



HAL
open science

Multi-Sensor-Based Predictive Control For Autonomous Parking

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

► **To cite this version:**

David Pérez-Morales, Olivier Kermorgant, Salvador Domínguez-Quijada, Philippe Martinet. Multi-Sensor-Based Predictive Control For Autonomous Parking. *IEEE Transactions on Robotics*, 2022, 38 (2), pp.835-851. 10.1109/TRO.2021.3091285. hal-03286432

HAL Id: hal-03286432

<https://inria.hal.science/hal-03286432v1>

Submitted on 14 Jul 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multi-Sensor-Based Predictive Control For Autonomous Parking

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

Abstract—This paper formalizes, under a single common Multi-Sensor-Based Predictive Control framework, five different types of parking maneuvers: perpendicular, diagonal for both forward and backward motions and parallel for backward motions. Since, from a practical point of view, forward parallel parking is usually not advisable, it is not addressed in this work. By moving the effort from motion planning to control, the parking tasks can be completely defined solely from the detected empty parking spots. Additionally, the classical compromise between completeness and computational efficiency when compared to exploration-based path planning techniques is eliminated. The results of a few individual cases are presented and compared against a state of the art path planning approach to illustrate the behavior and performance of the proposed framework as well as results from exhaustive simulations to assess its convergence. As shown in the convergence analyses, the presented approach allows to park from virtually any sensible initial pose. Finally, real experimentation using a robotized Renault ZOE shows the validity and robustness in the convergence domain of the presented approach.

Index Terms—Intelligent vehicles, sensor-based control, model predictive control, motion control, intelligent parking system.

I. INTRODUCTION

A. Motivation and related work

EVEN for experienced drivers, parking can be a difficult task, especially in big cities where the parking spots are often very narrow. The search for an increase in comfort and safety when parking has led to a quite extensive literature [1], having explored many different approaches to automate this bothersome task.

Despite the fact that the automobile industry has already started to roll out some commercial parking assistants able to actively control acceleration, braking and steering [2], the research interest in the topic remains strong.

Path planning approaches for automated vehicles have been heavily investigated in recent years [3]. Among the different planning techniques relevant for parking applications it is possible to distinguish between geometric approaches, with either constant turning radius [4], using saturated feedback controllers [5], or continuous-curvature planning using clothoids [6], [7]. Recent developments on sampling based planners

using hybrid curvature steering functions based on clothoids [8] and using continuous curvature rate steering functions based on cubic spirals [9] have been reported as well. Heuristic techniques [10] and combination of sampling based planner with machine learning algorithms to guide the planner [11] can also be found in the literature, as well as optimization-based refinement of the planned trajectories [12], [13]. It is worth noting that parking maneuvers with forward motions are seldom considered, with [14] for the parallel parking case and [15] for the perpendicular case being some of the few works on this regard.

A well-known drawback of path planning is that it assumes that the whole environment is known beforehand. Also, the planned path must be tracked perfectly which might be difficult because of localization uncertainties (e.g. underground parking lots without any special infrastructure) or accumulated differences between the planned path and the performed one [6], [7]. In such situations, it might be necessary to replan the whole path which is not computationally efficient.

An interesting alternative is the use of a sensor-based control approach. It has been proven to be valid for navigation [16], dynamic obstacle avoidance [17] and for parking applications [18]. It should be noted that an important limitation of a purely sensor-based control approach considering only a straightforward parking task (like in [18]) 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.

B. 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 either the rear or front boundary of the parking spot (respectively for backward or forward parking maneuvers) while avoiding collision with surrounding obstacles during the whole maneuver. However, depending of various factors (e.g. initial pose with respect to the parking spot, size of the vehicle, surrounding obstacles, etc.), often multiple maneuvers are required to park successfully. In such situations it would be necessary to perform motions that go against the final goal (i.e. drive the vehicle away from the desired parked pose) but that in the end will allow to have a successful parking maneuver.

Assuming that the vehicle is capable of perceiving surrounding free parking spots, we will show that it is possible to park using a Multi-Sensor-Based Predictive Control (MSBPC)

Manuscript received ... 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.

D. Pérez-Morales, O. Kermorgant and S. Domínguez-Quijada are with the ARMEN Team, Laboratoire des Sciences du Numérique de Nantes, 1 rue de la Noë, 44321 Nantes, France.

P. Martinet is with INRIA Sophia Antipolis, 2004 Route des Lucioles, 06902 Valbonne, France



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

approach by minimizing the distance of a certain set of sensor features to its corresponding desired value while avoiding collision by imposing certain constraints on another set of sensor features along a given prediction horizon. The cost function should account for the main task that will actually park the vehicle and for the auxiliary task that drives the car away from the parking spot when needed. By exploiting the multi-sensor-based modeling, it is possible to express these two tasks by means of two (one for each task) pairs of 2D lines that can be easily extracted from the free parking spot. We highlight 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 formalization of parking operations under a common MSBPC framework allowing the vehicle to park autonomously into perpendicular and diagonal parking spots with both forward and backward motions and into parallel ones with backward motions. Parking into parallel spots with forward motions was not considered since it is not an advisable maneuver in general, especially when it cannot be done in just one shot. By extending our previous work [18], 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.

C. Content 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 obtained results are presented in Section V-A: a few cases in simulation environments are presented and compared against a state of the art path planning approach, together with exhaustive simulations assessing the convergence performance of the

presented approach. Additionally, Section V-B shows real experimentation with the car passing over a speed bump during the parking maneuver. Finally, some conclusions are given in Section VI

For the sake of clarity, for the equations and explanations presented in this work, the parking spot is considered to be always on the right side of the vehicle to park at the beginning of the maneuver

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 [19], the vehicle's twist is defined by the following column vector (elements separated by a semicolon):

$$\mathbf{v}_m = [v; \dot{\theta}], \quad (1)$$

where v and $\dot{\theta}$ 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 δ to $\dot{\theta}$ using the following equation:

$$\dot{\theta} = \frac{v \tan \delta}{l_{wb}}. \quad (2)$$

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

$$\mathbf{v}_r = [v; \delta] \quad (3)$$

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

$$\rho_m = \frac{l_{wb}}{\tan \delta}. \quad (4)$$

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

B. Multi-sensor-based control

Our approach relies on multi-sensor-based control [20] in order to easily design the control law. In this framework, a robot is assumed to carry several sensors S_i that give information about the environment. In this Section we recall the key concepts of this approach.

Each sensor is associated with a vector of sensor features s_i of dimension \mathfrak{d}_i . Typically, s_i can be the raw sensor signal or any output of a processing algorithm. For instance, the features corresponding to a camera can be the full image or a parametrization of primitives of interest [21]. In practice, the choice of sensor features depends on the considered task and desired behavior of the system.

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

$$\dot{s}_i = \mathbf{L}_i \mathbf{v}_i = \mathbf{L}_i \mathbf{T}_i \mathbf{v}_m \quad (5)$$

where \mathbf{L}_i is the interaction matrix [21] of s_i and \mathbf{T}_i is the screw transformation matrix that expresses the sensor twist \mathbf{v}_i (which is expressed in its corresponding frame \mathcal{F}_i) with respect to the robot twist \mathbf{v}_m , expressed in the mobile frame \mathcal{F}_m which is used as control frame. Classically, sensor-based control considers full free-flying 3D motions (i.e. $\dim(\mathbf{L}_i) = \mathfrak{d}_i \times 6$ and $\dim(\mathbf{T}_i) = 6 \times 6$). As we will see in the next Section, we rely on a simpler, planar model to express the motions of the sensors.

Sensor-based control is about designing a control law which induces the desired behavior in the feature space. This behavior can be twofold:

- Task features, simply denoted as \mathbf{s} , are associated to some desired value \mathbf{s}^* . The control should in this case have \mathbf{s} converge to \mathbf{s}^* .
- Constrained features, now denoted as \mathbf{c} , are associated to some bounds (upper, lower or both). The control should thus ensure that \mathbf{c} is always within its bounds.

We now detail how this approach is adapted to the planar case for autonomous parking.

C. Multi-sensor for autonomous parking

In order to perform autonomous parking, we rely on the perception of several points P_a and lines \mathcal{L}_j that correspond to the vertices and edges of the quadrilateral defined by a given parking spot (or can be straightforwardly obtained from them). Figure 2 shows the convention to number the points and lines of interest. Additional points, for instance corresponding to other obstacles, can be easily considered with the same formalism.

The presented technique does not impose any specific type of sensor as long as the information related to the parking spot (i.e. its vertices and edges) to which the car should park can be extracted. For example, one could approximate a parking spot from the free space between two already parked cars [18] or, if only one car (or none) was already parked, one could instead use the markings on the ground perceived by a vision system. If a precise enough map and localization system are available, another option would be to generate virtual sensory data from the relative pose between the vehicle and the known parking spots. Finally, one could use a combination of different sensor sources to get a more robust estimation of the parking spot and surrounding obstacles.

Regardless of the real sensory system of the autonomous vehicle, the extracted points and lines are expressed as if they were observed by several virtual sensors S_i . As shown in Fig. 1c, these sensors are placed at convenient frames \mathcal{F}_i in order to simplify the sensor features definitions and their interaction matrices. Changing the reference frames of points and lines is a trivial operation that allows easily expressing the task and constrained features.

In our approach, sensors S_1 and S_2 are laterally centered and mainly used for the parking task. Sensors S_3 to S_6 are placed on the corners of the car's bounding rectangle (taking into account the side mirrors) with the purpose of collision avoidance with surrounding obstacles. S_7 and S_8 are used to prevent hitting the curb on parallel parking maneuvers thus are placed on the right side corners (with respect to the vehicle) of the dashed red rectangle whose width and length are equal to w_b and $l_{rw} + l_{fw}$ respectively.

We rely on a planar model hence all sensor frames are aligned with the vehicle frame \mathcal{F}_m and only differ from a position offset (x_i, y_i) . As a consequence, (5) is expressed with lower dimension $\dim(\mathbf{L}_i) = \mathfrak{d}_i \times 3$ and \mathbf{T}_i is defined as:

$$\mathbf{T}_i = \begin{bmatrix} 1 & -y_i \\ 0 & x_i \\ 0 & 1 \end{bmatrix} \quad (6)$$

In this planar case, \mathbf{T}_i thus expresses the induced planar twist of a sensor (of dimension 3), when a control \mathbf{v}_m (1) is applied to the vehicle.

