Autonomous Perpendicular And Parallel Parking Using Multi-Sensor Based Control*

David Pérez-Morales[1], Olivier Kermorgant[1], Salvador Domínguez-Quijada[1] and Philippe Martinet[1]

***Abstract*— This paper addresses the perpendicular and parallel parking problems of car-like vehicles for both forward and reverse maneuvers in one trial by improving the work presented in [1] using a multi sensor based controller with a weighted control scheme. The perception problem is discussed briefly considering a Velodyne VLP-16 and a SICK LMS151 as the sensors providing the required exteroceptive information. The results obtained from simulations and real experimentation for different parking scenarios show the validity and potential of the proposed approach.**

I. Introduction

Even for experienced drivers, parking can be a difficult task, especially in big cities were the parking spots are often very narrow. The search for an increase in comfort and safety when parking has lead to a quite extensive literature [2], having explored many different approaches to automate this bothersome task.

Despite the fact that the automobile industry has already started to roll out some commercial implementations of active parking assistants capable of actively controlling acceleration, breaking and steering [3], the research interest in the topic remains strong.

Path planning approaches have been heavily investigated in recent years. Among the different planning techniques it is possible to distinguish between geometric approaches, with either constant turning radius [4], [5] using saturated feedback controllers, or continuous-curvature planning using clothoids [6]; heuristic approaches [7] and machine learning techniques [8]. It is worth to note that parking maneuvers with forward motions are seldom considered, with [9] for the parallel parking case and [10] for the perpendicular case being some of the few works on this regard.

A well known drawback of path planning and tracking is its dependence on the localization performance. An interesting alternative that does not rely on the localization is the use of a sensor based control approach. It has been proven to be valid for navigation [11], dynamic obstacles avoidance [12], and for parking applications [1].

*This work was supported by the Mexican National Council for Science and Technology (CONACYT). This paper describes work carried out in the framework of the Valet project, reference ANR-15-CE22-0013-02.

[1] David Pérez-Morales, Olivier Kermorgant, Salvador Domínguez-Quijada and Philippe Martinet are with LS2N, Laboratoire des Sciences du Numérique de Nantes, École Centrale de Nantes, 1 rue de la Noë, 44321 Nantes, France

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

[1] Olivier.Kermorgant@ec-nantes.fr

[1] Salvador.DominguezQuijada@ls2n.fr

[1] Philippe.Martinet@ec-nantes.fr

The contribution of this paper is an improvement on the approach described in [1], this time considering multiple sensors, a better suited sensor feature set that allows to park in one maneuver not only in perpendicular spots but also in parallel ones with either reverse or forward motions with only some minor changes, and improved constraints for collision avoidance.

In the next section the models considered as well as the notation used are presented. In Section III the perception problem is briefly addressed showing how the sensor data is processed in order to extract the empty parking spot to latter in Section IV describe the interaction model and how to extract the required sensor features from the computed empty parking spot. Afterwards, the controller is presented in Section V and the obtained results from simulation and real experimentation for different parking scenarios are shown in Section VI. Finally, some conclusions are given in Section VII.

II. Modeling and notation

Given that parking maneuvers are low speed motions, a kinematic model can be considered as accurate enough.

A. Car-like robot model and notation

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

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

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

Parameteres	*Notation*	*Value*
Wheelbase: Distance between the front and rear wheel axles	l_{wb}	2.588 m
Rear overhang: Distance between the rear wheel axle and the rear bumper	l_{ro}	0.657 m
Maximum steering angle	ϕ_{max}	30°
Total length of the vehicle	l_{ve}	4.084 m
Total width of the vehicle	w_{ve}	1.945 m
Maximum (desired) longitudinal velocity	$\lvert v_{max}\rvert$	2 km/h
Maximum acceleration increment	Δ_{acc}	$\text{sign}(v)\ 0.2\ T_s$
Maximum deceleration increment	Δ_{dec}	$\text{sign}(v)\ 2.5\ T_s$
Maximum ϕ increment	Δ_{ϕ}	$2°\ T_s$

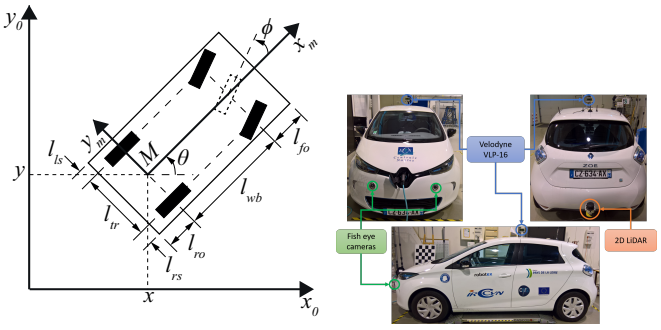
The point M is located at the mid-distance between the passive fixed wheels (rear) axle and the distance between the

rear and the front axle is described by l_{wb}. The generalized coordinates are $\mathbf{q} = [x, y, \theta, \phi]^T$ 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 $\dot{\theta}$:

$$\phi = \text{atan}(\frac{\dot{\theta}\, l_{wb}}{v}) \tag{2}$$

The vehicle used for experimentation and simulation is a Renault ZOE (Fig. 1b). It is represented by its bounding rectangle.



