Multi-Sensor-Based Predictive Control for Autonomous Backward Perpendicular and Diagonal Parking

David Pérez-Morales, Olivier Kermorgant, Salvador Domínguez-Quijada, Philippe Martinet

Multi-Sensor-Based Predictive Control for Autonomous Backward Perpendicular and Diagonal Parking

David Pérez-Morales[1], Olivier Kermorgant[2], Salvador Domínguez-Quijada[3] and Philippe Martinet[4]

***Abstract*— This paper explores the feasibility of a Multi-Sensor-Based Predictive Control (MSBPC) approach for addressing backward nonparallel (perpendicular and diagonal) parking problems of car-like vehicles as an alternative to more classical (e.g. path planning based) approaches. The results of a few individual cases are presented to illustrate the behavior and performance of the proposed approach as well as results from exhaustive simulations to assess its convergence and stability. Indeed, preliminary results are encouraging, showing that the vehicle is able to park successfully from virtually any sensible initial position.**

I. INTRODUCTION

Even though the research on autonomous parking started more than 20 years ago, leading to a quite extensive literature [1] and in spite of the fact that the automobile industry has already started to roll out some commercial implementations of active parking assistants capable of actively controlling acceleration, braking and steering [2], the research interest in the topic remains strong. This is, partially at least, due to the ever-growing size of many cities around the world, leading to an increment in the number of automobiles in the streets and thus causing parking to become an increasingly difficult and dangerous task.

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 [3], [4] using saturated feedback controllers, or continuous-curvature planning using clothoids [5], [6]; heuristic approaches [7] and machine learning techniques [8].

A well-known drawback of path planning is that it is necessary to have knowledge about the free and occupied space of the whole environment beforehand if online replanning is not feasible, potentially leading to costly infrastructure requirement. Moreover, it is known that path planning algorithms that consider some kind of space exploration step (such as A*, RRT, etc.) have to make a compromise between computation time and exploration's completeness. Furthermore, the tracking performance of a given path is highly dependent on the localization performance which might get degraded on certain environments (e.g. underground parking lots without any special infrastructure) or after a few maneuvers leading to non-negligible differences between the planned path and the performed one [5], [6].

An interesting alternative is the use of a sensor-based control approach. It has been proven to be valid for navigation [9], dynamic obstacle avoidance [10] and for parking applications [11], [12]. It should be noted that an important limitation of a purely sensor-based control approach is the possibility of getting trapped in local minima – i.e. if the car is not able to park in one maneuver from the initial pose then the parking maneuver won't be successful.

A. Reasoning and contribution

A natural goal for a human driver when parking would be to try to make the vehicle's longitudinal axis to be collinear to the main axis of the parking spot (i.e. to be centered lateral-wise) and finish the maneuver at a certain distance from the rear boundary of the parking spot while avoiding collision with surrounding obstacles during the whole maneuver.

Assuming that the vehicle is capable of perceiving surrounding free parking spots, it is possible to park without any path planning using a Multi-Sensor-Based Predictive Control (MSBPC) approach by minimizing the error between the current value of a certain set of sensor features (i.e. a line collinear to the parking spot's main axis and another collinear to the rear boundary of the parking spot) and its desired value while avoiding collision by imposing certain constraints on another set of sensor features (lines defining the boundaries of the parking spot, points at the corners of said spot, etc.). It is worth noting that, since the presented approach is based on the features perceived at each time instant and a certain desired fixed value for each feature, no localization is inherently required for it to be stable in spite of the prediction step considered.

The contribution of this paper is the exploration of a MSBPC approach for backward perpendicular and diagonal parking, being able now to park with multiple maneuvers. It should be noted that, in order to decouple the performance of the controller from the perception, the sensory data is generated virtually and assumed to be available all the time.

B. Contents of the paper

In the next section the kinematic model of the vehicle and the multi-sensor modeling are presented. Section III describes the interaction model allowing to formalize the parking tasks and the constraints for collision avoidance. Afterwards, the controller is presented in Section IV. The

[1] David Pérez-Morales, [2]Olivier Kermorgant and [3]Salvador Domínguez-Quijada are with LS2N, Laboratoire des Sciences du Numérique de Nantes, École Centrale de Nantes, 1 rue de la Noë, 44321 Nantes, France

[4] Philippe Martinet is with INRIA Sophia Antipolis, 2004 Route des Lucioles, 06902 Valbonne, France and École Centrale de Nantes - LS2N, 1 rue de la Noë, 44321 Nantes, France

[1] David.Perez-Morales@eleves.ec-nantes.fr
[2] Olivier.Kermorgant@ec-nantes.fr
[3] Salvador.DominguezQuijada@ls2n.fr
[4] Philippe.Martinet@inria.fr

obtained results are presented in Section V: a few cases in two different simulation environments are presented as well as exhaustive simulations results for assessing the convergence performance of the presented approach for the two different types of parking maneuvers addressed are shown. Finally, some conclusions are given in Section VI.

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 considered kinematic model is a car with rear-wheel driving:

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

where v and $\dot{\phi}$ are the longitudinal and steering velocities.

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 $\mathbf{q} = [x, y, \theta, \phi]^\mathsf{T}$ 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 denoted by ϕ (Fig. 1a).

The turning radius ρ_m around the instantaneous center of rotation (ICR) can be defined as:

$$\rho_m = \frac{l_{wb}}{\tan\phi} \tag{2}$$

The vehicle used for experimentation and simulation, represented by its bounding rectangle in Fig. 1a, is a Renault ZOE (Fig. 1b). Its relevant dimensional parameters are presented in Table I.