The interest of virtual sensors is first to make the modeling and control independent from the actual way the vehicle perceives the surroundings. It also makes easier to design the task and the constraints. For instance, a feature related to the parking task is to align the main axis \mathcal{L}_1 with the x -axis of \mathcal{F}_1 . Similarly, if the car is parking into perpendicular spot with a backward motion (Fig. 2a), the risk of collision with the obstacle on the left is the highest for the car's rear left corner. This distance is easily expressed by how sensor S_6 sees line \mathcal{L}_3 .

In the next Section we introduce the interaction model of the considered points and lines features.

III. INTERACTION MODEL

In this Section we express how lines and point features are parameterized, and derived their interaction matrix. We also

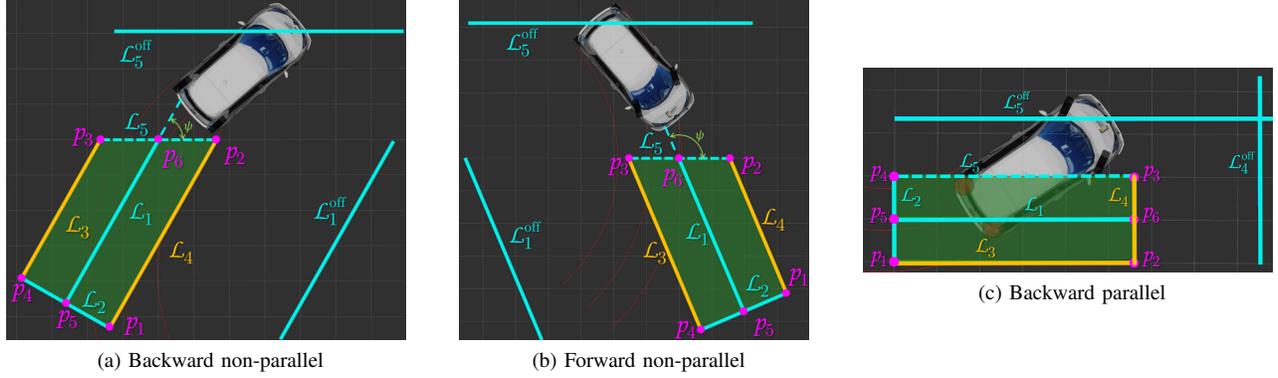


Fig. 2. Parking spot models for backward (a) and forward (b) non-parallel types and for (c) parallel ones.

detail which features are used to describe the parking task and the collision avoidance constraints.

A. Line parametrization

We denote ${}^i P_j$ and ${}^i \mathcal{L}_j$ the coordinates of P_j and \mathcal{L}_j expressed in \mathcal{F}_i . A line ${}^i \mathcal{L}_j$ passing through two distinct 3D points ${}^i P_1$ and ${}^i P_2$ can be represented using Plücker coordinates as a couple of 3-vectors [22], expressed in \mathcal{F}_i :

$$\begin{cases} {}^i \mathbf{u}_j &= [{}^i X_2; {}^i Y_2; {}^i Z_2] - [{}^i X_1; {}^i Y_1; {}^i Z_1] \neq 0 \\ {}^i \mathbf{r}_j &= [{}^i X_1; {}^i Y_1; {}^i Z_1] \times [{}^i X_2; {}^i Y_2; {}^i Z_2] \end{cases} \quad (7)$$

This work relies on normalized coordinates [23] expressed as:

$$\begin{cases} {}^i \underline{\mathbf{u}}_j &= {}^i \mathbf{u}_j / \|{}^i \mathbf{u}_j\| \\ {}^i \mathbf{h}_j &= {}^i \mathbf{r}_j / \|{}^i \mathbf{u}_j\| \end{cases} \quad (8)$$

From the planar world assumption ($Z = 0$), only 3 components in (8) are non-null. They are:

- The first two components of ${}^i \underline{\mathbf{u}}_j$, that correspond to the cosine and sine of the line orientation,
- The last component of ${}^i \mathbf{h}_j$, that is the signed distance from the origin to the line.

In the sequel, a line is thus parameterized from these 3 values denoted as:

$${}^i \mathbf{l}_j = [{}^i \underline{u}_{j,1}; {}^i \underline{u}_{j,2}; {}^i h_j], \quad (9)$$

The corresponding interaction matrix $\mathbf{L}_{i,j}$ can be expressed as [23]:

$$\mathbf{L}_{i,j} = \begin{bmatrix} 0 & 0 & {}^i \underline{u}_{j,2} \\ 0 & 0 & -{}^i \underline{u}_{j,1} \\ -{}^i \underline{u}_{j,2} & {}^i \underline{u}_{j,1} & 0 \end{bmatrix} \quad (10)$$

B. Parking spots models

As it can be seen in Figs. 2a-2c, 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) . Line \mathcal{L}_1 is placed along the main axis of the parking spot while lines \mathcal{L}_2 to \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 \mathcal{L}_5). Line \mathcal{L}_2 corresponds to

the rear and front boundary of the parking spot, respectively, for backward and forward maneuvers. Lines \mathcal{L}_3 and \mathcal{L}_4 correspond to the lateral boundaries on the non-parallel cases. For the parallel case, \mathcal{L}_3 corresponds to the right side lateral boundary while \mathcal{L}_4 corresponds to the front boundary of the parking spot. Additionally, $\mathcal{L}_1^{\text{off}}$, $\mathcal{L}_4^{\text{off}}$ and $\mathcal{L}_5^{\text{off}}$ are simply offsets of, respectively, \mathcal{L}_1 , \mathcal{L}_4 and \mathcal{L}_5 away from the parking spot.

The pair of points through which each line passes are defined in Table II.

TABLE II
PAIR OF POINTS THROUGH WHICH EACH LINE PASSES

Line	Non-parallel	Parallel (backward)
\mathcal{L}_1	(P_5, P_6)	(P_5, P_6)
\mathcal{L}_2	(P_1, P_4)	(P_1, P_4)
\mathcal{L}_3	(P_3, P_4)	(P_1, P_2)
\mathcal{L}_4	(P_1, P_2)	(P_2, P_3)
\mathcal{L}_5	(P_2, P_3)	(P_3, P_4)

C. Task sensor features

The set of task sensor features \mathbf{s} corresponding to the positioning is composed of two opposing tasks subsets: the main task \mathbf{s}_p (used to actually drive the car into the parking spot) and the auxiliary task \mathbf{s}_a (used to drive the car away from the parking spot). Both \mathbf{s}_p and \mathbf{s}_a are composed from two line features:

- \mathbf{s}_p is composed with lines \mathcal{L}_1 and \mathcal{L}_2 , that describe the pose of the parking spot
- \mathbf{s}_a is composed with lines $\mathcal{L}_5^{\text{off}}$ and $\mathcal{L}_1^{\text{off}}$ (parallel maneuvers) or $\mathcal{L}_4^{\text{off}}$ (non-parallel), that describe where to go when stuck during parking

Table III lists the frames where these lines are expressed, depending on the parking configuration.

The next Section details the constrained sensor features, used to model the surrounding obstacles.

D. Constrained sensor features

Let us consider Fig. 3 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

TABLE III
TASK SENSOR FEATURES DEFINITION

Maneuver type	Main task	Auxiliary task
Backward non-parallel	$\mathbf{s}_p = [{}^2\mathbf{1}_1; {}^2\mathbf{1}_2]$	$\mathbf{s}_a = [{}^1\mathbf{1}_{\text{off}}; {}^1\mathbf{1}_{\text{5off}}]$
Forward non-parallel	$\mathbf{s}_p = [{}^1\mathbf{1}_1; {}^1\mathbf{1}_2]$	$\mathbf{s}_a = [{}^2\mathbf{1}_{\text{off}}; {}^2\mathbf{1}_{\text{5off}}]$
Backward parallel	$\mathbf{s}_p = [{}^2\mathbf{1}_1; {}^2\mathbf{1}_2]$	$\mathbf{s}_a = [{}^1\mathbf{1}_{\text{4off}}; {}^1\mathbf{1}_{\text{5off}}]$

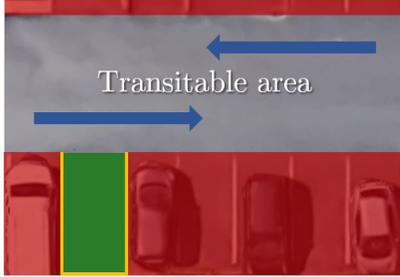


Fig. 3. Example of a parking environment. The green rectangle denotes the parking spot into which the car should park. Red areas are considered forbidden zones, as such the vehicle should never go into them. Furthermore, it is considered that parking maneuvers can only start inside the transitable area and if no portion of the vehicle is inside any of the forbidden zones.

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 obstacles in the red regions of Fig. 3, 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 coordinates of the entry corner points P_a of the parking spot, and signed differences of distances to the ICR. With the ICR denoted C , the distances of interest are denoted as:

$$\begin{cases} \rho_{lat} = |\rho_m| - \frac{w_{ve}}{2} & \text{Lateral distance to the ICR} \\ \rho_i = S_i C & \text{Distance from } S_i \text{ to the ICR} \\ \rho_{P_a} = P_a C & \text{Distance from } P_a \text{ to the ICR} \end{cases} \quad (11)$$

Similarly, the differences of interest are defined as:

$$\begin{cases} \rho_{lat,a} = \rho_{P_a} - \rho_{lat} \\ \rho_{i,a} = \rho_i - \rho_{P_a} \end{cases} \quad (12)$$

Figure 4 shows the geometrical interpretation (11) and (12).

The interaction matrices related to (11) and (12) can be easily computed from the classical 3D interaction matrix of a

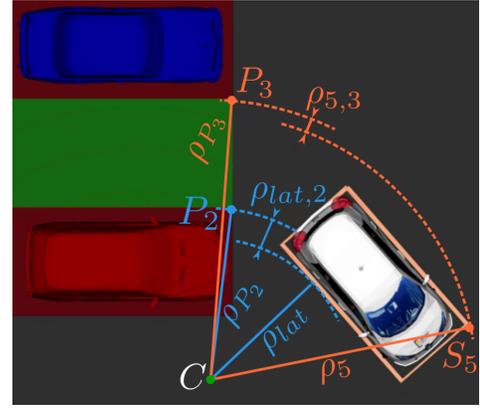


Fig. 4. Differences of radii: $\rho_{lat,2}$ from lateral distance and P_2 (blue) and $\rho_{5,3}$ from S_5 and P_3 (orange). All the radii are defined with regards to the ICR C .

point $({}^iX, {}^iY)$ expressed in \mathcal{F}_i [21]:

$$\begin{cases} \mathbf{L}_{i_x} = \begin{bmatrix} -1 & 0 & {}^iY \end{bmatrix} \\ \mathbf{L}_{i_y} = \begin{bmatrix} 0 & -1 & -{}^iX \end{bmatrix} \end{cases} \quad (13)$$

The set of constrained sensor features used for collision avoidance is denoted as \mathbf{c} . Unlike \mathbf{s} , due to the nature of each type of parking maneuver, the definition of the set of sensor features \mathbf{c} required for collision avoidance is different for each parking case although, in general, constraints on ${}^i h_j$ and X_a are used to avoid going outside the boundaries of the parking spot or into the forbidden areas (Fig. 3), constraints on (12) are used to keep the vehicle's path away from the obstacles and constraints on Y_a are used to keep a certain minimum lateral distance to point P_a .