(a) Kinematic model diagram (b) Robotized Renault ZOE

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

B. Multi-sensor modeling

Following our previous work [1], where a novel sensor based control technique based on the framework described in [13] was proposed, in this paper we explore a different sensor features set to park in one maneuver into perpendicular and parallel spots considering multiple sources for the sensor signals.

1) Kinematic model: Let us consider a robotic system equipped with k sensors (Fig. 2) that provide data about the robot pose in its environment. Each sensor S_i gives a signal (sensor feature) $\mathbf{s}_i$ of dimension d_i with $\sum_{i=1}^{k} d_i = d$.

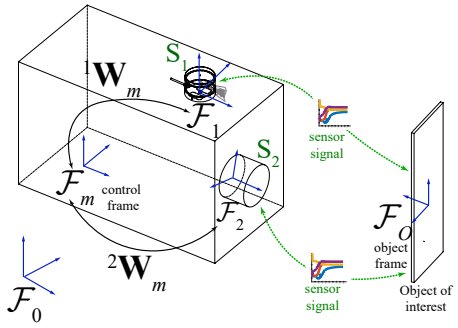


Fig. 2: Multi-sensor model

In a static environment, the sensor feature derivative can be expressed as follows:

$$\dot{\mathbf{s}}_i = \mathbf{L}_i \mathbf{v}_i = \mathbf{L}_i {}^i\mathbf{W}_m \mathbf{v}_m \tag{3}$$

where $\mathbf{L}_i$ is the interaction matrix of $\mathbf{s}_i$ and ${}^i\mathbf{W}_m$ is the screw transformation matrix that allows to express the sensor twist $\mathbf{v}_i$ with respect to the robot twist $\mathbf{v}_m$.

Assuming that the vehicle to which the sensors are rigidly attached to evolves in a plane and that the sensors and vehicle have vertical parallel z axes, $\mathbf{L}_i$ is of dimension $d_i \times 3$ and the screw transformation matrix takes the following form:

$${}^i\mathbf{W}_m = \begin{bmatrix} \mathrm{c}({}^m\theta_i) & \mathrm{s}({}^m\theta_i) & t_{i_x}\mathrm{s}({}^m\theta_i) - t_{i_y}\mathrm{c}({}^m\theta_i) \\ -\mathrm{s}({}^m\theta_i) & \mathrm{c}({}^m\theta_i) & t_{i_x}\mathrm{c}({}^m\theta_i) + t_{i_y}\mathrm{s}({}^m\theta_i) \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where ${}^m\mathbf{t}_i = [t_{i_x}, t_{i_y}]^T$ and ${}^m\theta_i$ are, respectively, the position and orientation of S_i with respect to $\mathcal{F}_m$ expressed in $\mathcal{F}_m$, with $\mathrm{c}({}^m\theta_i) = \cos({}^m\theta_i)$ and $\mathrm{s}({}^m\theta_i) = \sin({}^m\theta_i)$.

Denoting $\mathbf{s} = (\mathbf{s}_1, \ldots, \mathbf{s}_k)$ the d-dimensional signal of the multi-sensor system, the signal variation over time can be linked to the moving vehicle twist:

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v}_m \tag{5}$$

with:

$$\mathbf{L}_s = \mathbf{L}\mathbf{W}_m = \begin{bmatrix} \mathbf{L}_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \mathbf{L}_k \end{bmatrix} \begin{bmatrix} {}^1\mathbf{W}_m \\ \vdots \\ {}^k\mathbf{W}_m \end{bmatrix} \tag{6}$$

Nevertheless, since in our application the control frame $\mathcal{F}_m$ is attached to the vehicle's rear axis with origin at the M point (Fig. 1a), it is not possible to generate a velocity along y_m on the vehicle's frame due to the nonholonomic constraint of the kinematic model (1). Assuming that there is no slipping nor skidding (i.e. $v_{y_m} = 0$), the robot twist $\mathbf{v}_m = [v_{x_m}, v_{y_m}, \dot{\theta}]^T$ can be reduced to:

$$\mathbf{v}_m = [v_{x_m}, \dot{\theta}]^T \tag{7}$$

where $v_{x_m} = v$ and considering as well the consequent reduction of $\mathbf{L}_s$, being now of dimension $d \times 2$.

2) Weighted error: We consider the weighted multi-sensor error signal, as described in [13], which is defined as:

$$\mathbf{e}_\mathbf{H} = \mathbf{H}\mathbf{e} \tag{8}$$

where $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$ is the difference between the current sensor signal $\mathbf{s}$ and its desired value $\mathbf{s}^*$ and $\mathbf{H}$ is a diagonal positive semi-definite weighting matrix that depends on $\mathbf{s}$ with its associated interaction matrix being $\mathbf{L}_\mathbf{H} = \mathbf{H}\mathbf{L}_s$. Making a distinction between task and constraints features, $\mathbf{H} = \text{diag}(\mathbf{H}_t, \mathbf{H}_c)$ and $\mathbf{s} = [\mathbf{s}_t, \mathbf{s}_c]^T$. Each component h_i of $\mathbf{H}$ may or may not vary in order to optimize the system behavior, ensure specific constraints, manage priorities or add or remove a sensor or a feature from the control law.

