough to the car to be relevant in a parking application and that does not represent the floor and afterwards it is downsampled in order to reduce the computation time of the rest of the steps. Then, an Euclidean Cluster Extraction algorithm is used to have each obstacle represented as a cluster. Afterwards, the orientation of each cluster is extracted using the following approach:

```

if cluster_size > threshold then
    project the cluster into the ground plane and extract
    its concave hull;
    extract the orientation of the hull's main line;
else
    find the main vertical plane of the cluster;
    extract the orientation of the vertical plane;
end

```

Algorithm 1: How to find the obstacle orientation

where *threshold* is a threshold value of the number of points a cluster is composed of and *cluster_size* is the number of points in the cluster.

The orientation of the bounding box will be equal to the orientation of either the hull's main line or of the vertical plane. After, we proceed by finding the rotated bounding box of the cluster with the previously found orientation.

Cars when viewed from the top have a rectangular-like shape, for this reason it is acceptable to approximate an obstacle's (parked vehicle) size, position and orientation by the bounding box of its pointcloud cluster.

A. Extraction of empty parking place

In order to be able to perform the parking maneuver with the considered state of the art path planning approach [6], it is necessary to know the location and characteristics of the empty parking place.

From the obstacles' bounding boxes it is possible to extract the empty parking place. To exemplify our approach we will consider two obstacles, represented in red in Fig 2a. First it is necessary to find the two minimum distances between the points defined by the corners of the obstacles with the constraint that the four points that define the two distances have to be different.

In Fig. 2a the two minimum distances are d_1 , defined by c_{12} and c_{23} , and d_2 , defined by c_{11} and c_{24} . Then, we can find the midpoints between the points that define the two minimum distances and, with these two midpoints, we can

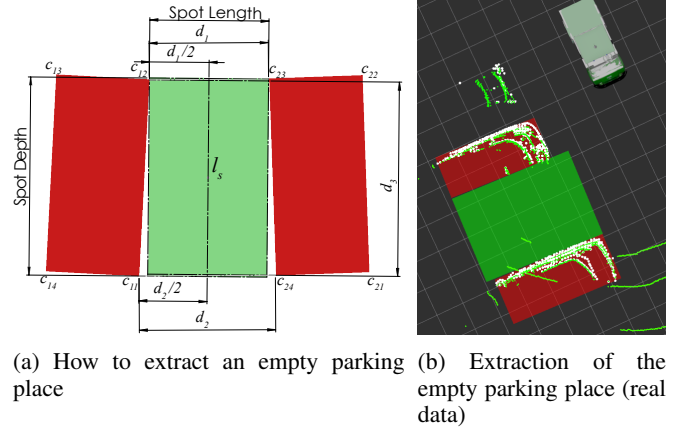


Fig. 2: Obstacles in red, empty parking place in green

construct a line l_s that allows us to extract the parking spot length by adding up the two minimum distances between the line and the points that define d_1 and d_2 , with one point of these new distances on each side of the line l_s . To extract the spot depth, we can project the points that define d_1 and d_2 onto l_s and then look for the largest distance among this four projected points. The center of the parking spot is located along the line l_s and at the mid-distance between the two projected points used to define the parking spot depth.

An example of the extraction of the empty parking place using real data is shown in Fig 2b where the green and white points correspond, respectively, the unfiltered and filtered data, the detected obstacles (two parked cars) are marked in red and the empty parking place is marked in green.

B. Extraction of current features for sensor based control

In order to be able to perform the parking maneuver with the previously presented sensor based control approach, it is necessary to extract the current sensor features required to perform the control.

Having two obstacles on the environment, first it is necessary to define if the difference of distances from the obstacles to the car is greater along x_m or along y_m . If the difference of distances is greater along x_m , obs_1 will be the obstacle that is further to the back of the vehicle for a reverse maneuver and further to the front for a forward maneuver. If the difference of distances is greater along y_m , obs_1 will be the obstacle that is further to the left of the vehicle.

There are of course many different strategies that could be considered to choose the sensor feature s_1 (i.e. [12]).

Since we are interested in having an easy way to compare the results obtained from the two different controllers, $s_1 = [x_1, y_1, \beta]^T$ was chosen so we can define a desired sensor feature value analogous to the goal position and orientation from the path planning scenario, taking advantage of the shared units.