The exact definition of the set of constrained sensor features \mathbf{c} is now given case by case.

Backward non-parallel case: 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{c} is defined as follows:

$$\mathbf{c} = [\mathbf{c}_3; \mathbf{c}_4; \mathbf{c}_5; \mathbf{c}_6] \quad (\text{dim. } 14) \quad (14)$$

with the ten constrained features perceived by the two rear sensors being:

$$\mathbf{c}_3 = [{}^3h_2; {}^3h_4; {}^3h_5; {}^3X_2; {}^3Y_2; \rho_{lat,2}], \quad (15a)$$

$$\mathbf{c}_6 = [{}^6h_2; {}^6h_3; {}^6h_5; {}^6X_3], \quad (15b)$$

and the four ones perceived by the two front sensors being:

$$\mathbf{c}_4 = [{}^4h_4; {}^4h_5], \quad (16a)$$

$$\mathbf{c}_5 = [{}^5h_3; {}^5h_5]. \quad (16b)$$

Forward non-parallel case: Analogously to the precedent case, given that now the front side of the vehicle should enter first into the parking spot, the majority of the constrained sensor features should be observed by the sensors placed at the front corners of the vehicle, thus \mathbf{c} is defined as follows:

$$\mathbf{c} = [\mathbf{c}_3; \mathbf{c}_4; \mathbf{c}_5; \mathbf{c}_6] \quad (\text{dim. } 13) \quad (17)$$

with the features perceived by the rear sensors being:

$$\mathbf{c}_3 = {}^3h_5, \quad (18a)$$

$$\mathbf{c}_6 = {}^6h_5, \quad (18b)$$

and those perceived by the front sensors being:

$$\mathbf{c}_4 = [{}^4h_2; {}^4h_3; {}^4h_5; {}^4X_3; {}^4Y_3], \quad (19a)$$

$$\mathbf{c}_5 = [{}^5h_2; {}^5h_4; {}^5h_5; {}^5X_2; {}^5Y_2; \rho_{5,2}]. \quad (19b)$$

Backward parallel case: Considering that for this case the vehicle is likely to maneuver while being mostly inside of the parking spot, it should be expected to have more constrained features than for the two previous cases. Additionally, even if the rear side of the vehicle should enter first into the parking spot, the greater risk of collision is on its right side thus the majority of the constrained sensor features should be observed by the sensors placed at the right corners of the car. Moreover, given that this type of parking maneuvers are typically performed on streets rather than in parking lots, the boundary ${}^i\mathcal{L}_3$ would often coincide with a curb. Therefore, \mathbf{c} is defined as follows:

$$\mathbf{c} = [\mathbf{c}_3; \mathbf{c}_4; \mathbf{c}_5; \mathbf{c}_6; \mathbf{c}_7; \mathbf{c}_8] \quad (\text{dim. } 18) \quad (20)$$

with the features perceived by the sensors placed on the right side corners being:

$$\mathbf{c}_3 = [{}^3h_2; {}^3h_5; {}^3X_3; {}^3Y_3; \rho_{lat,3}; {}^3X_4; {}^3Y_4], \quad (21a)$$

$$\mathbf{c}_4 = [{}^4h_4; {}^4h_5; {}^4X_3; {}^4Y_3; \rho_{4,3}], \quad (21b)$$

those perceived from the left side corners being:

$$\mathbf{c}_5 = [{}^5h_4; {}^5h_5], \quad (22a)$$

$$\mathbf{c}_6 = [{}^6h_2; {}^6h_5], \quad (22b)$$

and the ones used to avoid hitting the curb being:

$$\mathbf{c}_7 = {}^7h_3, \quad (23a) \quad \mathbf{c}_8 = {}^8h_3. \quad (23b)$$

Constraints deactivation: It should be noted that some constraints must be deactivated under certain conditions in order to be able to park successfully. As an example, considering a forward parallel parking maneuver, the constrained features 4Y_3 and ${}^3d_{lat3}$ used to keep ip_3 at a certain distance on the right side of the car should only be active as long as 3p_3 is not behind the vehicle, otherwise they would prevent a successful parking maneuver since at the desired pose ip_3 should be on the left side of the car. The constraints deactivation conditions used to obtain the results presented in this work are detailed in Appendix A.

IV. CONTROL

The presented MSBPC approach is based on the Visual Predictive Control (VPC) described in [24] with some modifications to impose an exponential decay of error related to the main task $\mathbf{e}_p = \mathbf{s}_p - \mathbf{s}_p^*$, where the superscript $*$ indicates the desired value. 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).

A. Structure

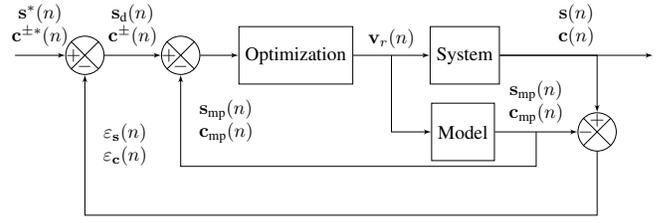


Fig. 5. Control structure [24]

The control structure is based on the internal-model-control (IMC) structure [25] (Fig. 5). This allows compensating for modeling error and disturbances which also helps the optimization (MPC) block to converge faster. The System block contains the robotized vehicle system and sensors whose input is the control variable \mathbf{v}_r and outputs \mathbf{s}, \mathbf{c} are the current values of the sensor features. The reference \mathbf{s}^* is the desired value of the task sensor features \mathbf{s} while \mathbf{c}^\pm are the constraints to be imposed on \mathbf{c} . The error signals $\varepsilon_s, \varepsilon_c$ represent all the modeling errors and disturbances between the current features (\mathbf{s}, \mathbf{c}) and the values that were predicted from the model ($\mathbf{s}_{mp}, \mathbf{c}_{mp}$):

$$\varepsilon_s(n) = \mathbf{s}(n) - \mathbf{s}_{mp}(n) \quad (24a)$$

$$\varepsilon_c(n) = \mathbf{c}(n) - \mathbf{c}_{mp}(n) \quad (24b)$$

where n is the current time.

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

$$\mathbf{s}_d(n) = \mathbf{s}^*(n) - \varepsilon(n) = \mathbf{s}^*(n) - (\mathbf{s}(n) - \mathbf{s}_{mp}(n)), \quad (25)$$

from where it is possible to deduce

$$\mathbf{s}_d(n) - \mathbf{s}_{mp}(n) = \mathbf{s}^*(n) - \mathbf{s}(n), \quad (26)$$

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

To predict the behavior of \mathbf{s}_{mp} and \mathbf{c}_{mp} over a finite prediction horizon N_p , the interaction model described in Sec. III is used. The difference between \mathbf{s}_d and \mathbf{s}_{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 . As with any classical MPC strategy, once a solution to J has been found, the first component $\mathbf{v}_r(n)$ of the optimal control sequence is applied to the vehicle at each iteration.

B. Constraint handling

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

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

$$|v| < v_{\max}, \quad (27a) \quad |\delta| < \delta_{\max}, \quad (27b)$$

where v_{\max} is an adaptive saturation value imposing a deceleration profile based on the velocity profile shown in [5] 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:

$$\begin{cases} |\dot{v}| \leq \dot{v}_{\max} \\ |\ddot{v}| \leq \ddot{v}_{\max} \end{cases} \quad \begin{cases} |\dot{\delta}| \leq \dot{\delta}_{\max} \\ |\ddot{\delta}| \leq \ddot{\delta}_{\max} \\ |\ddot{\delta}| \leq \ddot{\delta}_{\max} \end{cases} \quad (28)$$

The sensor features \mathbf{c} considered for collision avoidance are constrained as follows:

$$\mathbf{c}^- \leq \mathbf{c}_{\text{mp}} \leq \mathbf{c}^+ \quad (29)$$

where the superscripts $-$ and $+$, respectively, indicate the low and high sides of the constraints to be imposed. The numerical values of this constraints can be found in Tables VII – IX that shows most of the features are constrained only on one side.

A constraint domain \mathbb{C} can be defined for $\mathbf{v}_r = [v; \delta]$ (3):

$$\mathbb{C} = \{\mathbf{v}_r, C(\mathbf{v}_r) \leq 0\} \quad (30)$$

where $C(\mathbf{v}_r)$ gathers all inequalities from (28) and (29).

C. Mathematical formulation

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

$$\begin{aligned} \min J(\mathbf{v}_r) \\ \tilde{\mathbf{v}}_r \in \mathbb{C} \end{aligned} \quad (31)$$

with

$$\begin{aligned} J(\mathbf{v}_r) = \sum_{j=n+1}^{n+N_p} \left([\mathbf{s}_d(j) - \mathbf{s}_{\text{mp}}(j)]^T \mathbf{Q}(j) [\mathbf{s}_d(j) - \mathbf{s}_{\text{mp}}(j)] \right. \\ \left. + \mathbf{v}_m(j-1)^T \mathbf{R}(j-1) \mathbf{v}_m(j-1) \right) \end{aligned} \quad (32)$$

and

$$\tilde{\mathbf{v}}_r = \{\mathbf{v}_r(n), \mathbf{v}_r(n+1), \dots, \mathbf{v}_r(n+N_c), \dots, \mathbf{v}_r(n+N_p-1)\} \quad (33)$$

where $(\mathbf{s}_{\text{mp}}(j), \mathbf{c}_{\text{mp}}(j))$ are predicted from the first-order model (5).

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.

Due to the constrained nonlinear optimization, proving stability for non-linear model predictive controllers is rather difficult [26], particularly for a finite N_p . As for sensor-based techniques, only local asymptotic stability can be demonstrated [21] when sensor data is kept in the sensor space. Nonetheless, it has been shown that the local asymptotic stability property is kept for the optimal MSBPC considered [27].