TABLE I
DIMENSIONAL VEHICLE PARAMETERS

Parameters	*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
Total length of the vehicle	l_{ve}	4.084 m
Total width of the vehicle	w_{ve}	1.945 m

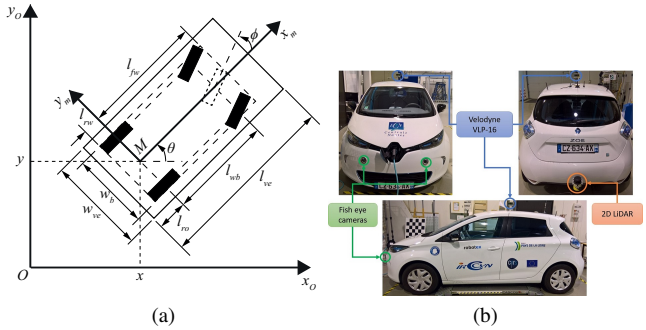
B. Multi-sensor modeling

The considered multi-sensor modeling is recalled in this subsection.

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

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

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



(a) (b)

Fig. 1. (a) Kinematic model diagram for a car-like rear-wheel driving robot. (b) Robotized Renault ZOE used for real experimentation

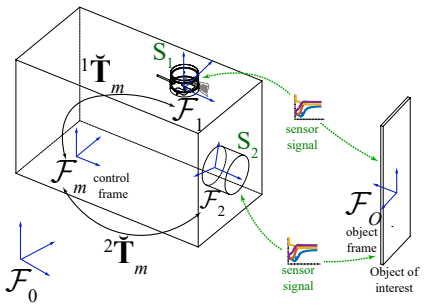


Fig. 2. Multi-sensor model

where $\breve{\mathbf{L}}_i$ is the interaction matrix [13] of $\mathbf{s}_i$ ($\dim(\breve{\mathbf{L}}_i) = \mathbb{d}_i \times 6$) and ${}^i\breve{\mathbf{T}}_m$ is the 3D screw transformation matrix that allows expressing the sensor twist $\breve{\mathbf{v}}_i$ (which is expressed in its corresponding frame $\mathcal{F}_i$) with respect to the robot twist $\breve{\mathbf{v}}_m$ (expressed in the control frame $\mathcal{F}_m$).

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

$$\dot{\mathbf{s}} = \breve{\mathbf{L}}_\mathbf{s} \breve{\mathbf{v}}_m \tag{4}$$

with:

$$\breve{\mathbf{L}}_\mathbf{s} = \breve{\mathbf{L}} \breve{\mathbf{T}}_m \tag{5}$$

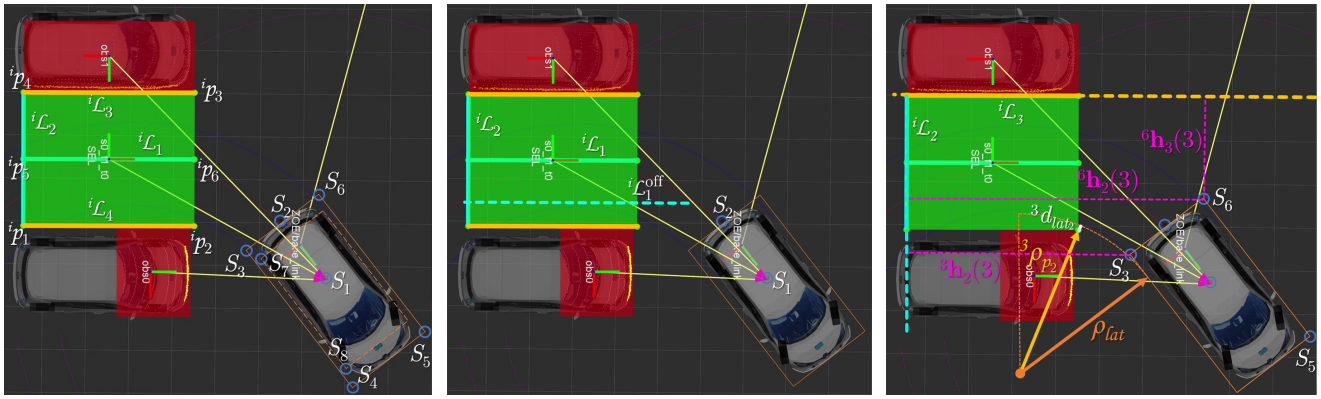
where $\breve{\mathbf{L}}$ and $\breve{\mathbf{T}}_m$ are obtained by concatenating either diagonally or vertically, respectively, matrices $\breve{\mathbf{L}}_i$ and ${}^i\breve{\mathbf{T}}_m$ $\forall\, i \in [1 \ \ldots \ k]$.

Planar world assumption: Assuming that the vehicle to which the sensors are rigidly attached evolves in a plane and that the sensors and vehicle have vertical parallel z axes, all the twists are reduced to $[v_{x_i}, v_{y_i}, \dot{\theta}_i]^\mathsf{T}$ hence the reduced forms $\breve{\mathbf{L}}$, $\breve{\mathbf{L}}_\mathbf{s}$, $\breve{\mathbf{L}}_i$, $\breve{\mathbf{v}}_m$ and ${}^i\breve{\mathbf{T}}_m$ of, respectively, $\breve{\mathbf{L}}$, $\breve{\mathbf{L}}_\mathbf{s}$, $\breve{\mathbf{L}}_i$, $\breve{\mathbf{v}}_m$ and ${}^i\breve{\mathbf{T}}_m$ are considered.

$\breve{\mathbf{L}}_i$ is of dimension $\mathbb{d}_i \times 3$, $\breve{\mathbf{v}}_m = [v_{x_m}, v_{y_m}, \dot{\theta}_m]^\mathsf{T}$ and ${}^i\breve{\mathbf{T}}_m$ is defined as:

$${}^i\breve{\mathbf{T}}_m = \begin{bmatrix} \cos({}^m\theta_i) & \sin({}^m\theta_i) & x_i \sin({}^m\theta_i) - y_i \cos({}^m\theta_i) \\ -\sin({}^m\theta_i) & \cos({}^m\theta_i) & x_i \cos({}^m\theta_i) + y_i \sin({}^m\theta_i) \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

where ${}^m\mathbf{t}_i = [x_i, y_i]^\mathsf{T}$ and ${}^m\theta_i$ are, respectively, the position and orientation of S_i (frame $\mathcal{F}_i$) with respect to $\mathcal{F}_m$ expressed in $\mathcal{F}_m$.