The sensor feature s_1 is extracted from the closest side (of the two largest sides) of the bounding box obs_1 to the vehicle as it is shown in Fig. 3b. The point (x_1, y_1) is chosen to be always the one further to the back of the vehicle. The

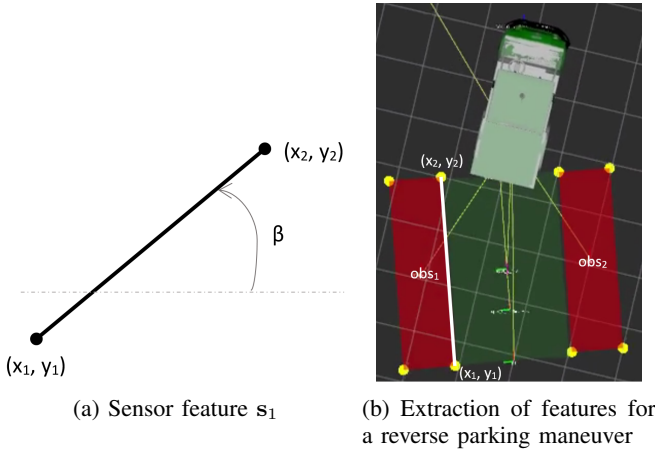


Fig. 3: Sensor features

angle β express the angle between the line segment defined by the points (x_1, y_1) and (x_2, y_2) and the x_m axis.

The constraint sensor features s_2 and s_3 correspond to, respectively, the closest distance between either the left or right side of the vehicle and obs_1 or obs_2 .

$$\forall j \in [1, 2] : s_{1+j} = distToObs_j = \sqrt{x_{obs_j}^2 + y_{obs_j}^2} \quad (7)$$

$$\mathbf{L}_{s_1} = \begin{bmatrix} -1 & y_1 \\ 0 & -x_1 \\ 0 & -1 \end{bmatrix} \quad (8)$$

$$\mathbf{L}_{s_{1+j}} = \begin{bmatrix} -\frac{x_{obs_j}}{distToObs_j} & 0 \end{bmatrix} \quad (9)$$

$$\mathbf{L}_i = \frac{\mathbf{L}_{s_i} + \mathbf{L}_{s_i}^*}{2} \quad (10)$$

IV. RESULTS

A. Fast prototyping environment

A homemade fast prototyping environment using the same software architecture than the one embedded inside the car is used for simulation purposes.

One of the current limitations of this homemade simulator is that it cannot simulate some exteroceptive sensors such as LiDARs and cameras. To overcome this limitation, an interface that allows to perform a co-simulation with Gazebo has been developed.

B. Simulation results

In order to perform the comparison between the two approaches using the fast prototyping environment. Considering the placement of VLP16 (on the roof of the vehicle) and the visibility limitations of the sensor, two box-shaped obstacles with a height of 2 m representing two parked cars were included in the Gazebo world. The free space length between the obstacles is of 2.5 m. For both approaches, the value of τ used in the respective velocity profile is equal to 0.5.

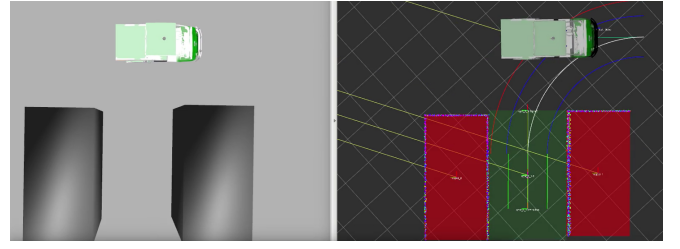


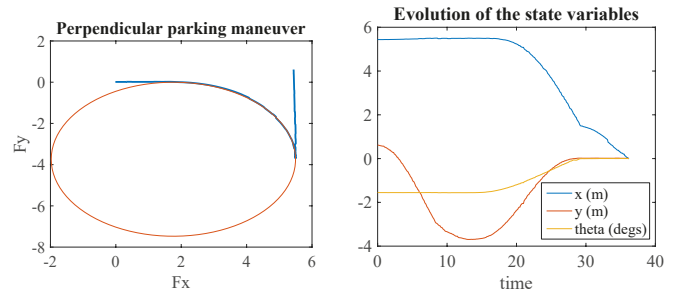
Fig. 4: Gazebo environment with two obstacles on the left side, rviz visualizer showing the generated path and the detected obstacles on the right side

1) *Path planning approach:* The technique presented in [6] was chosen as a reference path planning based approach due to its relatively simplicity and good performance.