Task features $\mathbf{s}_t$, as their name suggest, are used to perform the task by driving $\mathbf{e}_t$ to 0. On the other hand, since the constraints features $\mathbf{s}_c$ are used only to ensure certain constraints, we don't care about $\mathbf{e}_c$ as the desired value $\mathbf{s}_c^*$ is meaningless.

III. Perception

We focus the perception on the detection of parked cars. They can be approximated by boxes considering that, when viewed from the top, have a rectangular-like shape.

The vehicle used (Fig. 1b) has been equipped with many sensors (Velodyne VLP-16, SICK LMS151, 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 at, potentially close distances, the VLP-16 and SICK LMS151 were the sensors chosen to work with.

Because both sensors provide information of a very similar nature, the data can be fused by simply converting the *LaserScan* data provided by the LMS151 to *PointCloud2* and then transforming the point cloud from LMS151's frame to the VLP-16's frame so it can be added to the point cloud provided by the latter sensor. For this, it is assumed that the time difference between the data provided by each sensor is reasonably small, i.e. the data is sufficiently synchronized.

The complete point cloud obtained from the two sensors is first filtered with a couple of crop boxes. The first crop box is keeps 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 and the second one is used to filter out the points that belong to the car's body (self-collision sensor readings). Then, an Euclidean Cluster Extraction algorithm is used to have each obstacle represented as a cluster. The orientation of each cluster is extracted by fitting a line model to the points belonging to the contour of the cluster using a RANSAC algorithm. The orientation of the bounding box will be equal to the orientation of the fitted line. After, we proceed by finding the rotated bounding box of the cluster using the previously found orientation.

The empty parking place (green rectangle in Fig. 3) is extracted using the approach described in [1]. The sensor features required for the controller are extracted from this computed parking place.

IV. Interaction model

For the interaction model, we rely on the perception of several lines $\mathcal{L}_j$ and points from several sensors. Since the sensor data is expressed in the Cartesian space, it can be easily transformed from one frame to another, thus allowing us to use virtual sensors placed at will.

The sensor's placement can be seen in Fig. 3. S_1 corresponds to the VLP-16 while S_2 to the LMS151. S_3 to S_5 are virtual sensors placed on the corners of the car's bounding rectangle. All the frames of the virtual sensors have the same orientation as the control frame.

To illustrate the feature extraction approach, the case of a reverse perpendicular maneuver is now detailed. As it can be seen in Fig. 3, points p_1 to p_4 correspond to the corners of the parking spot while p_5 and p_6 are, respectively, the midpoints between (p_1, p_4) and (p_2, p_3). $\mathcal{L}_1$ is a line that passes through p_5 and p_6, i.e. it passes through the center of the parking spot. $\mathcal{L}_2$ is a line that passes through p_1 and p_4 thus corresponding to the depth limit of the parking spot. $\mathcal{L}_3$ is a line that passes through p_3 and p_4. All the lines are parametrized using normalized Plücker coordinates.

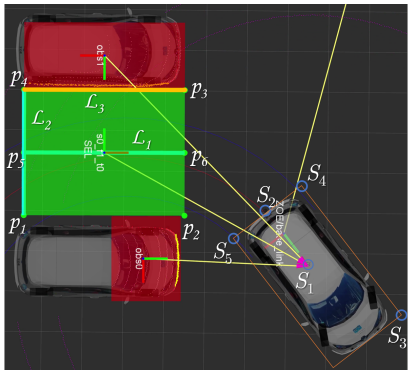


Fig. 3: Sensors' configuration and sensor features

The exact definition of the sensor features varies depending on the parking scenario although in any case, $\mathcal{L}_1^*$ should be collinear with x_m and $\mathcal{L}_2^*$ should be parallel to y_m and be behind the car for reverse maneuvers and in front for forward ones. In this paper we only detail the actual features used for a specific case, but deducing the features that should be used for other cases isn't complicated.

Considering the previously mentioned assumption that the vehicle to which the sensors are attached to evolves in a plane (sensors and vehicle with parallel z axes), the sensor signal $\mathbf{s}_{i_{\mathcal{L}_j}}$ and interaction matrix $\mathbf{L}_{i_{\mathcal{L}_j}}$ for the line $\mathcal{L}_j$ observed by S_i are defined respectively by (9) and (10)

$$\mathbf{s}_{i_{\mathcal{L}_j}} = \left[{}^i\underline{\mathbf{u}}_j(1), {}^i\underline{\mathbf{u}}_j(2), {}^i\mathbf{h}_j(3)\right]^T \tag{9}$$

$$\mathbf{L}_{i_{\mathcal{L}_j}} = \begin{bmatrix} 0 & 0 & {}^i\underline{\mathbf{u}}_j(2) \\ 0 & 0 & -{}^i\underline{\mathbf{u}}_j(1) \\ -{}^i\underline{\mathbf{u}}_j(2) & {}^i\underline{\mathbf{u}}_j(1) & 0 \end{bmatrix} \tag{10}$$

where ${}^i\underline{\mathbf{u}}_j = {}^i\mathbf{u}_j/||{}^i\mathbf{u}_j||$ with ${}^i\mathbf{u}_j \neq 0$ denoting the orientation of $\mathcal{L}_j$ and ${}^i\mathbf{h}_j = {}^i\mathbf{w}_j/||{}^i\mathbf{u}_j||$ with ${}^i\mathbf{w}_j$ containing the coefficients of the interpretation plane equation [14]. In the 2D configuration considered, the components of ${}^i\underline{\mathbf{u}}_j$ and ${}^i\mathbf{h}_j$ that don't appear in (9) are equal to 0 thus ${}^i\mathbf{h}_j(3)$ can be interpreted as the distance to the line.