(a) Sensors' configuration and sensor features (b) Task features (c) Constrainted features illustration

Fig. 3. (a) General sensors' configuration and sensor features. (b) Features considered for the parking task. (c) Example of the constrained sensor features

Furthermore, since in the considered model the control frame $\mathcal{F}_m$ is attached to the vehicle's rear axis with origin at the point M (Fig. 1a), it is not possible to generate a velocity along y_m on the vehicle's frame and assuming that there is no slipping nor skidding (i.e. $v_{y_m} = 0$), the robot twist $\check{\mathbf{v}}_m$ can be further reduced to:

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

with $v_{x_m} = v$ and $\dot{\theta}_m = \dot{\theta}$ according to the model (1), thus it is possible to write:

$$\dot{\mathbf{s}} = \mathbf{L_s}\, \mathbf{v}_m \tag{8}$$

where $\mathbf{L_s}$ is composed of the first and third columns of $\check{\mathbf{L}}_\mathbf{s}$.

III. INTERACTION MODEL

For the interaction model, we rely on the perception of several lines $\mathcal{L}_j$ and points from several (virtual) sensors placed at in convenient frames in order to simplify the sensor features definitions and their interaction matrices. The usefulness of virtual sensors can be exemplified as follows: if the car is parking into perpendicular spot with a backward motion (Fig. 3a), the risk of collision with the obstacle on the left is the highest for the car's rear left corner, therefore it would be convenient to have a virtual sensor (S_6) placed on said corner to measure directly the distance to left boundary ($^6\mathcal{L}_3$) of the parking spot.

The sensor's placement can be seen in Fig. 3a. S_1 corresponds to the VLP-16 while S_2 to the 2D LiDAR placed on the rear (LMS151). S_3 to S_6 are placed on the corners of the car's bounding rectangle and have the same orientation as the control frame.

As it can be seen in Fig. 3a, 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 rear boundary 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.

A. Line parametrization

Given two distinct 3D points ^ip_f and ^ip_g in homogeneous coordinates, with

$$^ip_f = [^iX_f, {}^iY_f, {}^iZ_f, {}^iW_f]^\mathsf{T} \tag{9a}$$

$$^ip_g = [^iX_g, {}^iY_g, {}^iZ_g, {}^iW_g]^\mathsf{T}, \tag{9b}$$

a line passing through them can be represented using normalized Plücker coordinates as a couple of 3-vectors [14]:

$$^i\mathcal{L}_j = [^i\underline{\mathbf{u}}_j, {}^i\mathbf{h}_j]^\mathsf{T} \tag{10}$$

where $^i\underline{\mathbf{u}}_j = {}^i\mathbf{u}_j / ||{}^i\mathbf{u}_j||$ (with $^i\mathbf{u}_j \neq 0$) describes the orientation of the line and $^i\mathbf{h}_j = {}^i\mathbf{r}_j / ||{}^i\mathbf{u}_j||$ where $^i\mathbf{r}_j$ encodes the plane containing the line and the origin (*interpretation plane*) and the distance from the origin to the line. The two 3-vectors $^i\mathbf{u}_j$ and $^i\mathbf{r}_j$ are defined as [15]:

$$^i\mathbf{u}_j^\mathsf{T} = {}^iW_f[^iX_g, {}^iY_g, {}^iZ_g] - {}^iW_g[^iX_f, {}^iY_f, {}^iZ_f] \tag{11a}$$

$$^i\mathbf{r}_j^\mathsf{T} = [^iX_f, {}^iY_f, {}^iZ_f] \times [^iX_g, {}^iY_g, {}^iZ_g] \tag{11b}$$

Due to the planar world assumption considered in this paper, the third element of $^i\underline{\mathbf{u}}_j$ and the first and second elements of $^i\mathbf{h}_j$ are equal to zero, i.e. $^i\underline{\mathbf{u}}_j(3) = {}^i\mathbf{h}_j(1) = {}^i\mathbf{h}_j(2) = 0$, therefore the sensor signal $\mathbf{s}_{i_{\mathcal{L}_j}}$ and interaction matrix $\check{\mathbf{L}}_{i_{\mathcal{L}_j}}$ for the line $^i\mathcal{L}_j$ observed by S_i are defined respectively as:

$$\mathbf{s}_{i_{\mathcal{L}_j}} = \left[{}^i\underline{\mathbf{u}}_j(1), {}^i\underline{\mathbf{u}}_j(2), {}^i\mathbf{h}_j(3)\right]^\mathsf{T} \tag{12}$$

$$\check{\mathbf{L}}_{i_{\mathcal{L}_j}} = \begin{bmatrix} 0 & 0 & {}^i\underline{\mathbf{u}}_j(2) \\ 0 & 0 & -{}^i\underline{\mathbf{u}}_j(1) \\ -{}^i\underline{\mathbf{u}}_j(2) & {}^i\underline{\mathbf{u}}_j(1) & 0 \end{bmatrix} \tag{13}$$

B. Task sensor features

The set of task sensor features $\mathbf{s}^t$ is defined as:

$$\mathbf{s}^t = [s_1^t, \ldots, s_9^t]^\mathsf{T} = [\mathbf{s}_1^t, \mathbf{s}_2^t]^\mathsf{T} = [\mathbf{s}_{1_{\mathcal{L}_1^{\text{off}}}}, \mathbf{s}_{2_{\mathcal{L}_1}}, \mathbf{s}_{2_{\mathcal{L}_2}}]^\mathsf{T}, \tag{14}$$

where $^1\mathcal{L}_1^{\text{off}}$ is simply $^1\mathcal{L}_1$ with an offset to the right with respect to the parking spot (Fig. 3b).