Additionally, it has been shown that a local model based on the interaction matrix (such as the one we consider) is enough to predict the evolution of the sensor features and thus allows performing sensor-based tasks successfully while satisfying the considered constraints [24], [27]. We illustrate the performance of our approach in Sec. V.

1) *Parking strategy*: Considering the definition of ${}^i\mathcal{L}_1$ and ${}^i\mathcal{L}_2$ (Sec. III-B), 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 or front boundary of the vehicle for, respectively, backward and forward maneuvers.

Moreover, since the objective of the auxiliary task is to pull the vehicle away from the parking spot, the desired values ${}^a\mathcal{L}_q^{\text{off}*}$ and ${}^a\mathcal{L}_5^{\text{off}*}$ 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 subsection. Nonetheless, the actual desired values for the offset lines may vary (but remain constant during the parking maneuver) depending on the characteristics of the vehicle (e.g. l_{wb} , δ_{\max}), collision-avoidance constraints (e.g. size of the parking spot, type of maneuver) and weighting strategy, although in general they should be defined such that the vehicle reaches a favorable unparked pose.

Furthermore, using only the features introduced in Sec. III-B, it is possible to reconstruct what the desired parked/unparked poses of the vehicle would be and thus what desired value the sensor features would have. To this goal, one can simply define some auxiliary parked/unparked frames with respect to the parking spot and express the desired lines and points in the aforementioned frames.

The interaction matrix \mathbf{L}_p for the features observed by S_p is computed by a 2nd order approximation [28] of the form:

$$\mathbf{L} = \frac{\mathbf{L}_{\mathcal{L}} + \mathbf{L}_{\mathcal{L}}^*}{2} \quad (34)$$

where $\mathbf{L}_{\mathcal{L}} = [\mathbf{L}_{i,1}; \mathbf{L}_{i,2}]$ and $\mathbf{L}_{\mathcal{L}}^*$ is equal to the value of $\mathbf{L}_{\mathcal{L}}$ at the desired pose. As it has been shown in our previous work [18], [29], the use of this type of interaction matrix (for the parking task) in the control architecture 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 \mathbf{L}_a for the features observed by S_a is computed at each iteration and is defined by (10).

2) *Weighting strategy*: As mentioned in Sec.III-C, \mathbf{s}_p contains the features used to actually drive the car into the parking spot, thus the minimization of the error $\mathbf{e}_p = \mathbf{s}_p - \mathbf{s}_p^*$ is what drives the vehicle towards a parked pose, typically with a backward motion for backward maneuvers and forward ones for the forward case, while $\mathbf{e}_a = \mathbf{s}_a - \mathbf{s}_a^*$ would be normally minimized with opposite directions of motion. This enables automatic multi-maneuver, that is online switching of the moving direction, without any need for planning or replanning. In order to automatically maneuver the vehicle with the appropriate direction of motion, the influence of each sensor feature is regulated by means of the weighting matrix \mathbf{Q} , which remains constant along the prediction horizon. It is

defined as:

$$\mathbf{Q} = \left[\begin{array}{c|c} Q_p \mathbf{W}_p & 0_{6 \times 6} \\ \hline 0_{6 \times 6} & Q_a \mathbf{W}_a \end{array} \right] \quad (35)$$

where:

- Q_p is the global adaptive gain, taking into account the speed of the vehicle with regards to the vicinity of the constraints
- Q_a is the auxiliary adaptive gain, allowing the unparking maneuver
- $\mathbf{W}_p = \text{diag}(w_1, \dots, w_6)$ balances the importance of the 6 parking features s_p
- $\mathbf{W}_a = \text{diag}(w_7, \dots, w_{12})$ balances the importance of the 6 unparking features s_a

The auxiliary gain Q_a activates the unparking maneuver. As such, it is defined as:

$$Q_a = \begin{cases} 0 & \text{if } \|\mathbf{p}\mathbf{1}_1 - \mathbf{p}\mathbf{1}_1^*\| < \epsilon_{\mathcal{L}_1} \text{ and } Q_p > 0 \\ 1 - Q_p & \text{otherwise} \end{cases} \quad (36)$$

where $\epsilon_{\mathcal{L}_1}$ is a small positive scalar value that serves to nullify Q_a (and consequently the influence of s_a) when the vehicle is almost collinear to ${}^p\mathcal{L}_1$ (parking spot main axis). The values of Q_p and $w_{i \in [1,12]}$ are computed from a combination of the same elementary function denoted w in the sequel.

Due to the different nature of each of the considered cases, the exact definition of $(Q_p, w_{i \in [1,12]})$ varies slightly depending on the type of maneuver although their structure remains similar. We now detail the rationale behind the adaptive weighting for the non-parallel backward case. Appendix B exposes the shape of the w function and goes further into numerical implementation for all parking configurations.

Let us consider a boolean variable `fwd` that serves as a hint of whether the vehicle should move forward or not. If the vehicle is almost collinear to ${}^p\mathcal{L}_1$ (i.e. $\|\mathbf{p}\mathbf{1}_1 - \mathbf{p}\mathbf{1}_1^*\| < \epsilon_{\mathcal{L}_1}$) or near the desired parked pose (i.e. $\|\mathbf{e}_p\| < \epsilon_p$), `fwd` is set to false. The vehicle is considered to be at (or in the vicinity of) a local minimum if $\|\mathbf{e}_p\| \geq \epsilon_p$ (i.e. non-null error) while $|v|$ is below a given threshold (i.e. null output velocity). In this case, `fwd` is flipped and the control automatically switches to the opposite maneuver.

The procedure for computing the weight multiplier Q_p at each iteration depends on `fwd` as shown in Algorithm 1. The idea is to drive the vehicle typically with a backward motion if it starts to move while being on the right side of ${}^i\mathcal{L}_1$ and with a forward motion if the vehicle starts on the left side of said line or if a local minimum had been detected when moving in reverse, generally not driving farther away than ${}^i\mathcal{L}_1^{\text{off}}$.

Regarding \mathbf{W}_p , 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_3 and w_6 can be set to constant values while the remaining diagonal elements of \mathbf{W}_p can be defined as:

$$w_i = w(e_6) \forall i \in \{1, 2, 4, 5\}. \quad (37)$$

As for \mathbf{W}_a , its role is to induce small *corrective* motions if the vehicle is relatively close to be collinear to the main

Algorithm 1: Q_p computation at each iteration for backward non-parallel case

Update sensor features \mathbf{s} and \mathbf{c} ;

`fwd` = true if car is moving forward;

if $\|\mathbf{p}\mathbf{1}_1 - \mathbf{p}\mathbf{1}_1^*\| < \epsilon_{\mathcal{L}_1}$ **or** $\|\mathbf{e}_p\| < \epsilon_p$ **then**

`fwd` = false;

if local minimum detected **then**

`fwd` = !`fwd`;

if `fwd` == false **then**

$Q_p = w(e_3)$;

 /* $Q_p \rightarrow 0$ if the car is on the left
of ${}^p\mathcal{L}_1$ */

else

$Q_p = w(e_9) w({}^3h_5, {}^5h_5, {}^6h_5)$;

 /* $Q_p \rightarrow 1$ if the car is far from the
transitable area limits and is
(or is approaching to be) on the
right side of ${}^a\mathcal{L}_1^{\text{off}}$ */

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. Therefore, the diagonal elements of \mathbf{W}_a are given by:

$$w_7 = w_8 = w({}^5h_5, {}^6h_5, \max(w(e_{10}), w(e_{11}))) \quad (38a)$$

$$w({}^m h_1) w({}^1 h_1, \max(w(e_7), w(e_8))),$$

$$w_9 = (w(s_3) + \max(w(e_7), w(e_8))) w(e_9), \quad (38b)$$

$$w_{10} = w_{11} = w(s_9, {}^1 h_1) + w(e_3) + (1 - w({}^5 h_5)), \quad (38c)$$

$$w_{12} = (1 - w(e_9)) w({}^m h_1) + \left(w({}^5 h_5) (1 - w(e_3)) \right. \\ \left. (1 - \max(w(e_7), w(e_8))) w({}^3 h_5) \right) \\ \left. + (1 - w({}^3 h_5, {}^5 h_5, {}^6 h_5)) \max(w(e_7), w(e_8)). \quad (38d)$$

3) *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 and θ are regulated by means of the weighted matrix \mathbf{R} , which remains constant along the prediction horizon. It is defined as:

$$\mathbf{R} = \begin{bmatrix} \lambda_v Q_p & 0 \\ 0 & 1 \end{bmatrix}. \quad (39)$$

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

V. RESULTS

For the results shown in this section, the parameters in Table IV are considered, $\epsilon_{\mathcal{L}_1} = 0.125$ for non-parallel maneuvers while $\epsilon_{\mathcal{L}_1} = 0.035$ for parallel ones. The value of δ_{\max}

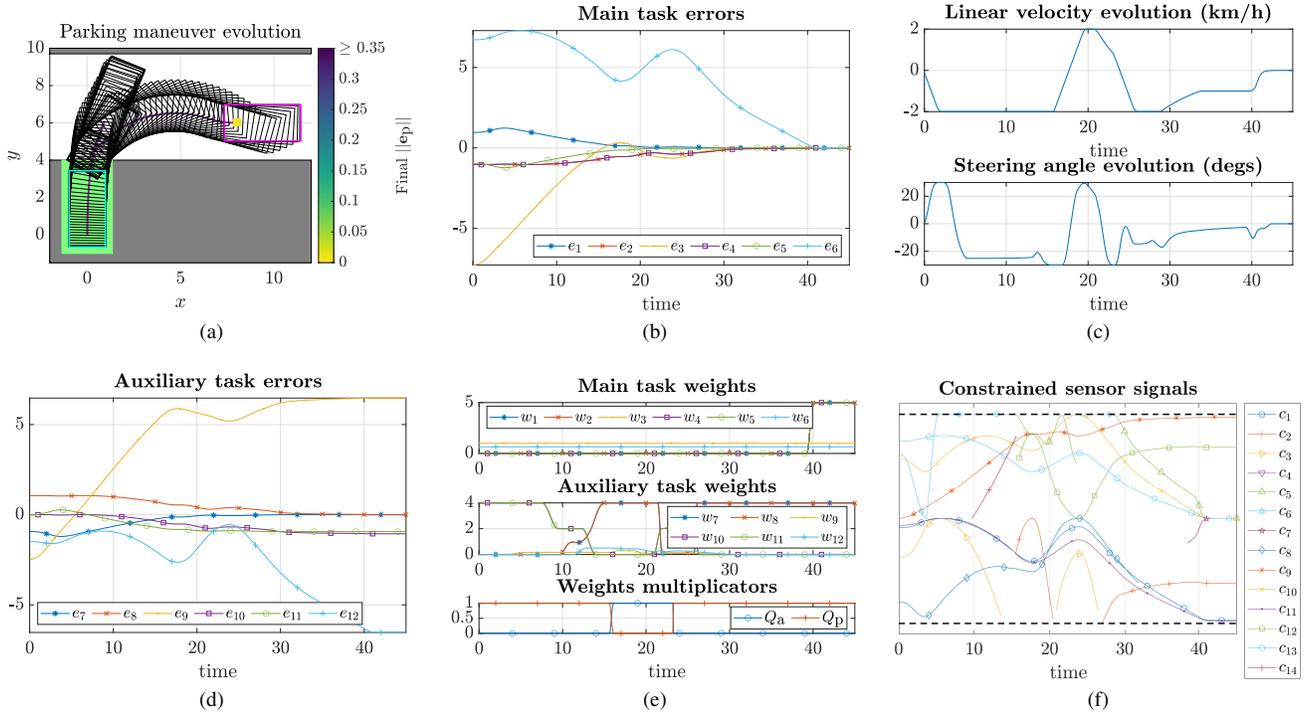


Fig. 6. Constrained \perp backward parking maneuver signals: (a) performed maneuver, (c) control signals, (e) weighting-related signals, (b) main and (d) auxiliary task errors and (d) constrained sensor signals. . Initial pose = (8 m, 6 m, 0°)

