Multi-Sensor-Based Predictive Control for Autonomous Parking in Presence of Pedestrians

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 in order to have constraint-based backward non-parallel (perpendicular and diagonal) parking maneuvers capable of dealing with moving pedestrians and, if necessary, performing multiple maneuvers. Our technique relies solely in sensor data expressed relative to the vehicle and therefore no localization is inherently required. Since the proposed approach does not plan any path and instead the controller maneuvers the vehicle directly, the classical path planning related issues are avoided. Real experimentation validates the effectiveness of our approach.**

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. 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], using continuous curvature rate steering functions based on cubic spirals [7]; heuristic approaches [8] and machine learning techniques [9].

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. 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 [10], dynamic obstacle avoidance [11] and for parking applications [12], [13]. 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 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 and pedestrians, it is possible to park without any path planning using a 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 in order to have constraint-based parking maneuvers capable of dealing with moving pedestrians. By extending our previous work [13], now considering an additional auxiliary subtask and a predictive approach, the presented technique is capable of performing multiple maneuvers (if necessary) in order to park successfully in constrained workspaces. The auxiliary subtask is key since it allows to account for the potential motions that go essentially against the final goal (i.e drive the vehicle away from the parking spot) but that in the end allow to park successfully. Moreover, by formalizing pedestrians as constrained moving sensor features, our approach is able to adjust itself to the motion of the moving obstacle in order to avoid colliding with it. Finally, real experimentation demonstrates that our MSBPC technique is able to perform safe parking maneuvers without any path planning, even when a pedestrian purposefully interferes with the vehicle's motion.

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 obtained results are presented in Section V. Finally, some conclusions are given in Section VI.

[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

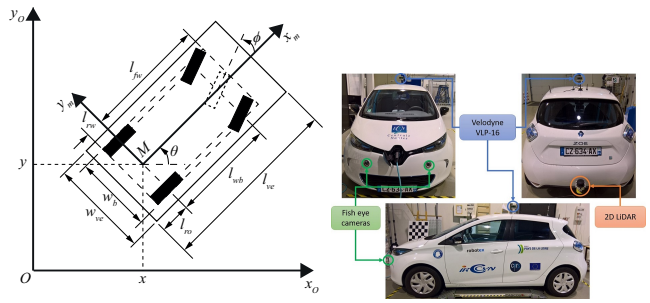
[4] Philippe Martinet is with INRIA Sophia Antipolis, 2004 Route des Lucioles, 06902 Valbonne, France

[1] David.PerezMorales@ls2n.fr

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

[3] Salvador.DominguezQuijada@ls2n.fr

[4] Philippe.Martinet@inria.fr



(a) (b)

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

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

Considering the well-known kinematic model of a car with rear-wheel driving [14], the vehicle's twist is defined by the following column vector (elements separated by a semicolon):

$$\mathbf{v}_m = [v_{x_m}; \dot{\theta}_m], \tag{1}$$

where v_{x_m} and $\dot{\theta}_m$ are, respectively the longitudinal (along x_m) and rotational velocities expressed in the moving base frame $\mathcal{F}_m$. Additionally, one can link the steering angle ϕ to $\dot{\theta}_m$ using the following equation:

$$\dot{\theta}_m = \frac{v_{x_m} \tan\phi}{l_{wb}}. \tag{2}$$

Therefore, it is possible to consider as control input of the robotized vehicle the following expression:

$$\mathbf{v}_r = [v_{x_m}; \phi] \tag{3}$$

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

$$\rho_m = \frac{l_{wb}}{\tan\phi}. \tag{4}$$

It should be noted that, thanks to the multi-sensor-based formalism considered (introduced in the next subsection), our closed-loop control law does not need to have any knowledge about the Cartesian pose of the vehicle (x,y,θ).

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

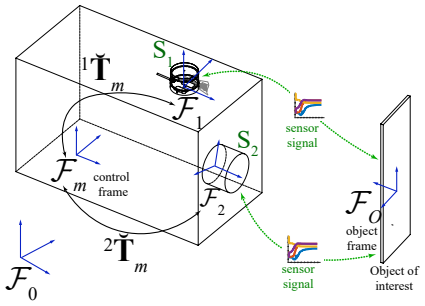


Fig. 2. Multi-sensor model

B. Multi-sensor modeling

For the sake of clarity, the considered multi-sensor modeling (detailed in [15]) 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:

$$\dot{\mathbf{s}}_i = \check{\mathbf{L}}_i \check{\mathbf{v}}_i = \check{\mathbf{L}}_i {}^i\check{\mathbf{T}}_m \check{\mathbf{v}}_m \tag{5}$$

where $\check{\mathbf{L}}_i$ is the interaction matrix [16] of $\mathbf{s}_i$ ($\dim(\check{\mathbf{L}}_i) = \mathbb{d}_i \times 6$) and ${}^i\check{\mathbf{T}}_m$ is the 3D screw transformation matrix that expresses the sensor twist $\check{\mathbf{v}}_i$ (which is expressed in its corresponding frame $\mathcal{F}_i$) with respect to the robot twist $\check{\mathbf{v}}_m$ (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:

$$\dot{\mathbf{s}} = \check{\mathbf{L}}_\mathbf{s} \check{\mathbf{v}}_m \tag{6}$$

with:

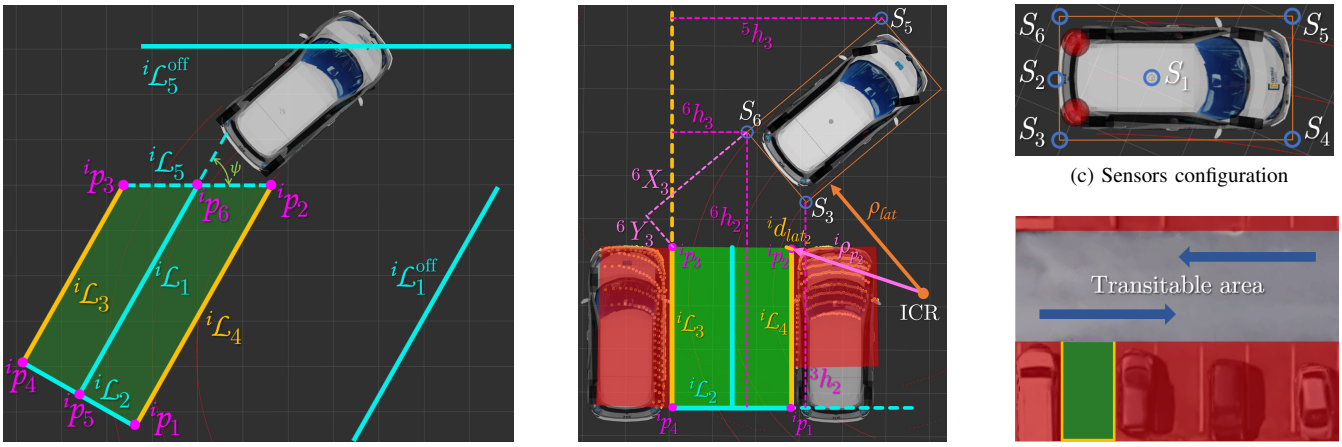
$$\check{\mathbf{L}}_\mathbf{s} = \check{\mathbf{L}} \check{\mathbf{T}}_m \tag{7}$$