The idea behind considering $\mathbf{s}_{1_{\mathcal{L}_1^{\text{off}}}}$ in addition to $\mathbf{s}_2^t$ as part of the set of task sensor features is to have some features that will pull the vehicle out of the parking spot with a forward motion, like a human driver would likely do, in order to escape from local minima therefore being able to park with multiple maneuvers.

The interaction matrix $\check{\mathbf{L}}_1^t$ for the features observed by S_1 is computed at each iteration and is defined by (13) while, for the features observed by S_2, the corresponding interaction matrix $\check{\mathbf{L}}_2^t$ is computed by a 2nd order approximation [16] of the form:

$$\check{\mathbf{L}}^t = \frac{\check{\mathbf{L}}_{\mathcal{L}} + \check{\mathbf{L}}_{\mathcal{L}}^*}{2} \tag{15}$$

where $\check{\mathbf{L}}_{\mathcal{L}} = [\check{\mathbf{L}}_{i_{\mathcal{L}_1}}, \check{\mathbf{L}}_{i_{\mathcal{L}_2}}]^\mathsf{T}$ and $\check{\mathbf{L}}_{\mathcal{L}}^*$ is equal to the value of $\check{\mathbf{L}}_{\mathcal{L}}$ at the desired pose.

Considering the definition of ${}^i\mathcal{L}_1$ and ${}^i\mathcal{L}_2$, a sensible choice would be for ${}^i\mathcal{L}_1^*$ to be collinear with the vehicle's longitudinal axis (x_m-axis) and ${}^i\mathcal{L}_2^*$ to be parallel to y_m-axis at a safe distance from either the rear boundary of the vehicle.

C. Constrained sensor features

The set of constrained sensor features (Fig. 3c) used for collision avoidance $\mathbf{s}^c$ is defined as:

$$\mathbf{s}^c = [s_1^c, \ldots, s_{10}^c]^\mathsf{T} = [\mathbf{s}_3, \mathbf{s}_5, \mathbf{s}_6]^\mathsf{T} \tag{16}$$

with

$$\mathbf{s}_3 = [{}^3\mathbf{h}_2(3), {}^3\mathbf{h}_4(3), {}^3X_2, {}^3Y_2, {}^3d_{lat_2}]^\mathsf{T} \tag{17a}$$

$$\mathbf{s}_5 = {}^5\mathbf{h}_3(3) \tag{17b}$$

$$\mathbf{s}_6 = [{}^6\mathbf{h}_2(3), {}^6\mathbf{h}_3(3), {}^6X_3, {}^6Y_3]^\mathsf{T} \tag{17c}$$

where the difference of raddi ${}^id_{lat_a}$ is defined as:

$${}^id_{lat_a} = {}^i\rho_{p_a} - \rho_{lat}, \tag{18}$$

with:

$${}^i\rho_{p_a} = \sqrt{({}^iX_a + x_i)^2 + ({}^iY_a + y_i - \rho_m)^2}, \tag{19}$$

$$\rho_{lat} = |\rho_m| - \frac{w_{ve}}{2}. \tag{20}$$

The interaction matrices $\check{\mathbf{L}}_{i_{X_a}}$ and $\check{\mathbf{L}}_{i_{Y_a}}$ associated, respectively, to ${}^iX_a$ and ${}^iY_a$ are:

$$\check{\mathbf{L}}_{i_{X_a}} = \begin{bmatrix} -1 & 0 & {}^iY_a \end{bmatrix} \tag{21}$$

$$\check{\mathbf{L}}_{i_{Y_a}} = \begin{bmatrix} 0 & -1 & -{}^iX_a \end{bmatrix} \tag{22}$$

while interaction matrix associated to ${}^id_{lat_a}$ is defined as:

$$\check{\mathbf{L}}_{i_d} = \begin{bmatrix} 0 & \frac{{}^i\varrho_y}{{}^i\rho_{p_a}^2} & \frac{{}^iX_a\,{}^i\varrho_y}{{}^i\rho_{p_a}^2} \end{bmatrix} \tag{23}$$

with ${}^i\varrho_y = -|{}^iY_a + y_i - \rho_m|$. The interaction matrices associated to the rest of the features used as constraints can be deduced from the third row of (13).

The corresponding interaction matrix $\check{\mathbf{L}}_{\mathbf{s}}^c$ is computed at each iteration.

It should be noted that some constraints must be deactivated under certain conditions in order to be able to park successfully. For instance, the constraints on ${}^3X_2$ and ${}^6X_3$ are used to avoid collision, respectively, with points ${}^3p_2$ and ${}^6p_3$, but they would prevent the vehicle from entering the parking spot if they remain active all the time. Thus, if the vehicle is in a configuration where it can safely enter the parking spot without colliding with the aforementioned points, the previously mentioned constraints should be deactivated. Some other constraints must be deactivated under certain circumstances in order to ensure a successful, collision-free parking maneuver. The equations detailing the deactivation conditions (relying only on the sensor features and control signals) used to obtain the results presented in this work can be found in the appendix.

IV. CONTROL

The control input of the robotized vehicle is defined as:

$$\mathbf{v}_r = [v, \phi]^\mathsf{T} \tag{24}$$

with ϕ, considering (1) and (2), being mapped to $\dot{\theta}$ by

$$\dot{\theta} = \frac{v}{\rho_m}. \tag{25}$$

The MSBPC approach being explored is based on the Visual Predictive Control (VPC) described in [17].