corresponds to the maximum steering angle of the real vehicle while the rest of the parameters required to solve (31) were determined by empirical testing, nevertheless some guidelines on how to tune them can be given:

- The maximum longitudinal velocity v_{\max} and the maximum values of the constraints (28) 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 but small enough to be able to meet the computational time requirements. Additionally, it is known that if N_p tends to infinity, the control problem becomes an optimal control and thus closed-loop stability is ensured [26].
- The threshold value $\epsilon_{\mathcal{L}_1}$ used to determine whether or not Q_a 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 a potential increase on the number of maneuvers required to park.

A. Simulation results

The simulations carried out in MATLAB use the *fmincon* solver with a Sequential Quadratic Programming (SQP) algorithm.

TABLE IV
CONTROL-RELATED VEHICLE PARAMETERS

Parameters	Notation	Value
Sampling time	T_s	0.1 s
Control horizon	N_c	10 (1 s)
Prediction horizon	N_p	25 (2.5 s)
Maximum longitudinal velocity	v_{\max}	≤ 0.556 m/s
Maximum longitudinal acceleration	\dot{v}_{\max}	0.3 m/s ²
Maximum longitudinal jerk	\ddot{v}_{\max}	0.5 m/s ³
Maximum steering angle	δ_{\max}	0.5236 rad
Maximum δ velocity	$\dot{\delta}_{\max}$	0.6981 rad/s
Maximum δ acceleration	$\ddot{\delta}_{\max}$	0.9 rad/s ²
Maximum δ jerk	$\ddot{\delta}_{\max}$	0.9 rad/s ³
Threshold value for ϵ_p	ϵ_p	0.1
Minimum velocity threshold	v_{th}	0.06 m/s
Iterations threshold	\dot{t}	5
Longitudinal velocity gain	λ_v	0.1

1) *Individual cases - MATLAB simulations:* To illustrate the performance of the proposed approach, a few examples of each type of parking situation are shown below. The initial position is marked with a colored (yellow and light green in the shown cases) square whose color depends on the final value of $\|\mathbf{e}_p\|$ as the associated colorbar indicates. The initial and final bounding rectangles of the vehicle are colored in magenta and cyan respectively.

In addition to the parking maneuver evolution, the evolution of the different control-related signals can be seen in Fig. 6. Thanks to the constraints related to the controls signals, they evolve smoothly in general (Fig. 6c). From the constraints

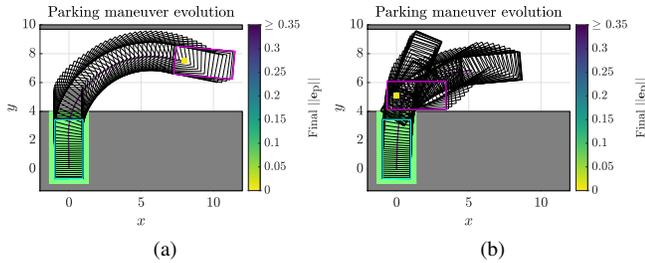


Fig. 7. Constrained \perp backward parking maneuvers. (a) Initial pose = (8 m, 7.5 m, -5°), (b) Initial pose = (0 m, 5.1 m, 0°)

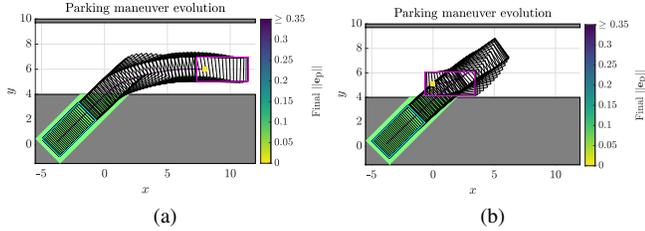


Fig. 8. Constrained diagonal backward parking maneuvers. (a) Initial pose = (8 m, 6 m, 0°), (b) Initial pose = (0 m, 5.1 m, 0°)

values defined in VII, it can be seen that the (active) constraints imposed on \mathbf{c} are satisfied at each time instant (Fig. 6f) ensuring a collision-free maneuver. It can be seen how the main task error ($e_p = s_p - s_p^*$) (Fig. 6b) is minimized as the vehicle moves backwards while $e_a (= s_a - s_a^*)$ (Fig. 6d) is minimized with motions in the opposite direction. Additionally, one can notice that when Q_p is larger than Q_a (Fig. 6e), the vehicle is moving backwards and when Q_a is larger the opposite occurs. 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 influence of s_9 pulls the vehicle out of the parking spot while the weights related to s_7 and s_8 try to keep the vehicle parallel to main axis of the parking spot.

Two more backward perpendicular cases can be seen in Fig. 7. Fig. 7a shows a case where parking with a single maneuver is possible while Fig. 7b presents a rather challenging initial pose from where the vehicle has to perform four maneuvers in order to park. For the three backward perpendicular cases shown, the final value of $\|e_p\|$ is rather small (< 0.0082).

The initial poses of the cases presented in Figs. 8a and 8b are, respectively, the same as those in Figs. 6a and 7b. It can be seen how, as expected, when considering diagonal parking spots (Fig. 8) instead of perpendicular ones the complexity of the maneuvers is reduced. The case shown in Fig. 8a still requires only one maneuver but now there are much larger portions of the maneuver of mostly straight motions. As for the case presented in Fig. 8b, the number of maneuvers was reduced to only two compared to four for the perpendicular case.

As most experienced drivers know, parking maneuvers with

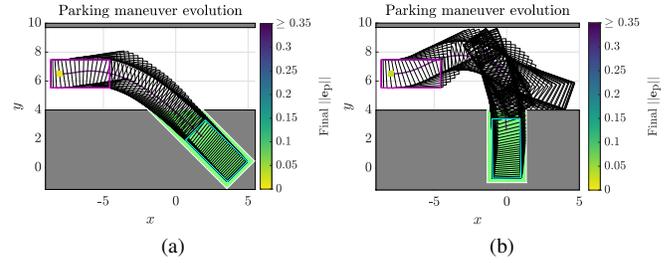


Fig. 9. Constrained non-parallel forward parking maneuvers: (a) diagonal and (b) perpendicular. Initial pose = (-8 m, 6.5 m, 0°)

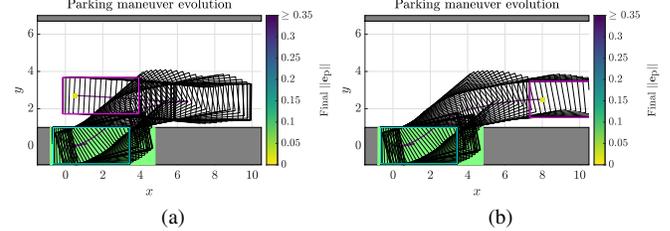


Fig. 10. Constrained \parallel backward parking maneuvers. (a) Initial pose = (0.5 m, 2.7 m, 0°), (b) Initial pose = (8 m, 2.5 m, 0°)

forward motions are usually more challenging than their backward motions counterparts thus a potential increase on the number of required maneuvers should not be surprising. The two non-parallel forward cases shown in Fig. 9 share the same initial pose. On the one hand, for the diagonal case (Fig. 9a) parking with a single maneuver is possible with a final $\|e_1\| = 0.0125$. On the other hand, for the perpendicular case shown in Fig. 9b the vehicle has to perform five maneuvers (four changes of direction), achieving a final $\|e_1\| = 0.0345$. Even if the final error of the forward perpendicular case is considerably larger than the previous cases (around three times larger), it remains at a reasonable value, especially for such a challenging case.

Finally two backward parallel cases are shown in Fig. 10. In Fig. 10a, the vehicle starts the parking maneuver by moving forward in order to preposition itself (similarly to Fig. 7b) and then performs three more motions leading to a total of four maneuvers to park with a final $\|e_2\| = 0.0182$. For Fig. 10b no prepositioning is required and thus the car is able to park with only three maneuvers and a final $\|e_2\| = 0.0197$.

2) *Comparison against a state of the art path planning approach:* To put in perspective the performance of the proposed MSBPC approach, it is compared against a state of the art path planning approach [8] using a sampling-based motion planner and a Hybrid Curvature (HC) steer function. To implement said approach, the *steering_functions* package published by the authors was interfaced with the Open Motion Planning Library (OMPL) [30] using its RRT* implementation for the planner. To account for the different planner (BiRRT* in [8] and RRT* in our implementation) the maximum planning time was set to 20 s instead of 6 s as presented in [8]. It was chosen for the HC paths to have zero curvature at the beginning and at the end (denoted as HC⁰⁰). The path planning results shown below were obtained on an AMD Ryzen 7 2700X at 4.15 GHz.