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

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 $[v_{x_i}; v_{y_i}; \dot{\theta}_i]$ hence the reduced forms $\check{\mathbf{L}}$, $\check{\mathbf{L}}_\mathbf{s}$, $\check{\mathbf{L}}_i$, $\check{\mathbf{v}}_m$ and ${}^i\check{\mathbf{T}}_m$ of, respectively, $\check{\mathbf{L}}$, $\check{\mathbf{L}}_\mathbf{s}$, $\check{\mathbf{L}}_i$, $\check{\mathbf{v}}_m$ and ${}^i\check{\mathbf{T}}_m$ are considered. $\check{\mathbf{L}}_i$ is of dimension $\mathbb{d}_i \times 3$, $\check{\mathbf{v}}_m = [v_{x_m}; v_{y_m}; \dot{\theta}_m]$ and ${}^i\check{\mathbf{T}}_m$ is defined as:

$${}^i\check{\mathbf{T}}_m = \begin{bmatrix} \cos({}^m\theta_i) & \sin({}^m\theta_i) & x_i \sin({}^m\theta_i) - y_i \cos({}^m\theta_i) \\ -\sin({}^m\theta_i) & \cos({}^m\theta_i) & x_i \cos({}^m\theta_i) + y_i \sin({}^m\theta_i) \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

where ${}^m\mathbf{t}_i = [x_i; y_i]$ 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 $\mathcal{F}_m$. Furthermore, since in the considered model the control frame $\mathcal{F}_m$ is attached to the vehicle's rear axle



(a) Task features (b) Constrained features illustration (c) Sensors configuration (d) Parking environment example

Fig. 3. (a) Features considered for the parking task. (b) Example of the constrained sensor features. (c) General sensors' configuration and sensor features. (d) Example of a parking environment. The green rectangle denotes the parking spot into which the car should park. Red areas are considered forbidden zones, as such the vehicle should never go into them. Furthermore, it is considered that parking maneuvers can only start inside the transitable area and if no portion of the vehicle is inside any of the forbidden zones.

with origin at the point M (Fig. 1a), the robot twist $\check{\mathbf{v}}_m$ can be further reduced to (1). Thus, it is possible to write:

$$\dot{\mathbf{s}} = \mathbf{L_s}\, \mathbf{v}_m \tag{9}$$

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 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 express directly the distance to left boundary (${}^6\mathcal{L}_3$) of the parking spot.

The virtual sensor's placement can be seen in Fig. 3c. S_1 and S_2 are placed along x_m and at a certain distance of the rear vehicle's axle towards the front and rear respectively. 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 ${}^ip_1$ to ${}^ip_4$ correspond to the corners of the parking spot while ${}^ip_5$ and ${}^ip_6$ are, respectively, the midpoints between $({}^ip_1, {}^ip_4)$ and $({}^ip_2, {}^ip_3)$. Line ${}^i\mathcal{L}_1$ is placed along the main axis of the parking spot while lines ${}^i\mathcal{L}_2$ to ${}^i\mathcal{L}_4$ are placed around the edges of the parking spot, leaving one side open from where the vehicle to park can enter the spot (denoted by ${}^i\mathcal{L}_5$). Line ${}^i\mathcal{L}_2$ corresponds to the rear boundary of the parking spot. Lines ${}^i\mathcal{L}_3$ and ${}^i\mathcal{L}_4$ correspond to the lateral boundaries. Additionally, ${}^i\mathcal{L}_1^{\text{off}}$ and ${}^i\mathcal{L}_5^{\text{off}}$ are simply offsets of, respectively, ${}^i\mathcal{L}_1$ and ${}^i\mathcal{L}_5$ away from the parking spot. All the lines are parametrized using normalized Plücker coordinates.

A. Line parametrization

Given two distinct 3D points ${}^ip_f$ and ${}^ip_g$ expressed in frame $\mathcal{F}_i$ of sensor S_i in homogeneous coordinates, with

$${}^ip_f = [{}^iX_f; {}^iY_f; {}^iZ_f; {}^iW_f] \tag{10a}$$

$${}^ip_g = [{}^iX_g; {}^iY_g; {}^iZ_g; {}^iW_g], \tag{10b}$$

a line passing through them (expressed in the same frame $\mathcal{F}_i$) can be represented using normalized Plücker coordinates as a couple of 3-vectors [17]:

$${}^i\mathcal{L}_j = [{}^i\underline{\mathbf{u}}_j; {}^i\mathbf{h}_j] \tag{11}$$

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 position of the line in space. Additionally, ${}^i\mathbf{h}_j$ is orthogonal to the plane containing the line and the origin (*interpretation plane*). The two 3-vectors ${}^i\mathbf{u}_j$ and ${}^i\mathbf{r}_j$ are defined as [18]:

$${}^i\mathbf{u}_j = {}^iW_f[{}^iX_g; {}^iY_g; {}^iZ_g] - {}^iW_g[{}^iX_f; {}^iY_f; {}^iZ_f] \tag{12a}$$

$${}^i\mathbf{r}_j = [{}^iX_f; {}^iY_f; {}^iZ_f] \times [{}^iX_g; {}^iY_g; {}^iZ_g] \tag{12b}$$