It should be noted that the weighting and constraints required to park safely change depending on the type of parking spot (parallel, perpendicular or diagonal) and on which side the parking spot is placed with respect to the car at the beginning of the maneuver.

A. Task sensor features

The control features required to perform the parking task are defined by (11), with $t = 1$ for forward maneuvers and $t = 2$ for the reverse case.

$$\mathbf{s}_t = \left[\mathbf{s}_{t_{\mathcal{L}_1}}, \mathbf{s}_{t_{\mathcal{L}_2}}\right]^T \tag{11}$$

A 2nd order approximation of the form (12) is used for the interaction matrix.

$$\mathbf{L}_t = \frac{\mathbf{L}_{\mathcal{L}_j} + \mathbf{L}^*_{\mathcal{L}_j}}{2} \tag{12}$$

The weighting matrix $\mathbf{H}_t$ is defined by (13). The variable components h_i^t are computed using a smooth weighting function (Fig. 4) based on the one presented in [15].

$$\mathbf{H}_t = \text{diag}(h_1^t, h_2^t, h_3^{t_{const}}, h_4^t, h_5^t, h_6^{t_{const}}) \tag{13}$$

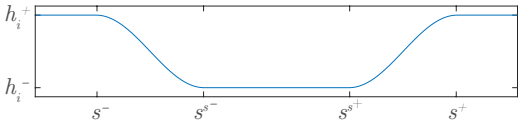


Fig. 4: Weighting function h_i^t

With this weighting function and considering the weighting parameters (31), $s_i \ \forall \ i = \{2, 4, 5\}$ could be seen as bounded constraints but in fact they are task features, given that we do care about the value of their corresponding e_i. If only s_3 and s_6 were considered as task features, the car may finish the maneuver with a bad orientation even if $e_3 \approx e_6 \approx 0$ because just 2 features are not enough to control the car's DOFs.

Due to space constraints, only the case of a reverse perpendicular parking maneuver with the spot placed on the right will be considered for the rest of this section.

B. Constraints sensor features

The constraints are defined by (14).

$$\mathbf{s}_c = [\mathbf{s}_3, \mathbf{s}_4, \mathbf{s}_5]^T \tag{14}$$

For the constraints sensor features we are interested only in the components of (9) related to the distance to the feature itself, therefore:

$$\mathbf{s}_3 = {}^3\mathbf{h}_3(3) \tag{15}$$

$$\mathbf{s}_4 = [{}^4\mathbf{h}_2(3), {}^4\mathbf{h}_3(3)]^T \tag{16}$$

$$\mathbf{s}_5 = [{}^5\mathbf{h}_2(3), {}^5d_{y_m}]^T \tag{17}$$

with ${}^5d_{y_m}$ being the difference $d_{y_m} = \rho_{corner} - \rho_{lat}$ measured with the sensor S_5, expressed in the sensor frame (Fig. 5) if $\phi < 0$, defined as:

$${}^5d_{y_m} = \sqrt{({}^5x_2 + t_{5_x})^2 + ({}^5y_2 + t_{5_y} - \rho_m)^2} + \rho_m - t_{5_y} \tag{18}$$

with $\rho_m = l_{wb}/\tan\phi$ and, when $\phi \geq 0$, being simply the distance from S_5 to p_2 along y_m measured with S_5, it is defined as:

$${}^5d_{y_m} = {}^5y_2 \tag{19}$$

with ${}^5p_2 = ({}^5x_2, {}^5y_2)$ being the point p_2 measured with S_5.

The corresponding interaction matrices are:

$$\mathbf{L}_3 = \begin{bmatrix} -{}^3\underline{\mathbf{u}}_3(2) & {}^3\underline{\mathbf{u}}_3(1) & 0 \end{bmatrix} \tag{20}$$

$$\mathbf{L}_4 = \begin{bmatrix} -{}^4\underline{\mathbf{u}}_2(2) & {}^4\underline{\mathbf{u}}_2(1) & 0 \\ -{}^4\underline{\mathbf{u}}_3(2) & {}^4\underline{\mathbf{u}}_3(1) & 0 \end{bmatrix} \tag{21}$$

$$\mathbf{L}_5 = \begin{bmatrix} -{}^5\underline{\mathbf{u}}_2(2) & {}^5\underline{\mathbf{u}}_2(1) & 0 \\ 0 & -1 & -{}^5x_2 \end{bmatrix} \tag{22}$$

Since the constraints are used for collision avoidance, only one side of the interval $[\mathbf{s}_c^-, \mathbf{s}_c^+]$ (25) has to be defined for each feature.