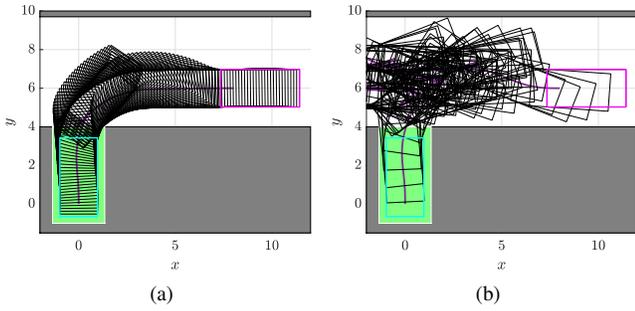


Fig. 11. RRT* path planning results using HC^{00} steering function. Initial pose = (8m, 6m, 0°)

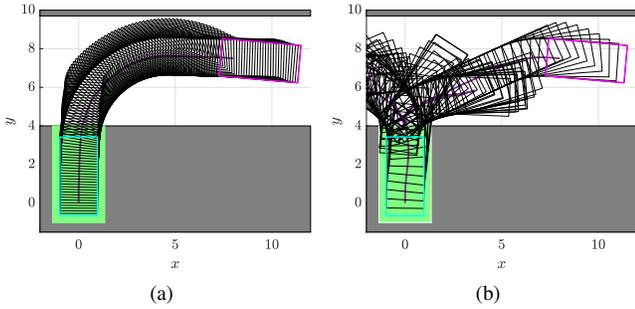


Fig. 12. RRT* path planning results using HC^{00} steering function. Initial pose = (8m, 7.5m, -5°)

The initial poses of the cases presented in Figs. 11, 13 and 12 are, respectively, the same as in Figs. 6a, 7a and 7b. All of them were tested in analogous simulated environments. Two different outcomes for each initial pose of the planned cases are shown: a desirable one (Figs. 11a, 12a and 13a) and an undesirable one (Figs. 11b, 12b and 13b). As one could expect, due to the planner’s randomness, the planned paths can be completely different each time the planning is performed. For this reason, achieving repeatable parking maneuvers with RRT-based planners is not straightforward, especially for short planning times.

As shown in Fig. 11 the planned path could be rather good (Fig. 11a), leading to a visibly shorter path length than our MSBPC approach (Fig. 6a) but could also be completely undesirable (Fig. 11b). When considering the case where parking in a single maneuver is possible for our approach (Fig. 7a), one can notice that the desirable planned path (Fig. 12) also performs only one maneuver. In this case it is harder to tell which one is better just by visual inspection. Finally, for the most challenging initial pose (of the presented ones) for our approach (Fig. 7b), the desirable planned path (Fig. 13) shows less maneuvers (4 vs 3) by starting with a backwards motion instead of a forward one.

One can see that path planning could yield shorter paths than our approach with the drawback of these good outcomes not being straightforwardly repeatable. Additionally, these comparisons do not account for sensor noise, model-plant mismatches, localization and path-tracking errors, etc. Furthermore, since our MSBPC approach does not plan any path and instead the controller runs online, there is (virtually)

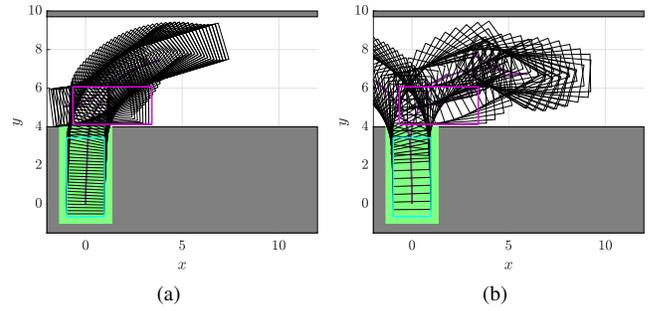


Fig. 13. RRT* path planning results using HC^{00} -RS steering function. Initial pose = (0m, 5.1m, 0°)

no inherent delay to be able to start a parking maneuver.

3) *Exhaustive simulations:* To assess the convergence of the proposed scheme, several exhaustive simulations were conducted for the different parking cases. Due to paper length constraints, for all the shown cases (Fig. 14), the initial orientation of the vehicle is equal to zero ($\theta_{T=0} = 0^\circ$). Cases with different initial orientations (ranging from -30° to 30° with a step of 5°) can be found in the attached video ¹.

Since the exhaustive simulations are an aggregation of the results obtained from several simulations (like those shown in Figs. 6a,7a-10b), each figure consists of a parking spot (represented by a green rectangle) adapted to each case, forbidden zones represented in gray and a scatter plot of the initial position of the vehicle, whose color depends (like in Sec. V-A1) on the final value of $\|e_p\|$. As denoted by the colorbar, yellow/light green values of each scatter plot are inside the region of attraction (ROA), the green/blue values can be interpreted as the boundaries of the ROA and the dark blue ones represent 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 maneuvers, the car is able to park from virtually any initial position in the analysis window with no points out of the ROA.

As expected, the most challenging type of parking maneuver is the forward perpendicular case, often ending the maneuver with a final main task error $0.03 < \|e_p\| < 0.05$ as opposed to $\|e_p\| < 0.015$ for the other non-parallel maneuvers. Backward parallel maneuvers are the second most challenging case often yielding larger final $\|e_p\|$ values than the less challenging diagonal (any direction) and backward perpendicular cases but not to the same extent as the forward perpendicular case. It should be noted that the lateral (associated to s_3) error is responsible for the larger (but still rather satisfactory) final $\|e_p\|$ values. Such behavior is reasonable considering that the lateral motion of the vehicle can’t be controlled directly. Nonetheless, the final lateral error may be improved by increasing its associated weight at the expense of a potential increase on the number of maneuvers required for the parking task.

¹<https://youtu.be/UO3yGzJnWrY>

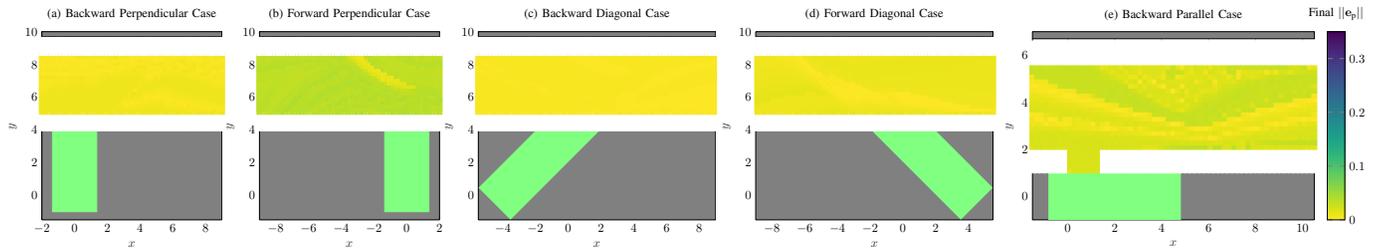


Fig. 14. Exhaustive simulations for different types of parking maneuvers with a sampling step of 20cm for the initial position. For non-parallel cases: spot length = 5m and width = 2.7m. For (e): spot length = 5.6m and width = 2m.

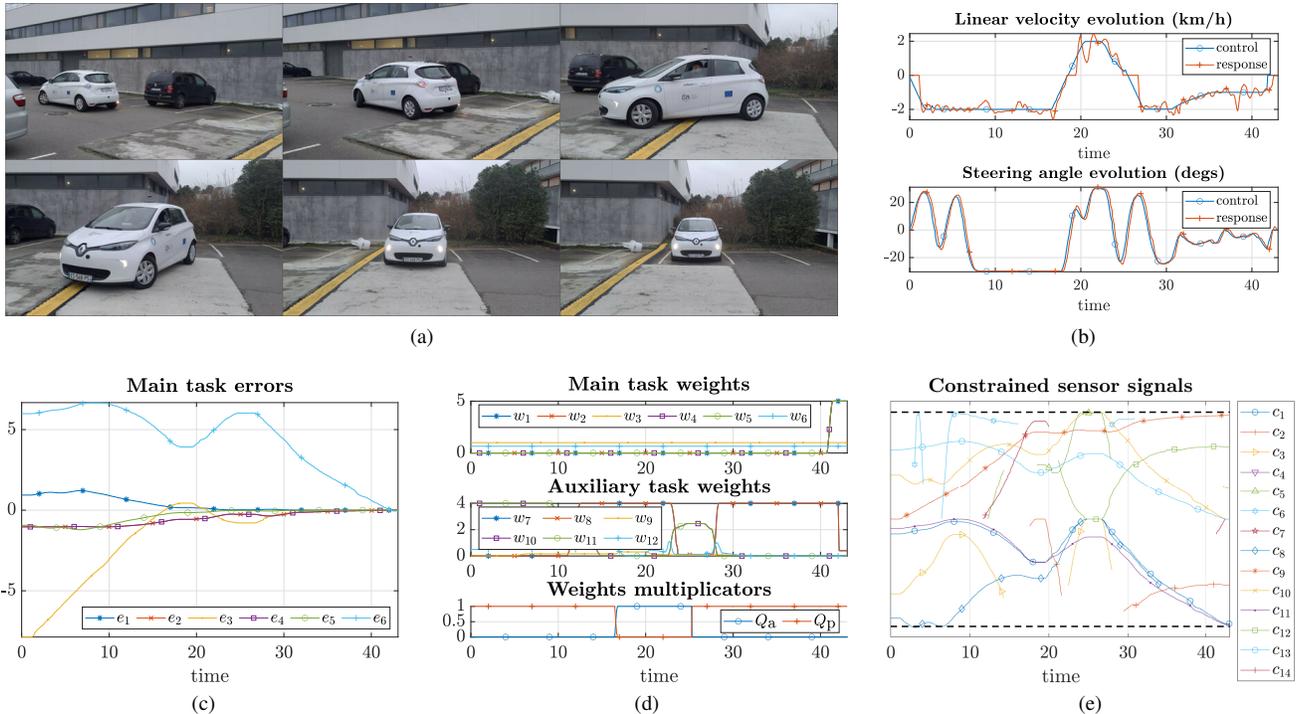


Fig. 15. (a) Experimental car parking in a perpendicular spot (video of the performed parking maneuver is attached to this paper). (b-e) Constrained real backward \perp parking maneuver signals

B. Real experimentation

Real experimentation was conducted for the backward perpendicular parking case (Fig. 15a) with the controller running at 10 Hz. In this case the controller is implemented in C++ using the solver NLOpt [31] with a Sequential Least Squares Programming (SLSQP) algorithm [32]. The whole software architecture (and not only the parking controller) runs on a dual core Intel Core i5-3610ME.

To address the robustness concerns that might arise from the considered planar world assumption, it was chosen to make the vehicle pass over a speed bump during the parking maneuver (Fig. 15a). The estimated minimum and maximum values of roll and pitch (in degrees) were, respectively, $[-2.57, 0.61]$ and $[-1.36, 3.51]$.