Due to the planar world assumption considered, 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. ${}^i\underline{u}_{j,3} = {}^ih_{j,1} = {}^ih_{j,2} = 0$ while, for the same reason, ${}^ih_{j,3}$ can be interpreted as the signed distance from the origin to the line. Considering this and for the sake of clarity, for the remaining of this article it would be deemed ${}^ih_j \equiv {}^ih_{j,3}$. As such, the sensor signal $\mathbf{s}_{i,j}$ 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:

$$\mathbf{s}_{i,j} = \left[{}^i\underline{u}_{j,1}; {}^i\underline{u}_{j,2}; {}^ih_j\right], \tag{13}$$

$$\check{\mathbf{L}}_{{}^i\mathcal{L}_j} = \begin{bmatrix} 0 & 0 & {}^i\underline{u}_{j,2} \\ 0 & 0 & -{}^i\underline{u}_{j,1} \\ -{}^i\underline{u}_{j,2} & {}^i\underline{u}_{j,1} & 0 \end{bmatrix} \tag{14}$$

B. Task sensor features

The set of task sensor features $\mathbf{s}^t$ (task superscript ^{t} not to be confused with transpose $^\mathsf{T}$) corresponding to the positioning is composed of two opposing tasks subsets: the main task $\mathbf{s}_2^t$ - used to actually drive the car into the parking spot, and the auxiliary task $\mathbf{s}_1^t$ - used to drive the car away from it. It is defined as:

$$\mathbf{s}^t = [\mathbf{s}_2^t; \mathbf{s}_1^t] = [\mathbf{s}_{2,1}^t; \mathbf{s}_{2,2}^t; \mathbf{s}_{1,1^{\text{off}}}^t; \mathbf{s}_{1,5^{\text{off}}}^t] \tag{15}$$

The reasoning for this is that, in practice, often is necessary to perform motions that go against the final goal (i.e drive the vehicle away from the parking spot) but that in the end will allow to have a successful parking maneuver.

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 x_m-axis and ${}^i\mathcal{L}_2^*$ to be parallel to y_m-axis at a safe distance from the rear boundary. Moreover, since the objective of the auxiliary task is to pull the vehicle away from the parking spot, the desired values ${}^1\mathcal{L}_1^{\text{off}^*}$ and ${}^1\mathcal{L}_5^{\text{off}^*}$ are chosen to be collinear with the x_m-axis. The conflicting goals for the orientations of these features are exploited by adapting their influence in function of the current sensor features' values. The consistency of the complete task is ensured by means of the weighting approach introduced in the next section.

The interaction matrix $\check{\mathbf{L}}_2^t$ for the features observed by S_2 is computed by a 2nd order approximation [19] of the form:

$$\check{\mathbf{L}}^t = \frac{\check{\mathbf{L}}_\mathcal{L} + \check{\mathbf{L}}_\mathcal{L}^*}{2} \tag{16}$$

where $\check{\mathbf{L}}_\mathcal{L} = [\check{\mathbf{L}}_{i_{\mathcal{L}_1}}; \check{\mathbf{L}}_{i_{\mathcal{L}_2}}]$ and $\check{\mathbf{L}}_\mathcal{L}^*$ is equal to the value of $\check{\mathbf{L}}_\mathcal{L}$ at the desired pose. As it has been shown previously [13], [20], the use of an interaction matrix of the form (16) for the parking task induces a rather interesting and useful behavior: whenever there is a large error in orientation and positioning (regarding ${}^i\mathcal{L}_1$ for the latter) the vehicle steers away from the parking spot initially giving itself more room to afterwards steer into the parking spot. Since this behavior is not as interesting for the auxiliary task and for simplicity reasons, the interaction matrix $\check{\mathbf{L}}_1^t$ for the features observed by S_1 is computed at each iteration and is defined by (14).

C. Constrained sensor features

Let us consider Fig. 3d as an example of a parking environment in the vicinity of a straight road with the road being labeled as *transitable area*. On the lower section of the figure one can see a row of parking spots. Among these parking spots, the chosen one to park the vehicle is denoted by a green rectangle and its boundaries are denoted by yellow lines leaving one side open from where the vehicle can enter the spot. The sections colored in red denote forbidden zones, i.e. areas that the vehicle should never go into. These forbidden zones may be comprised of other parking spots which may or may not be already occupied (like those on the sides of the selected parking spot and on the top section of the figure) as well as walls, sidewalks, bushes, etc., and thus not actually being part of the road anymore. Given that for parking scenarios one can often expect to have static

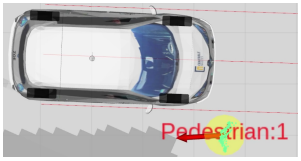


Fig. 4. Perceived pedestrian and its estimated orientation.

obstacles in the red regions of Fig. 3d, it is possible to avoid collision by constraining certain sensor features such that the vehicle is able to move only inside the transitable area and the chosen parking spot. Assuming a straight road, the necessary constraints can be defined considering only sensor features related to the parking spot as shown in this section.

To define the constraints, three other types of features are considered in addition to ${}^i\mathcal{L}_j$: the X and Y coordinates of a given point ${}^ip_a$ (${}^iX_a$, ${}^iY_a$) - which typically are used for the vertices of the parking spot closest to the transitable area, and the difference of radii (shown in Fig. 3b):

$${}^id_{lat_a} = {}^i\rho_{p_a} - \rho_{lat}, \tag{17}$$

where:

$${}^i\rho_{p_a} = \sqrt{({}^iX_a + x_i)^2 + ({}^iY_a + y_i - \rho_m)^2}, \tag{18}$$

$$\rho_{lat} = |\rho_m| - \frac{w_{ve}}{2}. \tag{19}$$

Since (17) already depends on the control input, there is no need to define any interaction matrix for them and instead one can compute/predict such features directly from their definition. 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:

$$\check{\mathbf{L}}_{i_{X_a}} = \begin{bmatrix} -1 & 0 & {}^iY_a \end{bmatrix}, \tag{20}$$