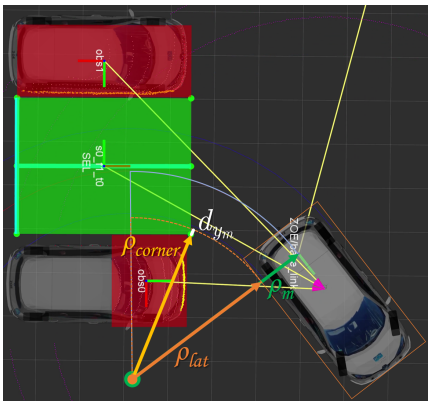


Fig. 5: Lateral constraint d

V. Control

When considering the constraints presented in Sec. IV-B (particularly (18)), a chattering problem appears if the controller presented in [1] is used, even with very small weights. For this reason, that controller had to be adapted to the quadratic programming form [16] with only inequality constraints (23):

$$\begin{aligned} \mathbf{v}_m &= \text{argmin}||\mathbf{L}_{\mathbf{H}_t}.\mathbf{v}_m + \lambda.\mathbf{e}_t||^2 \\ &\text{s.t. } \mathbf{A}\mathbf{v}_m \leq \mathbf{b} \end{aligned} \tag{23}$$

with:

$$\mathbf{A} = [\mathbf{L}_{\mathbf{H}_c}, -\mathbf{L}_{\mathbf{H}_c}]^T \tag{24}$$

$$\mathbf{b} = [\alpha(\mathbf{s}_c^+ - \mathbf{s}_c), -\alpha(\mathbf{s}_c^- - \mathbf{s}_c)]^T \tag{25}$$

where α is a gain constant, λ is the control gain, $\mathbf{H}_c$ is an identity matrix (i.e. there is no weighting on the constraints) and $[\mathbf{s}_c^-, \mathbf{s}_c^+]$ is the interval in which we want to keep $\mathbf{s}_c$.

To limit the speed of the vehicle as it approaches to the parking spot, a deceleration profile, based on the velocity profile shown in [5], is used. It is defined by (26)

$$\begin{aligned} &\texttt{if} \quad \mathbf{e}(6) < \mathbf{e}(6)^{th} \\ &\quad v_{max} = (|v_{max}| - v_{max}^0)(\mathbf{e}(6)/\mathbf{e}(6)^{th}) + v_{max}^0 \end{aligned} \tag{26}$$

with v_{max}^0 being the maximum desired velocity when the sixth component of the error vector $\mathbf{e}(6)$ tends to zero and $\mathbf{e}(6)^{th}$ is a threshold value for $\mathbf{e}(6)$. Since the low level velocity controller is not capable of reaching very small values, $v_{max}^0 = 0.2$ km/h.

The control signals v and ϕ are bounded by their respective maximum desired values as shown below:

$$|v| < |v_{max}| \tag{27}$$

$$|\phi| < \phi_{max} \tag{28}$$

To avoid large changes in the control signals at time k that may cause uncomfortable sensations for the passengers or surrounding witnesses, they are bounded by some increments (29) (30) with respect to the control signal at $k-1$.

$$(v_{k-1} - \Delta_{dec}) \geq v_k \leq (v_{k-1} + \Delta_{acc}) \tag{29}$$

$$(\phi_{k-1} - \Delta_\phi) \geq \phi_k \leq (\phi_{k-1} + \Delta_\phi) \tag{30}$$

To solve (23), a generic solver is used. To improve the stability and computation time, the optimization variables are $[v, \phi]$ and not $\mathbf{v}_m$, although inside the objective function $\dot{\theta}$ is computed from ϕ so (23) can be solved. When using ϕ instead of $\dot{\theta}$ one can easily impose the bounds (28) at the solving step instead of solving (23) directly with $\mathbf{v}_m$ as optimization variables and hope for the value of ϕ computed from $\dot{\theta}$ to fall inside the bounds (28).

VI. Results

To show the potential of our approach, several parking scenarios are presented below, all of them using the final form of the controller (23). The unconstrained cases were computed in MATLAB, using fmincon as solver. For the constrained cases, NLopt with the SLSQP algorithm was used.

A. Unconstrained cases - MATLAB

To evaluate the performance of the proposed approach, it was first tested in unconstrained cases with a sampling time $T_s = 0.1$. As it can be seen in Figs. 6-10, the presented technique allows to perform parking maneuvers for many different scenarios (perpendicular and parallel with either reverse or forward motions) just by adjusting the weighting parameters and the specific definition of the sensor features. The final errors for all of these cases are in the order of $\times 10^{-3}$ or smaller.

As an example of the weighting approach, for the case of a reverse perpendicular parking maneuver with the parking spot placed on the right, $h_i^+ = 5$, $h_i^- = 0$, $h_3^{t_{const}} = 1$, $h_6^{t_{const}} = 0.75$ and:

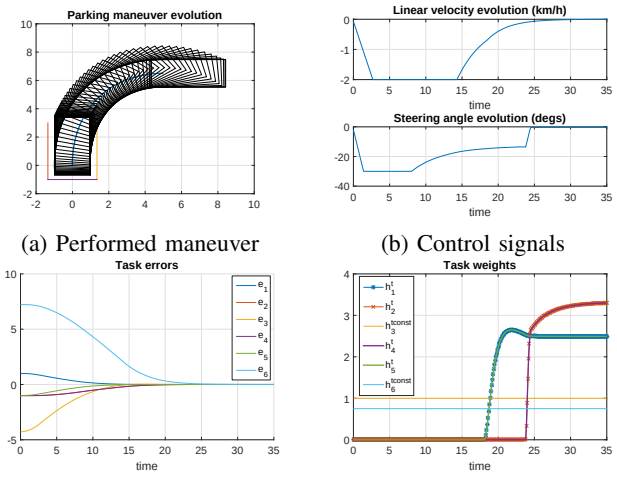
$$\begin{cases} s_1^{s^+} = s_1^+ = \infty \\ s_1^{s^-} = 0.001 + s_1^* \\ s_1^- = -0.001 + s_1^* \\ s_i^{s^-} = s_i^- = -\infty \quad \forall\, i = \{2,4,5\} \\ s_i^{s^+} = -0.001 + s_i^* \quad \forall\, i = \{2,4,5\} \\ s_i^+ = 0.001 + s_i^* \quad \forall\, i = \{2,4,5\} \end{cases} \tag{31}$$