It can be clearly seen that, in spite of the speed bump (which in addition to perturb the 3D orientation of the vehicle, disturbs as well the response of the linear velocity), the vehicle manages to park successfully (Figs. 15a, 15c) while satisfying the constraints during the whole maneuver (Fig. 15e) thus

validating the results obtained from the corresponding convergence analysis. The final $\|e_2\|$ achieved was 0.0499 which if reconstructed translates to errors of approximately 0.11 cm laterally, -4.97 cm longitudinally and 0.18° in orientation.

The linear velocity setpoint and response can be seen in Fig. 15b. One can notice that the setpoint is not perfectly followed by the low-level controller, especially when the car passes over a speed bump. This behavior occurs because the longitudinal velocity parameters had been identified for higher velocities (up to 130 km/h), showing an unperfect behavior for very low velocities (such as our application). Even if this behavior is not related to the presented technique, it serves to show the robustness of our approach against model-plant mismatches.

VI. CONCLUSIONS

As an extension of our previous work [29], here we have formalized under a single common framework five different parking cases: parking into perpendicular and diagonal parking spots with both forward and backward motions and into parallel ones with backward motions.

The task definition is essentially the same for every case: 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 either the rear or front boundary of the parking spot; having just to consider slight adaptations on the interaction model for the different cases.

Furthermore, 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 when compared to exploration-based path planning techniques.

Thanks to the convergence analyses performed, it has been shown that the presented MSBPC approach is able to successfully park the vehicle from virtually any initial pose, normally achieving rather small final $\|e_p\|$ values.

The results obtained from real experimentation validate the conducted convergence analysis as well as the robustness and effectiveness of the presented approach since, even in the presence of a speed bump that considerably disturbs the 3D orientation of the vehicle and its longitudinal velocity, the car parked successfully with a satisfactory final $\|e_2\| = 0.0499$ while satisfying the constraints during the whole maneuver.

Future work will focus on simplifying the weighting strategy and adapting it with online learning. Moving obstacle avoidance, that is theoretically addressed in this work, will also be investigated with real experiments.

APPENDIX A

CONDITIONS AND NUMERICAL VALUES FOR CONSTRAINTS DEACTIVATION

The constraints deactivation conditions used to obtain the results presented in this work are now detailed. Table V lists the car and environment-dependant parameters while Table VI details the tuning parameters with their values and semantics.

TABLE V
CAR AND ENVIRONMENT SYMBOLS

Symbol	Description
w_{rd}	Width of the transitable area
w_{sp}	Width of the parking spot
ρ_i^-	Distance S_iC for minimum steering angle $-\delta_{max}$
ρ_i^+	Distance S_iC for maximum steering angle δ_{max}

TABLE VI
TUNING SYMBOLS AND VALUES

Symbol	Semantic	Value
d_1	lateral margin, entry side	5 cm
d_2	lateral margin, opposite side	7.5 cm
d_3	final lateral margin	10 cm
d_4	front / rear margins	15 cm

TABLE VII
CONSTRAINTS DEACTIVATION - BACKWARD NON-PARALLEL CASE

Constraint	Deactivate if
$\rho_{lat,2} \leq -d_2$	$\delta \geq 0$ or $v \geq -v_{th}$ or $(v < 0$ and ${}^3X_2 > x_3 $) or ${}^3h_5 > \frac{3}{4}\rho_m^+$ or $({}^5h_4 < \frac{3}{4}\rho_m^+$ and ${}^3h_5 > \frac{3}{4}\rho_m^+)$
${}^3h_2, {}^6h_2 \geq d_4$	-
${}^3h_4 \geq d_3$	${}^3Y_2 \geq 0$ or ${}^6Y_3 \leq 0$ or ${}^3X_2 < 0$
${}^3h_5 \geq d_1$	${}^3h_4 > d_1$ and ${}^3h_3 < -d_1$
${}^3X_2 \geq l_{ve} + d_1$	${}^4h_5 > d_1$ or ${}^4h_4 < 0$ or ${}^3h_5 < 0$ or ${}^3Y_2 < -d_1$ or ${}^6Y_2 > d_1$
${}^3X_2 \leq -d_1$	${}^3X_2 < -2v_{max}$ or ${}^3Y_2 < -d_1$ or ${}^3h_3 > 0$ or $({}^3h_5 \leq d_1$ and $v \leq 0)$
${}^3Y_2 \leq -d_2$	${}^3h_3 > 0$ or $(v \leq 0$ and $({}^3h_5 \leq d_1$ or $\rho_{lat,2} \geq -d_2))$ or ${}^3Y_2 > 0$ or ${}^6h_4 < 0$
${}^4h_4 \geq d_1$	${}^4h_5 > d_1$ or ${}^4h_4 < 0$
${}^4h_5 \geq d_1$	${}^4h_4 > d_1$ and ${}^4h_3 < -d_1$ and ${}^3h_5 < {}^4h_5$
${}^5h_3 \leq -d_1$	${}^3h_3 > 0$ and ${}^3h_5 > d_1$
${}^5h_5, {}^6h_5 \leq w_{rd}$	-
${}^6h_3 \leq -d_1$	${}^3Y_3 < -d_3$ or $({}^6X_3 < 0$ and ${}^6Y_3 < d_3)$ or $({}^6X_3 > 0$ and ${}^3Y_3 < 0)$
${}^6X_3 \leq -d_1$	${}^3Y_3 < -d_1$ or ${}^6Y_3 > d_1$ or $({}^3h_3 > 0$ and ${}^3Y_3 < -d_1)$

TABLE VIII
CONSTRAINTS DEACTIVATION - FORWARD NON-PARALLEL CASE

Constraint	Deactivate if
${}^3h_5 \geq d_1$	${}^3h_4 > 0$ and ${}^3h_3 < 0$ and ${}^3h_5 > {}^4h_5$
${}^4h_2, {}^5h_2 \leq d_4$	-
${}^4h_5 \leq d_1$	${}^4h_4 > d_1$ and ${}^4h_3 < -d_1$
${}^4h_3 \geq -d_2$	${}^4X_3 > 2v_{max}$ or ${}^4Y_2 {}^4Y_3 > 0$ or ${}^4h_5 > 0$
${}^4X_3 \leq d_2$	${}^4Y_3 < 0$ or ${}^5Y_3 > d_1$ or ${}^4X_3 > 2v_{max}$ or ${}^3X_3 < -d_1$
${}^4Y_3 \geq -d_1$	${}^4X_3 > 2v_{max}$ or ${}^3h_4 > d_1$ or ${}^3h_5 < {}^4h_5$
$\rho_{5,2} \leq d_3$	$v \leq 0$ or $\delta \geq 0$ or ${}^4Y_3 \geq -d_3$ or $\rho_{5,2}^- \leq d_3$ or ${}^5h_5 \leq 0$ or ${}^5h_1 \geq 0$ or ${}^5h_5 \geq \frac{1}{4}\rho_5^- - \frac{3}{3}v_{max}$ or ${}^m h_3 \geq 0$
${}^5h_4 \leq d_1$	$\rho_{5,2} \geq d_3$ or ${}^5h_5 > 0$ or ${}^5X_2 > 0$
${}^5h_5, {}^6h_5 \leq w_{rd}$	-
${}^5X_2 \leq d_2$	${}^5Y_2 > d_1$ or ${}^4Y_2 < -d_1$ or ${}^5X_2 > 2v_{max}$ or ${}^6X_2 < -d_1$ or $({}^3h_3 < 0$ and ${}^3h_5 < d_1$ and ${}^3h_5 < {}^4h_5)$
${}^5Y_2 \geq -w_{ve} - d_1$	${}^4h_4 > 0$ or ${}^5X_2 > 2v_{max}$ or ${}^6X_2 < -d_1$

APPENDIX B

WEIGHTING FUNCTIONS NUMERICAL IMPLEMENTATION

In this appendix we detail the implementation of the weights. They are all computed from a sigmoid-like weighting function $w(s)$, illustrated in Fig. 16.

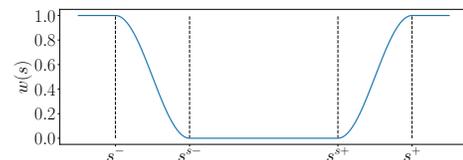


Fig. 16. Generic weighting function $w(s)$. This function smoothly maps a feature s to $[0, 1]$ according to lower and upper thresholds.

TABLE IX
CONSTRAINTS DEACTIVATION - BACKWARD PARALLEL PARKING

Constraint	Deactivate if
$\rho_{lat,3} \leq -d_2$	$\delta \geq 0$ or ${}^3Y_3 > -d_2$ or $(v \leq 0$ and ${}^3X_3 > x_3 $) or $(v \geq 0$ and $({}^3X_3 < x_3 $ or ${}^3h_5 > d_2))$
${}^3h_2 \geq d_3$	${}^3h_5 > d_2$
${}^3h_5 \geq d_2$	${}^3h_2 > d_1$ and ${}^3h_4 < -d_1$
${}^3X_3 \leq -d_3$	${}^3Y_3 < -d_1$ or ${}^6Y_3 > d_1$ or ${}^3X_3 < -2v_{max}$ or ${}^4X_3 > 0$ or ${}^4h_4 < 0$ or ${}^3h_5 \leq d_2$
${}^3X_4 \leq -d_2$	$v \geq 0$ or ${}^3Y_4 < -d_1$ or ${}^6Y_4 > d_1$
${}^3Y_3 \leq -d_2$	${}^3Y_3 > 0$ or $({}^3X_3 > x_3 $ and $v < 0$) or ${}^4X_3 > 0$ or ${}^4X_3 \leq d_3$ or $(\delta > 0$ and $({}^3h_5 > 0$ and ${}^3h_4 < 0))$
${}^3Y_4 \leq -d_2$	${}^4h_5 > d_1$ or ${}^3h_5 < {}^4h_5$ or $({}^3h_5 < d_2$ and ${}^4h_5 < d_2)$
${}^4d_3 \leq -d_4$	$\delta \leq 0$ or $v > 0$ or ${}^4X_3 > d_3$ or ${}^3h_5 > d_2$ or ${}^3h_4 > - x_3 $ or ${}^4h_5 < 0$
${}^4h_5 \geq d_2$	${}^4h_4 < 0$ and ${}^4h_2 > 0$ and $(({}^3h_5 < {}^4h_5$ and ${}^4h_5 > 0)$ or ${}^4h_5 < 0)$
${}^4h_4 \leq -d_3$	${}^4h_5 > d_2$
${}^4X_3 \geq d_3$	${}^3X_3 \geq -d_3$ or ${}^4Y_3 < -d_1$ or ${}^5Y_3 > d_1$ or ${}^4X_3 > 2v_{max}$ or ${}^6h_5 < 0$ or ${}^3h_4 > 0$ or ${}^3X_3 < 0$
${}^4Y_3 \leq -d_2$	${}^4Y_3 > 0$ or ${}^4h_4 < 0$ or $ \delta > 0$
${}^5h_4 \leq -d_3$	$v < 0$ or ${}^5h_5 > 0$
${}^5h_5, {}^6h_5 \leq w_{rd}$	-
${}^6h_2 \geq d_3$	${}^3h_5 > d_2$ or ${}^6h_5 > d_2$
${}^7h_3, {}^8h_3 \geq d_1$	-