A. Structure

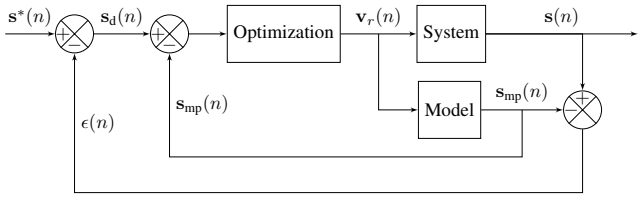


Fig. 4. Control structure [17]

The control structure is based on the internal-model-control (IMC) structure [18] (Fig. 4). The system block contains the robotized vehicle system and sensors whose input is the control variable $\mathbf{v}_r$ and output $\mathbf{s}$ is the current value of the sensor features. The reference $\mathbf{s}^*$ is the desired value of the task sensor features. The error signal ϵ represents all the modeling errors and disturbances between the current features and the values that were predicted from the model:

$$\epsilon(n) = \mathbf{s}(n) - \mathbf{s}_{\text{mp}}(n) \tag{26}$$

where n is the current time.

The optimization algorithm minimizes the difference between the desired value $\mathbf{s}_{\text{d}}$ and the predicted model output $\mathbf{s}_{\text{mp}}$. According to Fig. 4:

$$\mathbf{s}_{\text{d}}(n) = \mathbf{s}^*(n) - \epsilon(n) = \mathbf{s}^*(n) - (\mathbf{s}(n) - \mathbf{s}_{\text{mp}}(n)) \tag{27}$$

from where it is possible to deduce

$$\mathbf{s}_{\text{d}}(n) - \mathbf{s}_{\text{mp}}(n) = \mathbf{s}^*(n) - \mathbf{s}(n) \tag{28}$$

Therefore, to track $\mathbf{s}^*$ by $\mathbf{s}$ is equivalent to track $\mathbf{s}_{\text{d}}$ by $\mathbf{s}_{\text{mp}}$.

To predict the behavior of $\mathbf{s}_{\text{mp}}$ over a finite prediction horizon N_p, the interaction model described in Sec. III is

used. The difference between $\mathbf{s}_{\text{d}}$ and $\mathbf{s}_{\text{mp}}$ is used to define a cost function J to be minimized with respect to a control sequence $\tilde{\mathbf{v}}_r$ over N_p. It should be noted that only the first component $\mathbf{v}_r(n)$ of the optimal control sequence is actually applied to the vehicle.

B. Constraint handling

Model-predictive-control strategies are capable of explicitly take into into account constraints in the control-law design.

The longitudinal velocity v and steering angle ϕ are bounded by its maximum values as follows:

$$|v| < v_{\text{max}} \tag{29a}$$

$$|\phi| < \phi_{\text{max}} \tag{29b}$$

where v_{max} is an adaptive saturation value imposing a deceleration profile based on the velocity profile shown in [4] as the vehicle approaches the final pose. Furthermore, to avoid large changes in the control signals at the current iteration n that may cause uncomfortable sensations for the passengers or surrounding witnesses and, to consider to some extent the dynamic limitations of the vehicle, the control signals are saturated as well by some increments with respect to the previous control signals (at iteration $n-1$) as shown below:

$$(v_{n-1} - \Delta_{dec}) \leq v_n \leq (v_{n-1} + \Delta_{acc}) \tag{30a}$$

$$(\phi_{n-1} - \Delta_{\phi}) \leq \phi_n \leq (\phi_{n-1} + \Delta_{\phi}). \tag{30b}$$

$$(\dot{\phi}_{n-1} - \Delta_{\dot{\phi}}) \leq \dot{\phi}_n \leq (\dot{\phi}_{n-1} + \Delta_{\dot{\phi}}). \tag{30c}$$

The sensor features considered for collision avoidance (16) are constrained as follows:

$$\mathbf{s}^c_{\text{min}} \leq \mathbf{s}^c \leq \mathbf{s}^c_{\text{max}} \tag{31}$$

By writing the constraints (30) and (31) as nonlinear functions:

$$C(\mathbf{v}_r) \leq 0 \tag{32}$$

a constraint domain $\mathbb{C}$ can be defined.

C. Mathematical formulation

The MSBPC approach can be written in discrete time as follows:

$$\begin{aligned} &\min\ J(\mathbf{v}_r) \\ &\tilde{\mathbf{v}}_r \in \mathbb{C} \end{aligned} \tag{33}$$

with

$$J(\mathbf{v}_r) = \sum_{j=n+1}^{n+N_p} [\mathbf{s}_{\text{d}} - \mathbf{s}^t_{\text{mp}}(j)]^T \mathbf{Q}(j) [\mathbf{s}_{\text{d}} - \mathbf{s}^t_{\text{mp}}(j)] \tag{34}$$

and

$$\tilde{\mathbf{v}}_r = \{\mathbf{v}_r(n), \mathbf{v}_r(n+1), \ldots, \mathbf{v}_r(n+N_c), \ldots, \mathbf{v}_r(n+N_p-1)\} \tag{35}$$

subject to

$$\mathbf{s}^t_{\text{mp}}(j) = \mathbf{s}^t_{\text{mp}}(j-1) + \mathbf{L}^t_{\mathbf{s}}(j-1) T_s \mathbf{v}_m(j-1) \tag{36a}$$

$$\mathbf{s}^c_{\text{mp}}(j) = \mathbf{s}^c_{\text{mp}}(j-1) + \mathbf{L}^c_{\mathbf{s}}(j-1) T_s \mathbf{v}_m(j-1) \tag{36b}$$

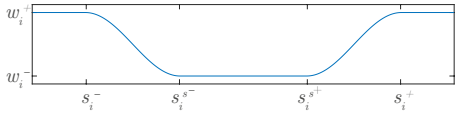


Fig. 5. Weighting function w_i^t

The weighted matrix $\mathbf{Q}$ remains constant along the prediction horizon and, in order to automatically adapt the influence of each task feature, is defined as:

$$\mathbf{Q} = \left[\begin{array}{c|c} Q_1\, \text{diag}(w^t_1, \ldots, w^t_3) & 0_{3\times 6} \\ \hline 0_{6\times 3} & Q_2\, \text{diag}(w^t_4, \ldots, w^t_9) \end{array}\right] \tag{37}$$

where w^t_1-w^t_3, w^t_6 and w^t_9 are constant while the values of $w^t_i\ \forall i = \{4,5,7,8\}$ and Q_2 are computed using a smooth weighting function (Fig. 5) based on the one presented in [19], while:

$$Q_1 = \begin{cases} 0 & \text{if } \|\mathbf{s}_{2_{\mathcal{L}_1}} - \mathbf{s}^*_{2_{\mathcal{L}_1}}\| < \epsilon_{\mathcal{L}_1} \\ 1 - Q_2 & \text{otherwise} \end{cases} \tag{38}$$

where $\epsilon_{\mathcal{L}_1}$ is a small positive scalar value.