These weighting parameters allow to prioritize the error in position over the orientation for the most part of the maneuver and, when ${}^i\mathbf{u}_j^*(a)$ is almost reached, smoothly increase the corresponding weights so we can gradually switch the priority from positioning the vehicle to orientate it to avoid finishing the maneuver with a bad orientation.

It can be seen how, for all shown cases, the weights (Figs. 6d-10d) push ϕ towards 0 (Figs. 6b-10b) once ${}^i\mathbf{u}_j^*(a)$ is almost reached when close to the completion of the maneuver to keep the orientation close to the desired value.

Unlike our previous work [1], which required to perform gain-tunning for different initial conditions, this newly presented approach allows to park successfully for different (reasonable) initial positions and orientations of the same parking case using the same weighting parameters, although it should be mentioned that the stability, specially when constraints are considered, is still under study.

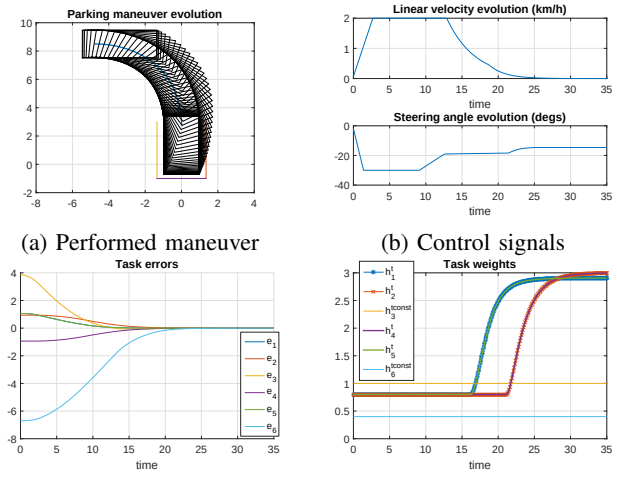
In Fig. 10, it can be seen how even if the car is placed considerably farther from the parking spot than in Fig. 6 and not exactly perpendicular to the spot, the vehicle is able to park correctly using the same weighting parameters, showing the stability of the presented approach.



(a) Performed maneuver (b) Control signals

(c) Task error signal (d) Task features' weights

Fig. 6: Unconstrained perpendicular reverse parking maneuver



(a) Performed maneuver (b) Control signals

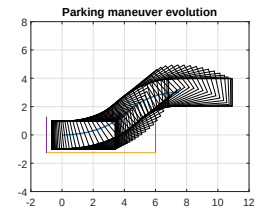
(c) Task error signal (d) Task features' weights

Fig. 7: Unconstrained perpendicular forward parking maneuver

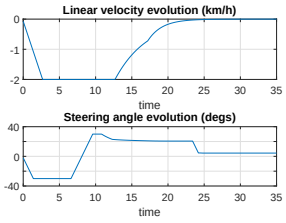
B. Constrained cases

1) Fast prototyping environment: A homemade fast prototyping environment using the same software architecture as the one embedded inside the car is used for simulation purposes. This homemade environment is interfaced with Gazebo to simulate the exteroceptive sensors.

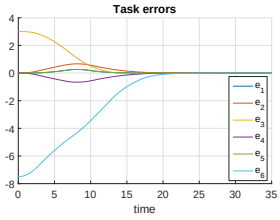
The case of a reverse perpendicular parking maneuver with the spot placed on the right is shown below. The weighting parameters remain the same as for the unconstrained case while the constraints are defined with $s_7^+ = -0.1$, $s_8^- = 0.15$, $s_9^+ = -0.1$, $s_{10}^- = 0.15$, $s_{11}^+ = -0.075$ and $T_s = 0.05$.