The constants of the saturated steering controller were $K_t = 8$, $K = 1.85$ and $a_0 = 0.17$.

In Fig 5a it can be seen how the path performed by the vehicle was really close to what it should be, achieving a final error of -7.2 mm over Fx , 4 mm over Fy and 0.0007° on the orientation.

In Fig. 6a it can be seen how, as expected, the vehicle starts the forward straight line motion with a positive velocity and changing to negative velocity when the point 1 was reached and stopping when the goal was reached. In Fig. 6b it can be observed how the saturated steering control signal has, as expected, very abrupt changes but almost no chattering on the final part of the maneuver when the vehicle's orientation is close to the desired one.



(a) Path performed by the vehicle and the reference circle that connects positions 1 and 2

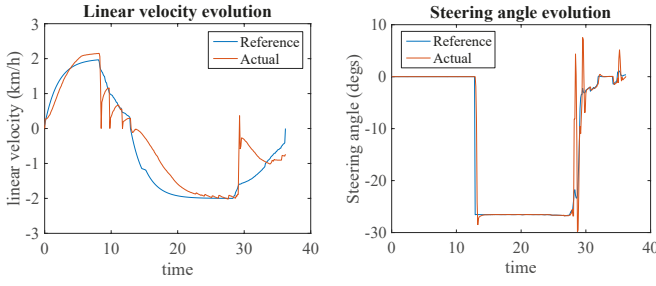
(b) Evolution of the state variables over time

Fig. 5: Path performed and evolution of the state variables

2) *Sensor based control approach:* Taking into account the obstacles characteristics, s_1^* was defined as follows:

$$\mathbf{s}_1^* = [-1, 1.25, 0]^T \quad (11)$$

From s_1^* , the first two components define the desired final position, comparable to the goal in a path planning scenario; for this reason and because they are expressed in the same units, the first two components of \mathbf{e} can be compared to the errors over Fx and Fy respectively from the path planning results. The third component of s_1^* is used to set the desired final orientation so the third component of \mathbf{e} can be compared



(a) Evolution of the vehicle's linear velocity - path planning (b) Evolution of the vehicle's steering angle

Fig. 6: Evolution of the control inputs and the response of the system - path planning

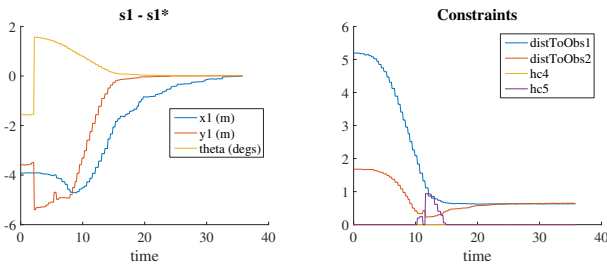
to the error in orientation from the path planning results since they are expressed in the same units.

In Fig. 7b and Fig. 9b it can be seen that the constraints are respected for both reverse and forward maneuvers. It can be noticed as well how, as expected, the vehicle gets closer to the obstacles when moving forward provoking a higher weight of h_4^c that leads to a reduction of the linear velocity (Fig. 10a).

In Fig. 8b and Fig. 10b it can be seen that for the most part of the maneuver, the evolution of the steering control signal has much less abrupt changes when compared to the saturated controller of the path planning approach, except when the vehicle is close to the goal where the controller tries to compensate for the remaining errors with strong changes on the steering angle.

Regarding the longitudinal velocity (Fig. 8a, Fig. 10a), it can be seen that the smoothness of the control signal is similar among both approaches when the constraints have small weights.

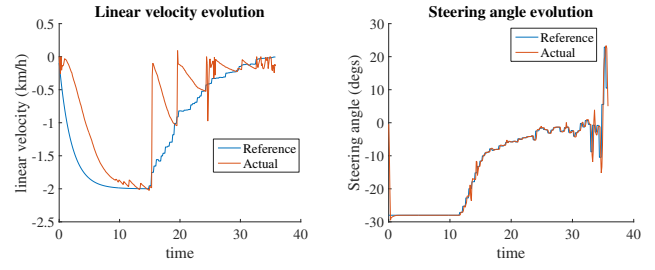
The oscillatory response from the system comes from the simulator and is not related to the controller.