Since in the parking scenarios considered, the error $\mathbf{e}^t_1 = \mathbf{s}^t_1 - \mathbf{s}^{t^*}_1$ would be generally minimized with a forward motion (particularly when the vehicle is close to the boundaries of the parking spot) while $\mathbf{e}^t_2 = \mathbf{s}^t_2 - \mathbf{s}^{t^*}_2$ with a backward one, by regulating the influence of each set of sensor features (by means of Q_1 and Q_2, respectively) the controller can automatically maneuver the vehicle with the appropriate direction of motion that would allow to have a successful parking maneuver. Regarding the use of $\epsilon_{\mathcal{L}_1}$, it serves to nullify Q_1 (and consequently the influence of $\mathbf{s}^t_1$) when the vehicle is close to be collinear to $\mathcal{L}_1$.

It should be noted that, from $\mathbf{v}_r(n+N_c)$ to $\mathbf{v}_r(n+N_p-1)$, the control input is constant and is equal to $\mathbf{v}_r(n+N_c)$, where N_c is the control horizon.

V. RESULTS

For the results shown in this section, the parameters in Table II are considered. The value of ϕ_{max} corresponds to the maximum steering angle of the real vehicle while the rest of the parameters were determined by empirical testing, nevertheless some guidelines on how to tune them can be given:

- The maximum longitudinal velocity v_{max} and the increments Δ_v, Δ_ϕ and $\Delta_{\dot{\phi}}$ should be large enough so that the vehicle can park in a reasonable amount of time (without a feeling of *sluggishness*) but not so large that the passengers and surrounding witnesses feel unease during the maneuver.
- A larger control horizon N_c allows the system to maneuver the vehicle more freely at the expense of a larger computation effort.
- N_p should be large enough so that a collision-free motion can be guaranteed (i.e. $N_p \geq v_{\text{max}}/\Delta_v$) but small enough to be able to meet the computational time requirements.
- The threshold value $\epsilon_{\mathcal{L}_1}$ used to determine whether or not Q_1 should be equal to zero has influence on the

total number of maneuvers required to park and on the convergence of the controller. In general, a smaller value of $\epsilon_{\mathcal{L}_1}$ enforces a smaller final error at the expense of an increase on the number of maneuvers required to park.

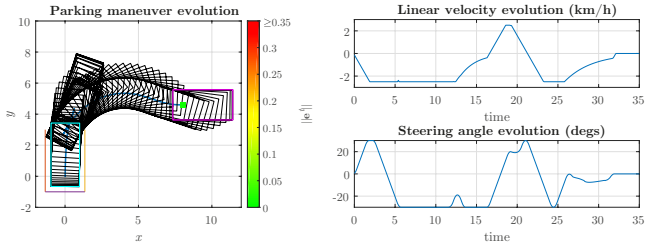
The nonlinear solver used for MATLAB implementations is fmincon with a Sequential Quadratic Programming (SQP) algorithm while for C++ implementations the solver NLopt with a Sequential Least Squares Programming (SLSQP) algorithm is used.

TABLE II

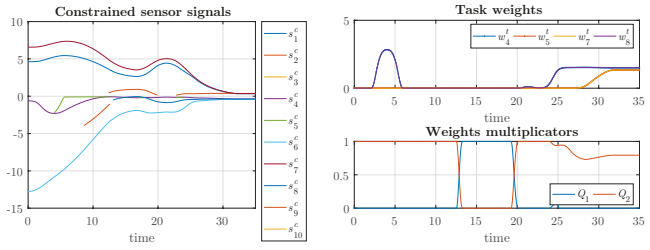
CONTROL-RELATED VEHICLE PARAMETERS

Parameteres	*Notation*	*Value*
Control horizon	N_c	4
Prediction horizon	N_p	20
Sampling time	T_s	0.1s
Maximum steering angle	$\phi_{\max}$	30°
Maximum longitudinal velocity	$v_{\max}$	$\leq$ 0.6944m/s
Maximum velocity increment	Δ_v	0.35m/s T_s
Maximum ϕ increment	Δ_ϕ	2° T_s
Maximum $\dot{\phi}$ increment	$\Delta_{\dot{\phi}}$	0.8 T_s
Threshold value to nullify Q_1	$\epsilon_{\mathcal{L}_1}$	0.125

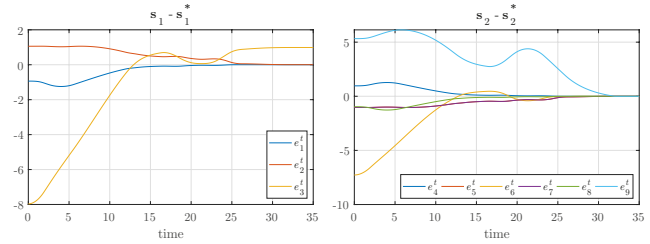
A. MATLAB simulations



(a) Performed maneuver (b) Control signals



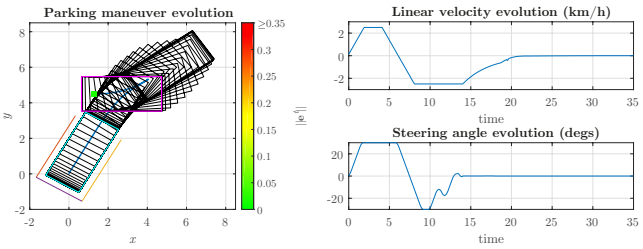
(c) Constrained sensor signals (d) Variable weighting signals