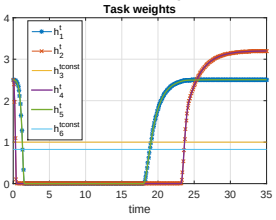
(a) Performed maneuver



(b) Control signals



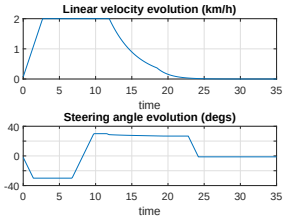
(c) Task error signal



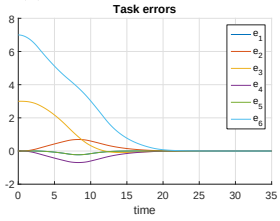
(d) Task features' weights

Fig. 8: Unconstrained parallel reverse parking maneuver

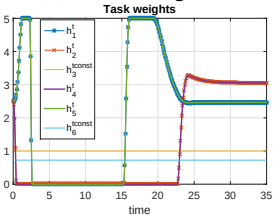
(a) Performed maneuver



(b) Control signals

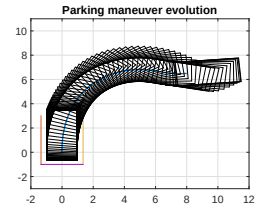


(c) Task error signal

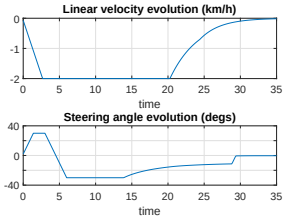


(d) Task features' weights

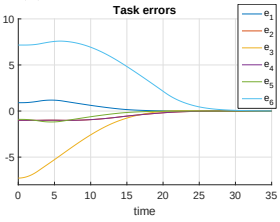
Fig. 9: Unconstrained parallel forward parking maneuver



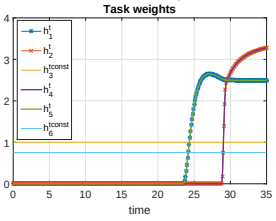
(a) Performed maneuver



(b) Control signals



(c) Task error signal



(d) Task features' weights

Fig. 10: Unconstrained perpendicular reverse parking maneuver from far

As it can be seen in Figs. 11-12, the car is able to park successfully while respecting the constraints in spite of the sensor noise and the less than perfect system response. The evolution of the many different signals, especially for the longitudinal velocity (Fig. 12a), is very similar to the unconstrained case (Fig. 6b). The fast deceleration at the end (Fig. 12a) is due to a stopping condition in the implementation related to $\mathbf{e}_t$. Regarding the evolution of ϕ, it can be seen how, contrary to the unconstrained case, it doesn't saturate; this behavior is caused by the constraints, particularly s_{11} (Fig. 12d). The final error is $\mathbf{e}_t = [-0.0003, -0.0096, 0.0207, -0.0096, 0.0003, 0.0569]^T$.

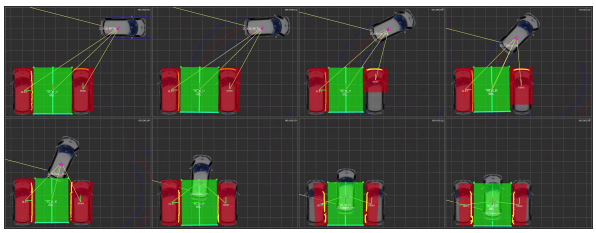
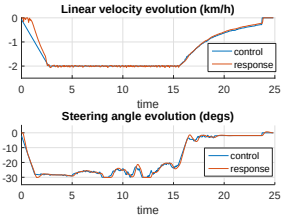
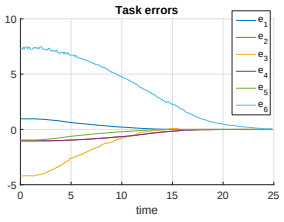


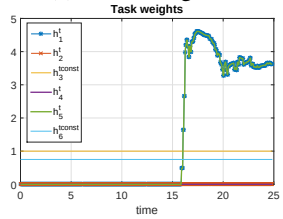
Fig. 11: Constrained perpendicular reverse parking maneuver



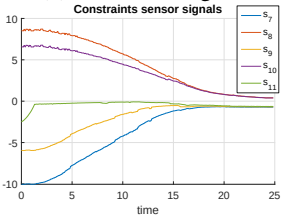
(a) Control signals



(b) Task error signal



(c) Task features' weights



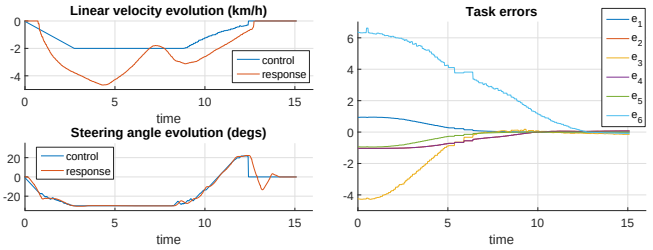
(d) Constraints sensor signals

Fig. 12: Constrained perpendicular reverse parking maneuver signals

2) Real experimentation: Real experimentation was conducted for the same parking case (Fig. 13) as with the fast prototyping environment shown above. The weighting parameters and constraints definition remain the same.

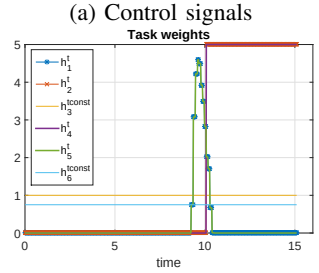
Fig. 13: Experimental car parking in a perpendicular spot

It is obvious that the response of the system, particularly for the linear velocity (Fig. 14a), is less than ideal, reaching

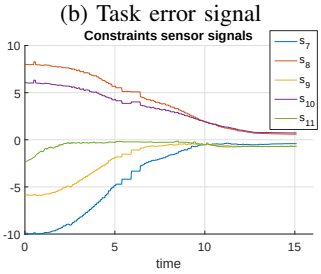


(a) Control signals

(b) Task error signal



(c) Task features' weights



(d) Constraints sensor signals

Fig. 14: Constrained perpendicular reverse parking maneuver signals

a speed more than twice as fast than what the controller indicates. This behavior can be attributed to the low-level velocity controller, which still requires some tunning to improve the performance at low velocities, therefore it has no relation to the presented technique.