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

Considering that for this type of maneuvers the rear side of the vehicle has to enter first into the parking spot, the majority of the constrained sensor features should be observed by the sensors placed at the rear corners of the vehicle, thus $\mathbf{s}^c$ is defined as follows:

$$\mathbf{s}^c = [s_1^c; \ldots; s_{14}^c] = [\mathbf{s}_3^c; \mathbf{s}_4^c; \mathbf{s}_5^c; \mathbf{s}_6^c] \tag{22}$$

with

$$\mathbf{s}_3^c = [{}^3h_2; {}^3h_4; {}^3h_5; {}^3X_2; {}^3Y_2; {}^3d_{lat_2}], \tag{23a}$$

$$\mathbf{s}_4^c = [{}^4h_4; {}^4h_5], \tag{23b}$$

$$\mathbf{s}_5^c = [{}^5h_3; {}^5h_5], \tag{23c}$$

$$\mathbf{s}_6^c = [{}^6h_2; {}^6h_3; {}^6h_5; {}^6X_3]. \tag{23d}$$

If the appropriate considerations are taken in the interaction model, sensor-based control strategies are capable of tracking moving targets [21], or as in our case, impose constraints on them to avoid collision. (6) can be modified to account for the velocity of a moving obstacle as follows:

$$\dot{\mathbf{s}} = \check{\mathbf{L}}_\mathbf{s}(\check{\mathbf{v}}_m - \check{\mathbf{v}}_{\text{mo}}) \tag{24}$$

where $\check{\mathbf{v}}_{\text{mo}}$ is the twist of the moving obstacle expressed in the vehicle frame (estimated by an extended Kalman filter with constant velocity in the fixed frame).

Without entering in matters of social interaction, the moving obstacle is considered as a pedestrian. Following a classical model in proxemics [22] where the personal distance is defined as a circle of radius of $\approx 46\,\mathrm{cm}$ around an individual, a pedestrian modeled as an inflated point ${}^{i}p_{\mathrm{mo}}$ with a given orientation (Fig. 4). As such, the constrained sensor features can be defined as:

$$\mathbf{s}^{c_{\mathrm{mo}}} = [s_1^{c_{\mathrm{mo}}}; \ldots; s_8^{c_{\mathrm{mo}}}] = [\mathbf{s}_3^{c_{\mathrm{mo}}}; \mathbf{s}_4^{c_{\mathrm{mo}}}; \mathbf{s}_5^{c_{\mathrm{mo}}}; \mathbf{s}_6^{c_{\mathrm{mo}}}] \tag{25}$$

with

$$\mathbf{s}_3^{c_{\mathrm{mo}}} = [{}^{3}X_{\mathrm{mo}}; {}^{3}Y_{\mathrm{mo}}], \tag{26a}$$
$$\mathbf{s}_4^{c_{\mathrm{mo}}} = [{}^{4}X_{\mathrm{mo}}; {}^{4}Y_{\mathrm{mo}}], \tag{26b}$$
$$\mathbf{s}_5^{c_{\mathrm{mo}}} = [{}^{5}X_{\mathrm{mo}}; {}^{5}Y_{\mathrm{mo}}], \tag{26c}$$
$$\mathbf{s}_6^{c_{\mathrm{mo}}} = [{}^{6}X_{\mathrm{mo}}; {}^{6}Y_{\mathrm{mo}}]. \tag{26d}$$

As it can be noticed, the constraints are imposed on features perceived by the sensors placed at the corners of the vehicle's bounding rectangle. By doing so, one can directly constrain the $({}^{i}X_{\mathrm{mo}}, {}^{i}Y_{\mathrm{mo}})$ coordinates of the point associated to the moving pedestrian in order to keep it out of the car's bounding rectangle (with a given safety margin) and thus avoid colliding with it.

Since $\mathbf{s}^c$ may not contain all the elements of each sensor feature required for performing the prediction (especially true for constraints on ${}^{i}h_j$), an extension $\acute{\mathbf{s}}^c$ of $\mathbf{s}^c$ containing all the necessary elements should be considered. The interaction matrix associated to $\acute{\mathbf{s}}^c$, which is computed at each iteration, is denoted as $\acute{\mathbf{L}}_{\mathbf{s}}^c$. The exact definition of the set of constrained sensor features $\mathbf{s}^c$ is now given case by case.

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 ${}^{3}X_2$ and ${}^{6}X_3$ are used to avoid collision, respectively, with points ${}^{3}p_2$ and ${}^{6}p_3$, 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. A similar reasoning applies for the moving obstacle. For example, considering a situation like in Fig. 4, it is only necessary to maintain active the constraints that keep the point on the right and front sides of the vehicle. 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 presented MSBPC approach is based on the Visual Predictive Control (VPC) described in [23] with some modifications to impose an exponential decay of error related to the main task $\mathbf{e}_2^t = \mathbf{s}_2^t - \mathbf{s}_2^{t^*}$. Furthermore, considerations have been made to deal with the nonholonomic constraints inherent to car-like robots in addition to some other constraints (most of them unilateral).

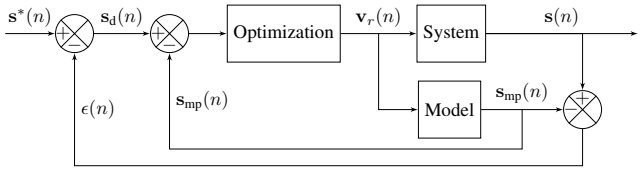


Fig. 5. Control structure [23]

A. Structure

The control structure is based on the internal-model-control (IMC) structure [24] (Fig. 5). 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:

$$\epsilon(n) = \mathbf{s}(n) - \mathbf{s}_{\mathrm{mp}}(n) \tag{27}$$

where n is the current time.

The optimization algorithm minimizes the difference between the desired value $\mathbf{s}_{\mathrm{d}}$ and the predicted model output $\mathbf{s}_{\mathrm{mp}}$. According to Fig. 5:

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

from where it is possible to deduce

$$\mathbf{s}_{\mathrm{d}}(n) - \mathbf{s}_{\mathrm{mp}}(n) = \mathbf{s}^*(n) - \mathbf{s}(n), \tag{29}$$

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

To predict the behavior of $\mathbf{s}_{\mathrm{mp}}$ over a finite prediction horizon N_p, the interaction model described in Sec. III is used. The difference between $\mathbf{s}_{\mathrm{d}}$ and $\mathbf{s}_{\mathrm{mp}}$ 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 at each iteration.

B. Constraint handling

Model-predictive-control strategies are capable of explicitly take into into account constraints in the control-law design. As such, in this subsection we present the imposed constraints.