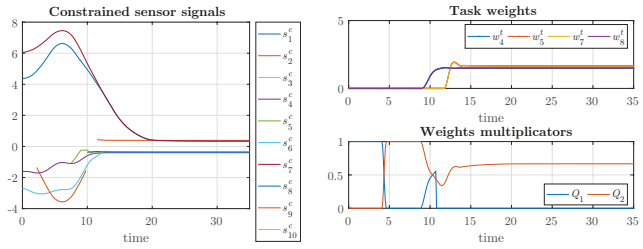
(e) Task error signal: $\mathbf{e}_1^t$ (f) Task error signal: $\mathbf{e}_2^t$

Fig. 6. Backward perpendicular parking maneuver. Initial pose = (8m, 4.6m, 0°)

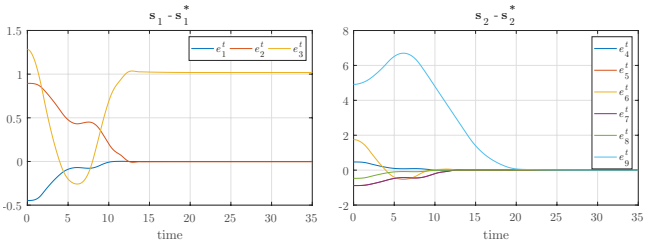
To illustrate the behavior of the MSBPC approach, a perpendicular (Fig. 6) and a diagonal (Fig. 7) maneuvers



(a) Performed maneuver (b) Control signals



(c) Constrained sensor signals (d) Variable weighting signals



(e) Task error signal: $\mathbf{e}_1^t$ (f) Task error signal: $\mathbf{e}_2^t$

Fig. 7. Backward diagonal parking maneuver. Initial pose = (1.3m, 4.5m, 0°)

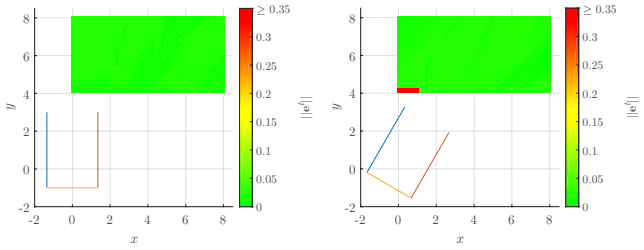
are shown. It can be clearly seen that, for both cases, the car is able to park successfully with generally smooth control signals (thanks to (30)) while satisfying the constraints on the sensor features at each time instant. Furthermore, it can be seen how, generally, when Q_2 is larger than Q_1, the vehicle is moving backward and when Q_1 is larger a transition towards a forward motion occurs, allowing the vehicle to perform multiple maneuvers in order to park successfully.

B. Exhaustive simulations

To assess the stability and convergence of the presented approach, various convergence analyses for the different parking cases were conducted by means of exhaustive simulations. Due to paper length constraints, for the two shown cases (Figs. 8a-8b), the initial orientation of the vehicle is 0°.

Since the exhaustive simulations are an aggregation of the results obtained from several simulations (like those shown in Figs. 6a and 7a), each figure consists of a parking spot (represented by 3 lines) adapted to each case and a scatter plot of the initial position of the vehicle (with a sampling step of 10cm), whose color depends on the final value of $||\mathbf{e}^t||$. The green portion of each scatter plot corresponds to the region of attraction (ROA) and the red one represents the initial positions that are outside of the ROA.

It can be clearly seen that, thanks to the capability of the MSBPC approach of performing automatically multiple



(a) Backward perpendicular case. (b) Backward diagonal case.

Fig. 8. Exhaustive simulations. Initial orientation = 0°. Parking spot length = 4m and width = 2.7m

maneuvers, the car is able to park from almost any initial position in the analysis window with the exception of a small portion on the diagonal case (Fig. 7a) where the vehicle is already violating the constraints from the initial position.

C. Fast prototyping environment

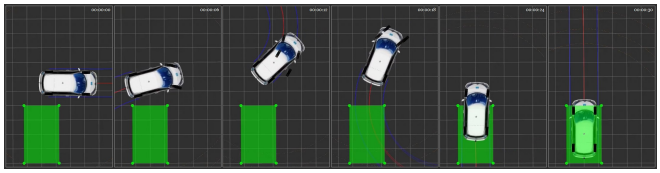
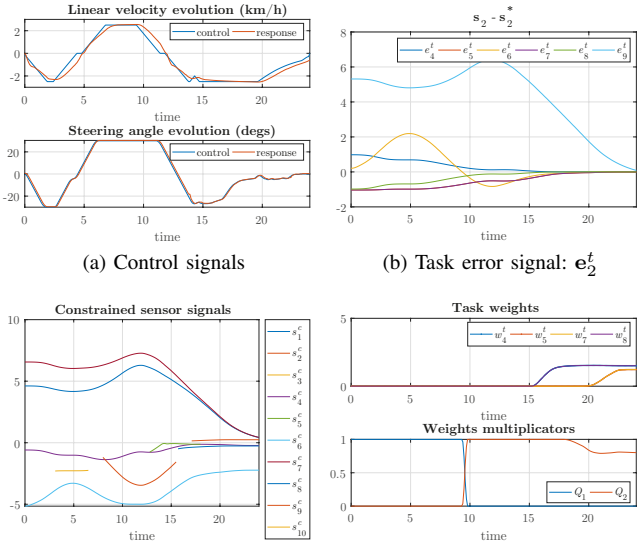


Fig. 9. Backward perpendicular parking maneuver in simulation using a homemade fast prototyping environment



(a) Control signals (b) Task error signal: $\mathbf{e}_2^t$

(c) Constrained sensor signals (d) Variable weighting signals

Fig. 10. Backward perpendicular parking maneuver signals

A homemade fast prototyping environment using the same software architecture as the one embedded inside the car is used for simulation purposes. In addition to behaving nearly identically (from a software architecture point of view) to the real vehicle, this fast prototyping environment simulates as well the dynamics of the vehicle, leading to more realistic simulations than the MATLAB environment used for the results presented in the previous subsections.