(a) Evolution of the error signal (b) Evolution of the constraints - reverse maneuver

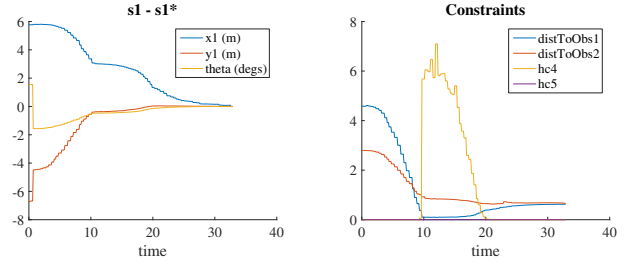
Fig. 7: Evolution of the sensor features and constraints - reverse maneuver

The final sensor signal error values can be seen in the table II. By performing a comparison between the two different approaches, path planning and sensor based control, it can be seen that the proposed approach is able to achieve slightly smaller errors in position than the path planning approach which already achieves a good performance.



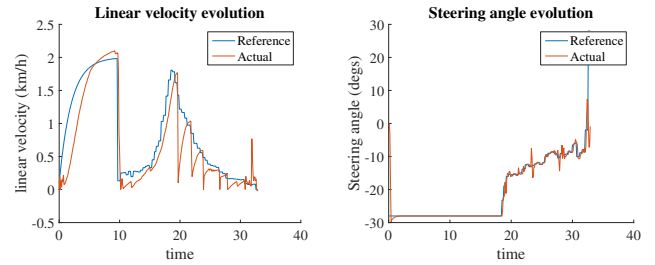
(a) Evolution of the linear velocity - reverse maneuver (b) Evolution of the steering angle - reverse maneuver

Fig. 8: Evolution of the control inputs and the response of the system - reverse maneuver



(a) Evolution of the error signal (b) Evolution of the constraints - forward maneuver

Fig. 9: Evolution of the sensor features and constraints - forward maneuver



(a) Evolution of the error signal (b) Evolution of the constraints - forward maneuver

Fig. 10: Evolution of the sensor features and constraints - forward maneuver

C. Real experimentation - Preliminary results

For the real experimentation, it was attempted to replicate the most relevant features of the simulated environment, having a free space length between the obstacles of approximately 2.5 m, the height of the obstacles being approximately equal to 2 m and s_1^* defined as in (11).

In Fig. 11b it can be seen that the constraints are respected during the maneuver. It can be noticed how the weight of h_5^c increases as the vehicle approaches to an obstacle.

In Fig. 12a and Fig. 12b it can be seen, respectively, how the evolution of the longitudinal velocity and steering control signals are very similar to the simulated case (Fig. 8b).

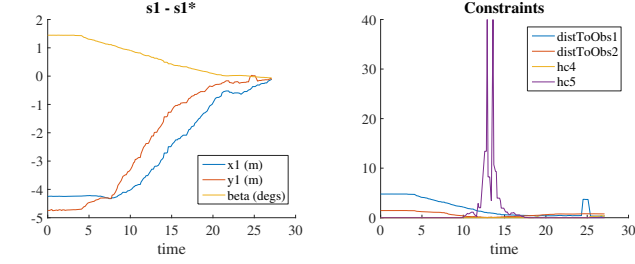
The final sensor signal error values are -10.4 cm for x_1 , -11.9 cm for y_1 and -0.0637° for β . It can be clearly seen

TABLE II: Final sensor signal error

$s_1 - s_1^*$	Reverse maneuver	Forward maneuver
x_1	-4.9 mm	-2.8 mm
y_1	7.4 mm	5.9 mm
β	-0.0068°	0.0018°

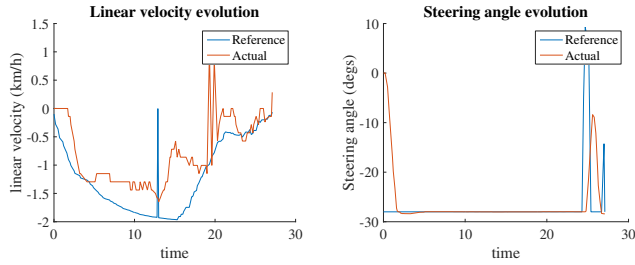
that these errors are higher when compared to the simulated case (table II). This can be explained by the erratic response of the longitudinal velocity and the slower response of the steering angle when compared to the simulations.