The longitudinal velocity v_{x_m} and steering angle ϕ are bounded by its maximum values as follows:

$$|v_{x_m}| < v_{\max}, \tag{30a}$$
$$|\phi| < \phi_{\max}, \tag{30b}$$

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 following constraints are considered:

$$|\dot{v}_{x_m}| \leq \dot{v}_{\max}, \tag{31a}$$
$$|\ddot{v}_{x_m}| \leq \ddot{v}_{\max}, \tag{31b}$$
$$|\dot{\phi}| \leq \dot{\phi}_{\max}, \tag{31c}$$
$$|\ddot{\phi}| \leq \ddot{\phi}_{\max}, \tag{31d}$$

$$|\dddot{\phi}| \leq \dddot{\phi}_{\max}. \tag{31e}$$

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

$$\mathbf{s}^{c}_{\min} \leq \mathbf{s}^{c}_{\text{mp}} \leq \mathbf{s}^{c}_{\max} \quad (32a) \qquad \mathbf{s}^{c_{\text{mo}}}_{\min} \leq \mathbf{s}^{c_{\text{mo}}}_{\text{mp}} \leq \mathbf{s}^{c_{\text{mo}}}_{\max} \quad (32b)$$

By writing the constraints (31) and (32) as nonlinear functions:

$$C(\mathbf{v}_r) \leq 0 \tag{33}$$

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

C. Mathematical formulation

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

$$\begin{aligned} &\min\ J(\mathbf{v}_r) \\ &\tilde{\mathbf{v}}_r \in \mathbb{C} \end{aligned} \tag{34}$$

with

$$\begin{aligned} J(\mathbf{v}_r) = \sum_{j=n+1}^{n+N_p} \Big(&[\mathbf{s}_{\text{d}} - \mathbf{s}^{t}_{\text{mp}}(j)]^{\mathsf{T}}\, \mathbf{Q}(j)\, [\mathbf{s}_{\text{d}} - \mathbf{s}^{t}_{\text{mp}}(j)] \\ &+ \mathbf{v}_m(j-1)^{\mathsf{T}}\, \mathbf{R}(j-1)\, \mathbf{v}_m(j-1) \Big) \end{aligned} \tag{35}$$

and

$$\tilde{\mathbf{v}}_r = \{\mathbf{v}_r(n), \mathbf{v}_r(n{+}1), \ldots, \mathbf{v}_r(n{+}N_c), \ldots, \mathbf{v}_r(n{+}N_p{-}1)\} \tag{36}$$

subject to

$$\mathbf{s}^{t}_{\text{mp}}(j) = \mathbf{s}^{t}_{\text{mp}}(j-1) + \mathbf{L}^{t}_{\mathbf{s}}(j-1) T_s \mathbf{v}_m(j-1), \tag{37a}$$

$$\acute{\mathbf{s}}^{c}_{\text{mp}}(j) = \acute{\mathbf{s}}^{c}_{\text{mp}}(j-1) + \acute{\mathbf{L}}^{c}_{\mathbf{s}}(j-1) T_s \mathbf{v}_m(j-1). \tag{37b}$$

$$\check{\mathbf{s}}^{c_{\text{mo}}}_{\text{mp}}(j) = \check{\mathbf{s}}^{c_{\text{mo}}}_{\text{mp}}(j{-}1) + \check{\mathbf{L}}^{c_{\text{mo}}}_{\mathbf{s}}(j{-}1) T_s (\check{\mathbf{v}}_m(j{-}1) - \check{\mathbf{v}}_{\text{mo}}(j{-}1)) \tag{37c}$$

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.

1) Weighting strategy: As mentioned in Sec.III-B, $\mathbf{s}^{t}_{2}$ contains the features used to actually drive the car into the parking spot, thus the minimization of the error $\mathbf{e}^{t}_{2} = \mathbf{s}^{t}_{2} - \mathbf{s}^{t^*}_{2}$ is what drives the vehicle towards a parked pose, generally with a backward motion, while $\mathbf{e}^{t}_{1} = \mathbf{s}^{t}_{1} - \mathbf{s}^{t^*}_{1}$ would be generally minimized with opposite directions of motion. To automatically maneuver the vehicle with the appropriate direction of motion, the influence of each sensor feature is regulated by means of the weighted matrix $\mathbf{Q}$, which remains constant along the prediction horizon. It is defined as:

$$\mathbf{Q} = \left[\begin{array}{c|c} Q_2\, \mathbf{W}^{t}_{2} & 0_{6\times 6} \\ \hline 0_{6\times 6} & Q_1\, \mathbf{W}^{t}_{1} \end{array}\right] \tag{38}$$

with $\mathbf{W}^{t}_{2} = \text{diag}(w^{t}_{1}, \ldots, w^{t}_{6})$, $\mathbf{W}^{t}_{1} = \text{diag}(w^{t}_{7}, \ldots, w^{t}_{12})$ and

$$Q_1 = \begin{cases} 0 & \text{if } \|\mathbf{s}_{2_{\mathcal{L}_1}} - \mathbf{s}^{*}_{2,1_{\mathcal{L}_1}}\| < \epsilon_{\mathcal{L}_1} \text{ and } Q_2 > 0 \\ 1 - Q_2 & \text{otherwise} \end{cases} \tag{39}$$

where $\epsilon_{\mathcal{L}_1}$ is a small positive scalar value that serves to nullify Q_1 (and consequently the influence of $\mathbf{s}^{t}_{1}$) when the vehicle is almost collinear to ${}^{2}\mathcal{L}_1$. The values of $w^{t}_{i}\ \forall i = \{1\text{–}12\}$ and Q_2 are given by the weighting functions $w^{t}_{i}(\mathbf{s}, \mathbf{v}_r(n))$ and $Q_2(\mathbf{s}, \mathbf{v}_r(n))$ which are composed of different combinations of a generic smooth weighting function w (Fig. 6).