Despite of the erratic response of the system in addition to the noise coming from the sensors, the constraints were respected during the whole maneuver (Fig. 14d), getting no closer than 33.88cm (s_9) to $\mathcal{L}_3$.

Furthermore, the evolution of $\mathbf{e}_t$ (Fig. 14b) is very similar to the simulated case (Fig. 12b), although the final error is not as good, being in this case $\mathbf{e}_t = [0.0054, 0.0743, -0.1436, 0.0743, -0.0054, -0.0833]^T$.

The smallest $||\mathbf{e}_t||$ was achieved at $T = 12.6399$s, with $\mathbf{e}_t = [0.0025, 0.0429, -0.0851, 0.0429, -0.0025, 0.0207]^T$ and $[v, \phi]^T = [0, 0]^T$ from the controller starting at $T = 12.42$s.

VII. Conclusions

Following our previous work [1], we showed how a better choice of the sensor features allows to improve the performance, stability and versatility of the presented sensor based approach, this time not only being able to deal successfully with perpendicular parking maneuvers but also with parallel ones with both reverse and forward motions with just some minor adjustments for each type of parking. The stability, specially when constraints are considered, is still under study.

Preliminary results obtained from real experimentation validate the robustness and effectiveness of the presented approach, considering that, despite of the erratic response of the system due to the low-level velocity controller, the car parked successfully while respecting the constraints during the whole maneuver.

It is important to mention that, due to visibility constraints and in order to keep the results obtained with the fast prototyping environment as close to the reality as possible, only reverse parking maneuvers have been tested outside MATLAB with the presented sensor feature set. Nevertheless, the multi-sensor framework gives a high expandability, allowing for future upgrades to the perception capabilities of the system.

References

[1] D. Pérez Morales, S. Domínguez Quijada, O. Kermorgant, and P. Martinet, "Autonomous parking using a sensor based approach," in *8th Workshop on Planning, Perception and Navigation for Intelligent Vehicles, PPNIV'16 at 19th IEEE ITSC 2016*, Rio de Janeiro, Brazil, 2016, pp. 211–216.

[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] Y. Song and C. Liao, "Analysis and Review of State-of-the-Art Automatic Parking Assist System," in *2016 IEEE International Conference on Vehicular Electronics and Safety*, Beijing, China, 2016, pp. 61–66.

[4] P. Petrov, F. Nashashibi, and M. Marouf, "Path Planning and Steering control for an Automatic Perpendicular Parking Assist System," in *7th Workshop on Planning, Perception and Navigation for Intelligent Vehicles, PPNIV'15*, Hamburg, Germany, 2015, pp. 143–148.

[5] P. Petrov and F. Nashashibi, "Saturated Feedback Control for an Automated Parallel Parking Assist System," in *13th International Conference on Control, Automation, Robotics and Vision (ICARCV'14)*, Marina Bay Sands, Singapore, 2014, pp. 577–582.

[6] H. Vorobieva, N. Minoiu-Enache, S. Glaser, and S. Mammar, "Geometric Continuous-Curvature Path Planning for Automatic Parallel Parking," in *2013 10th IEEE International Conference on Networking, Sensing and Control (ICNSC)*, Evry, France, 2013, pp. 418–423.

[7] C. Chen, M. Rickert, and A. Knoll, "Path planning with orientation-aware space exploration guided heuristic search for autonomous parking and maneuvering," in *2015 IEEE Intelligent Vehicles Symposium*, Seoul, Korea, 2015, pp. 1148–1153.

[8] G. Notomista and M. Botsch, "Maneuver segmentation for autonomous parking based on ensemble learning," in *2015 International Joint Conference on Neural Networks (IJCNN)*, Killarney, Ireland, 2015, pp. 1–8.

[9] H. Vorobieva, S. Glaser, N. Minoiu-Enache, and S. Mammar, "Automatic parallel parking in tiny spots: Path planning and control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 396–410, 2015.

[10] 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)*, Gwangju, South Korea, oct 2013, pp. 1714–1717.

[11] D. A. de Lima and A. C. Victorino, "Sensor-Based Control with Digital Maps Association for Global Navigation: A Real Application for Autonomous Vehicles," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Las Palmas, Spain, 2015, pp. 1791–1796.

[12] 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)*, Seoul, South Korea, jun 2015, pp. 77–82.

[13] 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.

[14] N. Andreff, B. Espiau, and R. Horaud, "Visual Servoing from Lines," *International Journal of Robotics Research*, vol. 21, no. 8, pp. 679–699, 2002.

[15] V. K. Narayanan, F. Pasteau, M. Marchal, A. Krupa, and M. Babel, "Vision-based adaptive assistance and haptic guidance for safe wheelchair corridor following," *Computer Vision and Image Understanding*, vol. 149, pp. 171–185, 2016.

[16] O. Kanoun, F. Lamiraux, and P.-B. Wieber, "Kinematic control of redundant manipulators: generalizing the task priority framework to inequality tasks," *IEEE Trans. on Robotics*, vol. 27, no. 4, pp. 785–792, 2011.