In Fig. 13 a sequence of snapshots of the evolution of the reverse parking maneuver can be seen.



(a) Evolution of the error signal (b) Evolution of the constraints

Fig. 11: Evolution of the sensor features and constraints - real experimentation



(a) Evolution of the linear velocity (b) Evolution of the steering angle

Fig. 12: Evolution of the control inputs and the response of the system - real experimentation



Fig. 13: Evolution of the sensor based perpendicular parking maneuver

V. CONCLUSIONS

From the results obtained it can be seen that the path planning approach is able to achieve a very good accuracy in the parking maneuver when a good enough localization

is available. During some of the many different conducted simulations, it was observed that when the localization's drift grows the performance of the path planning approach is considerably affected leading to higher final errors and in some cases to an oscillation effect on the steering angle.

The sensor based approach due to its nature is not affected by the localization's accuracy, in fact, the localization is not required. This characteristic could be very useful because it is to be expected that the localization accuracy degrades in parking lots, especially in those that are underground.

Furthermore, as it was demonstrated, the modifications required to change between a reverse and a forward maneuver with the proposed sensor based approach are really small, showing its high versatility.

Even though the preliminary results obtained from real experimentation are not ideal, they are encouraging, considering the fact that the erratic response does not come from the proposed sensor based controller but rather from the low level velocity controller.

REFERENCES

- [1] M. Seiter, H.-J. Mathony, and P. Knoll, "Parking Assist," in *Handbook of Intelligent Vehicles*, A. Eskandarian, Ed. London: Springer London, 2012, no. 1, pp. 829–864.
- [2] W. Wang, Y. Song, J. Zhang, and H. Deng, "Automatic parking of vehicles: A review of literatures," *International Journal of Automotive Technology*, vol. 15, no. 6, pp. 967–978, 2014.
- [3] M. Marouf, E. Pollard, and F. Nashashibi, "Automatic parallel parking and platooning to redistribute electric vehicles in a car-sharing application," *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 486–491, 2014.
- [4] C. Laugier and I. E. Paromtchik, "Autonomous Parallel Parking of a non Holonomic Vehicle," *Intelligent Vehicle Symposium 1996, Proceedings of the 1996 IEEE*, pp. 13–18, 1996.
- [5] Jaeyoung Moon, I. Bae, Jae-gwang Cha, and Shiho Kim, "A trajectory planning method based on forward path generation and backward tracking algorithm for Automatic Parking Systems," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2014, pp. 719–724.
- [6] P. Petrov, F. Nashashibi, and M. Marouf, "Path Planning and Steering control for an Automatic Perpendicular Parking Assist System," *7th Workshop on Planning, Perception and Navigation for Intelligent Vehicles, PPNIV'15*, pp. 143–148, 2015.
- [7] H. Vorobieva, S. Glaser, N. Minoiu-Enache, and S. Mammari, "Automatic parallel parking in tiny spots: Path planning and control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 396–410, 2015.
- [8] K. Min and J. Choi, "A control system for autonomous vehicle valet parking," in *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, no. Iccas. IEEE, oct 2013, pp. 1714–1717.
- [9] D. A. de Lima and A. C. Victorino, "Sensor-Based Control with Digital Maps Association for Global Navigation: A Real Application for Autonomous Vehicles," *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 1791–1796, 2015.
- [10] Y. Kang, D. A. de Lima, and A. C. Victorino, "Dynamic obstacles avoidance based on image-based dynamic window approach for human-vehicle interaction," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, no. Iv. IEEE, jun 2015, pp. 77–82.
- [11] O. Kermorgant and F. Chaumette, "Dealing with constraints in sensor-based robot control," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 244–257, 2014.
- [12] A. C. Victorino, P. Rives, and J.-J. Borrelly, "Safe Navigation for Indoor Mobile Robots. Part I: A Sensor-based Navigation Framework," *The International Journal of Robotics Research*, 2004.