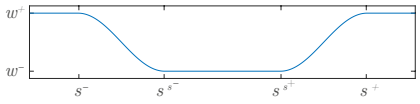


Fig. 6. Generic weighting function $w(s)$

Regarding $\mathbf{W}^{t}_{2}$, its role is to prioritize the error in position by letting the orientation (mostly) free for the most part of the maneuver and, as the current orientation approaches to the desired one, smoothly change the priority from position to orientation. As such, w^{t}_{3} and w^{t}_{6} can be set to constant values while the remaining diagonal elements of $\mathbf{W}^{t}_{2}$ can be defined as:

$$w^{t}_{i} = w(e^{t}_{6})\ \forall i \in \{1, 2, 4, 5\}. \tag{40}$$

As for $\mathbf{W}^{t}_{1}$, its role is to induce small *corrective* motions if the vehicle is relatively close to be collinear to the main axis of the parking spot and otherwise try to drive the vehicle away from the parking spot towards a more suitable *unparked* position as a sort of prepositioning/maneuver-restarting mechanism.

2) Imposing an exponential decay of the error: The inclusion of the term related to $\mathbf{v}_m$ as part of the cost function J allows to impose an exponential decay on the error which additionally improves the stability of the controller. Nevertheless, this behavior (particularly regarding the longitudinal velocity) is mostly only desired for the main parking task. For this purpose, the influence of v_{x_m} and $\dot{\theta}_m$ are regulated by means of the weighted matrix $\mathbf{R}$, which remains constant along the prediction horizon. It is defined as:

$$\mathbf{R} = \begin{bmatrix} \lambda_v Q_2 & 0 \\ 0 & 1 \end{bmatrix}. \tag{41}$$

where λ_v is a constant gain that regulates how fast the vehicle decelerates when approaching the desired parked pose.

V. RESULTS

For the results shown in this section, the parameters in Table II are considered. The value of $\phi_{\max}$ corresponds to the maximum steering angle of the real vehicle while the rest of the parameters required to solve (34) were determined by empirical testing. The controller is implemented in C++ using the solver NLopt [25] with a Sequential Least Squares Programming (SLSQP) algorithm [26]. The whole software architecture (and not only the parking controller) runs on a dual core Intel Core i5-3610ME.

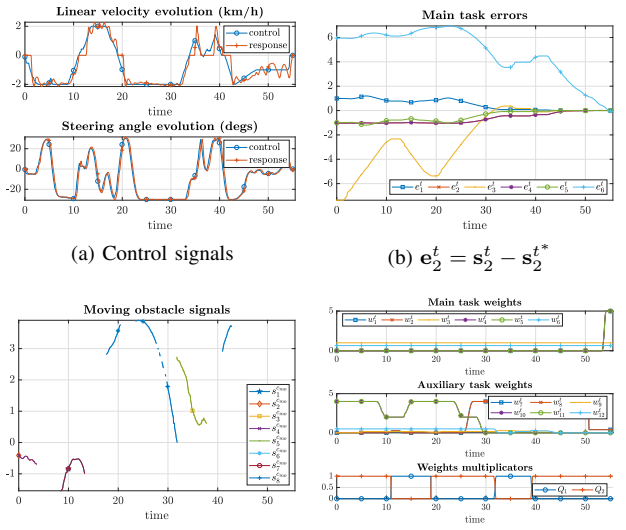
It can be clearly seen that, in spite of the moving pedestrian purposefully disturbing the vehicle, our approach to park successfully (Figs. 7, 8b) while satisfying the collision avoidance constraints during the whole maneuver. The final $||\mathbf{e}^{t}_{2}||$ achieved was 0.0396 which if reconstructed translates to errors of approximately $0.27\,\text{cm}$ laterally, $-3.94\,\text{cm}$ longitudinally and $-0.1°$ in orientation.

TABLE II

CONTROL-RELATED VEHICLE PARAMETERS

Parameteres	*Notation*	*Value*
Sampling time	T_s	$0.1\,\mathrm{s}$
Control horizon	N_c	10 (1 s)
Prediction horizon	N_p	25 (2.5 s)
Maximum longitudinal velocity	$v_{\max}$	$\leq 0.556\,\mathrm{m/s}$
Maximum longitudinal acceleration	$\dot{v}_{\max}$	$0.3\,\mathrm{m/s^2}$
Maximum longitudinal jerk	$\ddot{v}_{\max}$	$0.5\,\mathrm{m/s^3}$
Maximum steering angle	$\phi_{\max}$	$0.5236\,\mathrm{rad}$
Maximum ϕ velocity	$\dot{\phi}_{\max}$	$0.6981\,\mathrm{rad/s}$
Maximum ϕ acceleration	$\ddot{\phi}_{\max}$	$0.9\,\mathrm{rad/s^2}$
Maximum ϕ jerk	$\dddot{\phi}_{\max}$	$0.9\,\mathrm{rad/s^3}$
Longitudinal velocity gain	λ_v	0.1
Threshold value to nullify Q_1	$\epsilon_{\mathcal{L}_1}$	0.125

Fig. 7. Parking in presence of a pedestrian (`https://youtu.be/_Reew0eqg4o`)



(a) Control signals

(b) $\mathbf{e}_2^t = \mathbf{s}_2^t - \mathbf{s}_2^{t^*}$

(c) Constraints on pedestrian

(d) Weights' evolution

Fig. 8. Constrained real backward $\perp$ parking maneuver signals

One can notice that when Q_2 is larger than Q_1 (Fig. 8d), the vehicle is moving backwards and when Q_1 is larger the opposite occurs. The main task error $\mathbf{e}_2^t$ (Fig. 8b) is minimized as the vehicle moves backwards (i.e. towards the parking spot, $Q_2 > Q_1$) while it grows when the vehicle moves in the opposite direction (i.e. away from the parking spot). Regarding the main task weights, it can be seen how the elements related to the orientation have no influence during the most part of the maneuver and only when the vehicle is close to the desired pose the priority changes from position to orientation. As for the auxiliary task weights, it can be seen how a small weight associated s_9^t pulls the vehicle out of the parking spot (Fig. 8d, at $\approx 33\,\mathrm{s}$) while the weights related to s_7^t and s_8^t try to keep the vehicle parallel to main axis of the parking spot. About the evolution of the active constraints on the moving pedestrian (Fig. 8c), one can notice that $s_6^{c_{mo}}$ and $s_8^{c_{mo}}$ (used to keep the pedestrian on the car's left side) get close to zero at $\approx 31\,\mathrm{s}$. This occurred because the pedestrian was purposefully moving diagonally towards the vehicle at that moment, knowing that the risk is lower for him in such context due to the lateral motion limitations of the vehicle. Indeed, if a pedestrian wants to touch the vehicle, it is possible (especially true if approaching from the sides). However, as shown in the rest of the constraints on the pedestrian, the car will always try to avoid contact.