As it can be seen in Figs. 9-10, the car is able to park successfully into the parking spot (represented by a green rectangle) in three motions while satisfying the constraints during the whole maneuver, with the evolution of the many different signals being very similar to the MATLAB cases in spite of the slight discrepancy between the control signals and the response of the vehicle (Fig. 10a). The fast deceleration at the end (Fig. 10a) is due to a stopping condition in the implementation related to $\mathbf{e}_t$.

VI. CONCLUSIONS

Following our previous work [12], we've shown how the use of a prediction step makes possible to overcome the main limitation of the previously presented Multi-Sensor-Based control approach - being able to park with only one maneuver. Indeed, thanks to the prediction step considered, the presented MSBPC approach is able to successfully deal with backward perpendicular and diagonal parking problems (using the same formalism) in multiple motions from virtually any sensible initial position.

It is worth noting that the modifications in the interaction model with respect to the MSBC approach are minor.

APPENDIX

The constraints deactivation conditions used to obtain the results presented in this work are now detailed (Table III). To simplify the content of the table, the following notation is considered: subscripts $_{\text{min}}$ denotes a minimum radius when turning with the maximum steering angle (ϕ_{max}), ${}^i p_a^{\text{Cart}}$ describes the point ${}^i p_a$ in Cartesian coordinates, the superscript ${}^{\text{c(angle)}}$ denotes a multiplication of the base by $\cos(\text{angle})$ with angle expressed in degrees and, ϵ_{long} and ϵ_{lat} are small positive values considered for constraints that are mostly related to, respectively, the longitudinal or lateral motions ($\epsilon_{\text{long}} = 0.05$ and $\epsilon_{\text{lat}} = 0.1$). Furthermore, it should be noted that the conditions should be verified at each prediction step along the whole prediction horizon with the appropriate predicted value for each feature and corresponding control signal.

TABLE III
CONSTRAINTS DEACTIVATION CONDITIONS

Constraint	*Deactivate if*
${}^3\mathbf{h}_4(3)$	$!({}^3Y_2 < 0$ and ${}^6Y_3 > 0)$ or ${}^3X_2 < 0$
${}^3X_2$	${}^3x_2 < -2v_{\text{max}}^{\text{abs}}$ or ${}^3Y_2 < -\epsilon_{\text{long}}$
${}^3d_{lat_2}$	$\phi \geq 0$ or $(v < 0$ and ${}^3X_2 > -x_i)$ or $(\lvert{}^5\mathbf{h}_4(3)\rvert > \rho^{\text{c45}}_{m_{\text{min}}}$ and ${}^3p_2^{\text{Cart}} > \rho^{\text{c45}}_{m_{\text{min}}})$
${}^6\mathbf{h}_3(3)$	${}^3Y_3 < -\epsilon_{\text{lat}}$ or $({}^6Y_3 < \epsilon_{\text{lat}}$ and ${}^6X_3 < 0)$ or $({}^6X_3 > 0$ and ${}^3Y_3 < 0)$
${}^6X_3$	${}^3Y_3 < 0$ or ${}^6Y_3 > \epsilon_{\text{long}}$ or $({}^3\mathbf{h}_4(3) > 0$ and ${}^3Y_3 < 0)$
${}^6Y_3$	${}^5X_3 > 2v_{\text{max}}^{\text{abs}}$ or ${}^6X_3 < 0$ or ${}^6Y_3 > \epsilon_{\text{lat}}$

ACKNOWLEDGMENT

This work was supported by the Mexican National Council for Science and Technology (CONACYT). This paper des-

cribes work carried out in the framework of the Valet project, reference ANR-15-CE22-0013-02.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] Y. Yi, Z. Lu, Q. Xin, L. Jinzhou, L. Yijin, and W. Jianhang, "Smooth path planning for autonomous parking system," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, no. Iv. IEEE, 2017, pp. 167–173.

[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] 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.

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

[11] 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.

[12] D. Pérez-Morales, O. Kermorgant, S. Domínguez-Quijada, and P. Martinet, "Laser-Based Control Law For Autonomous Parallel And Perpendicular Parking," in *Second IEEE International Conference on Robotic Computing*, Laguna Hills, CA, 2018, pp. 65–71.

[13] F. Chaumette and S. Hutchinson, "Visual servo control, part I : Basic Approaches," *IEEE Robotics Automation Magazine*, vol. 13, no. December, pp. 82–90, 2006.

[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] B. Přibyl, P. Zemčík, and M. Čadik, "Camera Pose Estimation from Lines using Plücker Coordinates," in *Procedings of the British Machine Vision Conference 2015*. British Machine Vision Association, 2015, pp. 45.1–45.12.

[16] O. Thari and Y. Mezouar, "On the efficient second order minimization and image-based visual servoing," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, may 2008, pp. 3213–3218.

[17] G. Allibert, E. Courtial, and F. Chaumette, "Predictive Control for Constrained Image-Based Visual Servoing," *IEEE Trans. on Robotics*, vol. 26, no. 5, pp. 933–939, 2010.

[18] M. Morari and E. Zafiriou, *Robust Process Control*. Englewood Cliffs, New Jersey: Prentice Hall, 1989.

[19] 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.