VI. CONCLUSIONS

We have shown that, by moving the effort from the motion planning to the control, the need of having knowledge about the free and occupied space of the whole environment beforehand is eliminated as well as the classical compromise between completeness and computational efficiency. Indeed, thanks to the auxiliary task and the prediction step considered, the presented MSBPC approach is able to successfully deal with non-parallel backward parking problems in multiple motions even when surrounding pedestrians purposefully disturb the vehicle's motion. Moreover, it has been shown that in spite of the disturbances, the amount of maneuvers remains reasonable and the final $||\mathbf{e}_2^t||$ is rather satisfactory.

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 tables, the following notation is considered: w_{spot} denotes the width of the parking spot, $v_{\max}^{\text{abs}}$ denotes the absolute maximum desired velocity in m/s, the subscripts $_{\min^+}$ and $_{\min^-}$ denote a minimum radius or difference of minimum radii when turning with either, respectively, the maximum or minimum steering angle ($\phi_{\max}$ or $-\phi_{\max}$), the superscript $^{\text{c(angle)}}$ denotes a multiplication of the base by $\cos(\text{angle})$ with angle expressed in degrees and $\text{act}(constraint)$ is a boolean variable that is equal to true when the $constraint$ is active and false otherwise and, ϵ_1 to ϵ_5 are small positive values considered for certain constraint deactivation conditions. The superscripts $^{c^-}$ and $^{c^+}$ denote the constraints, respectively, on the low and on the high side of the associated sensor feature. It is worth noting that most sensor features are only constrained on either their low or high side. Thus, if a given constraint does not appear on the table it means that it does not exists. 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.

ACKNOWLEDGMENT

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.

TABLE III

CONSTRAINTS DEACTIVATION CONDITIONS

Constraint	*Deactivate if*
${}^3h_2^{c^-}$	–
${}^3h_4^{c^-}$	${}^3Y_2 \geq 0$ or ${}^6Y_3 \leq 0$ or ${}^3X_2 < 0$
${}^3h_5^{c^-}$	${}^3h_4 > \epsilon_1$ and ${}^3h_3 < -\epsilon_1$
${}^3X_2^{c^-}$	${}^4h_5 > \epsilon_1$ or ${}^4h_4 < 0$ or ${}^3h_5 < 0$ or ${}^3Y_2 < -\epsilon_1$ or ${}^6Y_2 > \epsilon_1$
${}^3X_2^{c^+}$	${}^3X_2 < -2v_{\text{max}}^{\text{abs}}$ or ${}^3Y_2 < -\epsilon_1$ or ${}^3h_3 > 0$ or (act$({}^3h_5^{c^-})$ and $v_{x_m} \leq 0$)
${}^3Y_2^{c^+}$	${}^3h_3 > 0$ or $\left(v_{x_m} \leq 0 \text{ and } \left(\text{act}\left({}^3h_5^{c^-}\right) \text{ or act}\left({}^3d_{lat_2}^{c^+}\right)\right)\right)$ or ${}^3Y_2 > 0$ or ${}^6h_4 < 0$
${}^3d_{lat_2}^{c^+}$	$\phi \geq 0$ or $v_{x_m} \geq 0$ or $\left(v_{x_m} < 0 \text{ and } {}^3X_2 > \lvert x_3 \rvert\right)$ or ${}^3h_5 > \rho^{\text{c30}}_{m_{\text{min}^+}}$ or $\left(\lvert {}^5h_4 \rvert < \rho^{\text{c45}}_{m_{\text{min}^+}} \text{ and } {}^3h_5 > \rho^{\text{c45}}_{m_{\text{min}^+}}\right)$
${}^4h_4^{c^-}$	${}^4h_5 > \epsilon_1$ or ${}^4h_4 < 0$
${}^4h_5^{c^-}$	${}^4h_4 > \epsilon_1$ and ${}^4h_3 < -\epsilon_1$ and ${}^3h_5 < {}^4h_5$
${}^5h_3^{c^+}$	${}^3h_3 > 0$ and ${}^3h_5 > \epsilon_1$
${}^5h_5^{c^+}$	–
${}^6h_2^{c^-}$	–
${}^6h_3^{c^+}$	${}^3Y_3 < -\epsilon_3$ or $({}^6X_3 < 0$ and ${}^6Y_3 < \epsilon_3)$ or $({}^6X_3 > 0$ and ${}^3Y_3 < 0)$
${}^6h_5^{c^+}$	–
${}^6X_3^{c^+}$	${}^3Y_3 < -\epsilon_1$ or ${}^6Y_3 > \epsilon_1$ or $\left({}^3h_3 > 0 \text{ and } {}^3Y_3 < -\epsilon_1\right)$
${}^3X_{\text{mo}}^{c^+}$	${}^3X_{\text{mo}} > 0$ or ${}^3Y_{\text{mo}} < -\epsilon_4$ or ${}^5Y_{\text{mo}} > \epsilon_4$
${}^3Y_{\text{mo}}^{c^+}$	${}^3Y_{\text{mo}} > 0$ or ${}^3X_{\text{mo}} < -\rho_{m_{\text{min}^+}} - 1.5v_{\text{max}}^{\text{abs}}$ or ${}^4X_{\text{mo}} > \rho_{m_{\text{min}^+}} - 1.5v_{\text{max}}^{\text{abs}}$ or $\left({}^3X_{\text{mo}} < -\epsilon_4 \text{ and } ({}^m v_{x_{\text{mo}}} \leq 0 \text{ or } \lvert {}^m v_{y_{\text{mo}}} \rvert > \lvert {}^m v_{x_{\text{mo}}} \rvert)\right)$ or $\left({}^4X_{\text{mo}} > \epsilon_4 \text{ and } ({}^m v_{x_{\text{mo}}} \geq 0 \text{ or } \lvert {}^m v_{y_{\text{mo}}} \rvert > \lvert {}^m v_{x_{\text{mo}}} \rvert)\right)$ or $\left({}^3X_{\text{mo}} > -\epsilon_5 \text{ and } {}^4X_{\text{mo}} < \epsilon_5 \text{ and } {}^3Y_{\text{mo}} > -\epsilon_5 \text{ and } {}^6Y_{\text{mo}} < \epsilon_5\right)$
${}^4X_{\text{mo}}^{c^-}$	${}^4X_{\text{mo}} < 0$ or ${}^3Y_{\text{mo}} < -\epsilon_4$ or ${}^5Y_{\text{mo}} > \epsilon_4$
${}^4Y_{\text{mo}}^{c^+}$	!act $\left({}^3Y_{\text{mo}}^{c^+}\right)$
${}^5X_{\text{mo}}^{c^-}$	!act $\left({}^4X_{\text{mo}}^{c^-}\right)$
${}^5Y_{\text{mo}}^{c^-}$	${}^5Y_{\text{mo}} < 0$ or ${}^3X_{\text{mo}} < -\rho_{m_{\text{min}^+}} - 1.5v_{\text{max}}^{\text{abs}}$ or ${}^4X_{\text{mo}} > \rho_{m_{\text{min}^+}} - 1.5v_{\text{max}}^{\text{abs}}$ or $\left({}^3X_{\text{mo}} < -\epsilon_4 \text{ and } ({}^m v_{x_{\text{mo}}} \leq 0 \text{ or } \lvert {}^m v_{y_{\text{mo}}} \rvert > \lvert {}^m v_{x_{\text{mo}}} \rvert)\right)$ or $\left({}^4X_{\text{mo}} > \epsilon_4 \text{ and } ({}^m v_{x_{\text{mo}}} \geq 0 \text{ or } \lvert {}^m v_{y_{\text{mo}}} \rvert > \lvert {}^m v_{x_{\text{mo}}} \rvert)\right)$ or $\left({}^3X_{\text{mo}} > -\epsilon_5 \text{ and } {}^4X_{\text{mo}} < \epsilon_5 \text{ and } {}^3Y_{\text{mo}} > -\epsilon_5 \text{ and } {}^6Y_{\text{mo}} < \epsilon_5\right)$
${}^6X_{\text{mo}}^{c^+}$	!act $\left({}^3X_{\text{mo}}^{c^+}\right)$
${}^6Y_{\text{mo}}^{c^-}$	!act $\left({}^5Y_{\text{mo}}^{c^-}\right)$

REFERENCES

[1] W. Wang, Y. Song, J. Zhang, and H. Deng, "Automatic parking of vehicles: A review of literatures," *Int. J. Automot. Technol.*, 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 *IEEE Int. Conf. Veh. Electron. Saf.*, 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 Work. Planning, Percept. Navig. Intell. Veh.*, Hamburg, Germany, 2015, pp. 143–148.

[4] P. Petrov and F. Nashashibi, "Saturated Feedback Control for an Automated Parallel Parking Assist System," in *13th Int. Conf. Control. Autom. Robot. Vis.*, 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 *10th IEEE Int. Conf. Networking, Sens. Control*, 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 *IEEE Intell. Veh. Symp.*, 2017, pp. 167–173.

[7] H. Banzhaf, N. Berinpanathan, D. Nienhuser, and J. Marius Zollner, "From G 2 to G 3 Continuity: Continuous Curvature Rate Steering Functions for Sampling-Based Nonholonomic Motion Planning," in *IEEE Intell. Veh. Symp.*, Changshu, China, 2018, pp. 326–333.

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

[9] G. Notomista and M. Botsch, "Maneuver segmentation for autonomous parking based on ensemble learning," in *Int. Jt. Conf. Neural Networks*, Killarney, Ireland, 2015, pp. 1–8.

[10] 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 *IEEE 18th Int. Conf. Intell. Transp. Syst.*, Las Palmas, Spain, 2015, pp. 1791–1796.

[11] 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 *IEEE Intell. Veh. Symp.*, Seoul, South Korea, 2015, pp. 77–82.

[12] D. Pérez Morales, S. Domínguez Quijada, O. Kermorgant, and P. Martinet, "Autonomous parking using a sensor based approach," in *8th Work. Planning, Percept. Navig. Intell. Veh. 19th IEEE ITSC*, Rio de Janeiro, Brazil, 2016, pp. 211–216.

[13] D. Perez-Morales, O. Kermorgant, S. Dominguez-Quijada, and P. Martinet, "Laser-Based Control Law for Autonomous Parallel and Perpendicular Parking," in *Second IEEE Int. Conf. Robot. Comput.*, Laguna Hills, USA, 2018, pp. 64–71.

[14] A. De Luca, G. Oriolo, and C. Samson, "Feedback control of a nonholonomic car-like robot," in *Robot motion Plan. Control*, 1998, pp. 171–253.

[15] O. Kermorgant and F. Chaumette, "Dealing with constraints in sensor-based robot control," *IEEE Trans. Robot.*, vol. 30, no. 1, pp. 244–257, 2014.

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

[17] N. Andreff, B. Espiau, and R. Horaud, "Visual Servoing from Lines," *Int. J. Rob. Res.*, vol. 21, no. 8, pp. 679–699, 2002.

[18] B. Pibyl, P. Zemčík, and M. Čadik, "Camera Pose Estimation from Lines using Plücker Coordinates," in *Procedings Br. Mach. Vis. Conf.* British Machine Vision Association, 2015, pp. 45.1–45.12.

[19] O. Thari and Y. Mezouar, "On the efficient second order minimization and image-based visual servoing," in *IEEE Int. Conf. Robot. Autom.*, 2008, pp. 3213–3218.

[20] D. Pérez-Morales, O. Kermorgant, S. Domínguez-Quijada, and P. Martinet, "Multi-Sensor-Based Predictive Control for Autonomous Backward Perpendicular and Diagonal Parking," in *10th Work. Planning, Percept. Navig. Intell. Veh. IEEE/RSJ IROS*, Madrid, Spain, 2018.

[21] N. Shahriari, S. Fantasia, F. Flacco, and G. Oriolo, "Robotic visual servoing of moving targets," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2013, pp. 77–82.

[22] E. T. Hall, *The Hidden Dimension*. Doubleday, 1966.

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

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

[25] S. G. Johnson, "The NLopt nonlinear-optimization package." [Online]. Available: http://ab-initio.mit.edu/nlopt

[26] D. Kraft, "A Software Package for Sequential Quadratic Programming," Institut fuer Dynamik der Flugsysteme, Oberpfaffenhofen, Tech. Rep., 1988.