The weighting function is usually even (i.e. $s^- = -s^+$ and $s^{s^-} = -s^{s^+}$) and is often used with only one (upper or lower) threshold. In the sequel we thus denote this function as:

$$\begin{cases} w(s, s^{s^+}, s^+) & \text{for even thresholds} \\ w(s, s^{s^+}, s^+, s^{s^-}, s^-) & \text{for non-even thresholds} \\ w^+(s, s^{s^+}, s^+) & \text{for upper thresholds only} \\ w^-(s, s^{s^-}, s^-) & \text{for lower thresholds only} \end{cases}$$

As exposed in Section III-C, the task features are Plücker coordinates and as such can be:

- Distance-based ($e_i, i \in \{3, 6, 9, 12\}$, $\cdot h$), in this case the thresholds are given in meters. Some of the thresholds are the same as the ones introduced in Table VI.
- Orientation-based ($e_i, i \notin \{3, 6, 9, 12\}$), in this case the feature corresponds to the sine/cosine of the actual angle and the thresholds are given in this metric.

The numerical values that we use are listed in Table X.

TABLE X
WEIGHTING THRESHOLDS

Symbol	Semantic	Value
d_2	“close enough” for unparking	7.5 cm
d_4	“far enough” for parking	15 cm
d_5	“close enough” for parking	2.5 cm
d_6	“far enough” for unparking	25 cm
a_1	“aligned enough” for parking	$\sin 2.8^\circ$
a_2	“unaligned enough” for parking	$\sin 42^\circ$

Furthermore, the two weights associated to the orientation

of a given line are always equal:

$$w_8 = w_7, w_{11} = w_{10} \quad (40)$$

In the detailed weights we denote the offset from ${}^i\mathcal{L}_j$ to ${}^i\mathcal{L}_j^{\text{off}}$ as ${}^i o_j$. The distance $h_{rd} = \max({}^5h_5, {}^6h_5) - w_{rd}$, that expresses the distance to the other side of the road, is also introduced. We now proceed to give the numerical implementation of the considered weights for the different parking types.

A. Backward non-parallel case

During the parking phase (i.e. `fwd = False`) then Q_p only depends on $e_3 = {}^2h_1$, that is the centering of the vehicle with regards to the parking spot axis:

$$Q_p = w^-(e_3, d_4, d_5) \quad (41)$$

Otherwise, it is defined as:

$$Q_p = w^-(h_{\max}, -\frac{3}{4}v_{\max}, -\frac{3}{2}v_{\max})w^-(e_9, \frac{3}{2}v_{\max}, \frac{3}{4}v_{\max})$$

where $h_{\max} = \max(-{}^3h_5, {}^5h_5 - w_{rd}, {}^6h_5 - w_{rd})$ expresses the suitable distance to take into account while unparking.

The weights related to the parking position error are defined as:

$$w_i = 5w^-(e_6, \frac{3}{4}v_{\max}, \frac{3}{8}v_{\max}) \forall i \in \{1, 2, 4, 5\} \quad (42)$$

The weight related to the parking orientation error associated to ${}^1\mathcal{L}_1$:

$$w_{u_1} = w(\max(|e_7|, |e_8|), a_1, a_2) \quad (43)$$

the remaining weights are defined as follows:

$$\begin{aligned} w_7 &= (0.1 + w^+({}^1h_1, -v_{\max}w_{u_1}, v_{\max}w_{u_1})) \\ &\times (0.1 + 4w^-(h_{rd}, -2v_{\max}w_{u_1}, -3v_{\max}w_{u_1})) \\ &\times w^+({}^m h_1, \frac{3}{4}\rho_m^-, \frac{1}{2}\rho_m^-) \end{aligned} \quad (44)$$

$$\begin{aligned} w_9 &= 0.15(0.1 + w^+(s_3, d_2, d_6) + w_{u_1}) \\ &\times (0.05 + w^+(e_9, -2v_{\max}, 0)) \end{aligned} \quad (45)$$

$$\begin{aligned} w_{10} &= 2(1 - w^-({}^5h_5 - w_{rd}, -\frac{3}{4}v_{\max}, -\frac{3}{2}v_{\max})) \\ &+ 2w(-e_3, \frac{1}{4}w_{sp} - d_4, \frac{1}{2}w_{sp} - d_4, -v_{\max}s_\psi, -2v_{\max}s_\psi) \\ &+ 2w(s_9, {}^2o_1 + v_{\max}s_\psi, {}^2o_1 + 2v_{\max}s_\psi, \frac{1}{2}\rho_m^+, \frac{3}{4}\rho_m^+) \end{aligned} \quad (46)$$

where $s_\psi = \sin \psi$ indicates the angle of the parking spot with regards to the road (Fig. 2).

$$\begin{aligned} w_{12} &= (1 - 0.9w^+(e_9, -v_{\max}, 0)) \\ &\times w^-(-{}^m h_1, -v_{\max}, -2v_{\max}) \end{aligned} \quad (47)$$

$$\begin{aligned} &+ \left(2w^-({}^5h_5, w_{rd} - \frac{3}{4}v_{\max}, w_{rd} - \frac{3}{2}v_{\max}) \right. \\ &\times (1 - w(-e_3, \frac{3}{8}w_{sp}, \frac{1}{2}w_{sp}, d_4, d_1)) \\ &\times (1 - w_{u_1}) w^+({}^3h_5, \frac{3}{4}v_{\max}, \frac{3}{2}v_{\max}) \\ &\left. + \frac{1}{2}w_{u_1}(1 - w^-(h_{\max}, -\frac{3}{4}v_{\max}, -\frac{3}{2}v_{\max})) \right) \end{aligned} \quad (48)$$

B. Forward non-parallel case

$$Q_p = \begin{cases} w^+(2h_1, -d_4, -d_5) & \text{if } \text{fwd} \\ w^+(e_9, -\frac{3}{2}v_{\max}, -\frac{3}{4}v_{\max}) & \text{otherwise.} \end{cases} \quad (49)$$

Weights related to the position error are defined as:

$$w_i = 0.05 + w^-(e_6, \frac{3}{4}v_{\max}, \frac{3}{8}v_{\max}), \forall i \in \{1, 2, 4, 5\} \quad (50)$$

Considering the weights related to the orientation errors associated to ${}^2\mathcal{L}_1$ and ${}^2\mathcal{L}_2$ respectively:

$$\begin{aligned} w_{u_1} &= w^+(\max(e_7, e_8), 0, a_2) \\ w_{u_2} &= w(\max(|e_{10}|, |e_{11}|), 1 - a_2, 1) \end{aligned} \quad (51)$$

We denote $\bar{w}_{u_2} = 1 - w_{u_2}$ the complementary value to w_{u_2} . It is used to scale the maximum velocity according to how far to alignment the vehicle is. The remaining weights are defined as follows:

$$\begin{aligned} w_7 &= (0.1 + w^+({}^m h_1, -2v_{\max}w_{u_1}, -v_{\max}w_{u_1}) \\ &\times w^+(-{}^m h_1, \frac{3}{4}\rho_m^-, \frac{1}{2}\rho_m^-) \\ &\times (0.1 + 4w^-(h_{rd}, -2v_{\max}\bar{w}_{u_2}, -3v_{\max}\bar{w}_{u_2})) \end{aligned} \quad (52)$$

$$\begin{aligned} w_9 &= (0.1 + w^-(2h_1, -d_2, -d_6) + w_{u_1}) \\ &\times (0.05 + w^+(s_3, -2v_{\max}, -v_{\max})) \end{aligned} \quad (53)$$

$$\begin{aligned} w_{10} &= 2(1 - w^-(6h_5 - w_{rd}, -\frac{3}{4}v_{\max}, -\frac{3}{2}v_{\max})) \\ &+ 2w(e_3, \frac{1}{4}w_{sp} - d_4, \frac{1}{2}w_{sp} - d_4, -v_{\max}s_\psi, -2v_{\max}s_\psi) \\ &+ 2w(s_9, \frac{1}{2}\rho_m^-, \frac{1}{4}\rho_m^-, {}^1o_1 - v_{\max}s_\psi, {}^2o_1 + 2v_{\max}s_\psi) \end{aligned} \quad (54)$$

$$\begin{aligned} w_{12} &= (1 - 0.9w^+(e_9, -v_{\max}, 0)) \\ &+ w^-(\max(h_1, -v_{\max}, -2v_{\max})) \\ &+ \left(w^-(6h_5, w_{rd} - \frac{3}{4}v_{\max}, w_{rd} - \frac{3}{2}v_{\max}) \right. \\ &\times (1 - w(\frac{1}{2}({}^4h_1 + {}^5h_1), \frac{3}{8}w_{sp}, \frac{1}{2}w_{sp}, d_4, d_1)) \\ &\times (1 - w_{u_1}) w^+({}^4h_5, \frac{3}{4}v_{\max}, \frac{3}{2}v_{\max})) \\ &+ \frac{1}{2}w_{u_1}(1 - w^-(h_{\max}, -\frac{3}{4}v_{\max}, -\frac{3}{2}v_{\max})) \end{aligned} \quad (55)$$

C. Backward parallel case

When $\text{fwd} = \text{False}$ then Q_p only depends on $e_6 = 2h_2$, that is the distance to the rear bound of the parking spot:

$$Q_p = w^+(e_6, -d_4, -d_2) \quad (56)$$

Otherwise it is defined as:

$$Q_p = w^+(e_{12}, -\frac{3}{2}v_{\max}, -\frac{3}{4}v_{\max})w^-(h_{\max}, -\frac{3}{4}v_{\max}, -\frac{3}{2}v_{\max})$$

Weights related to the position error are defined as:

$$\begin{aligned} w_i &= 0.1 + 4.9w^-(e_6, \frac{3}{4}v_{\max}, \frac{3}{8}v_{\max})w^-(e_3, 0.2, 0.1) \\ &+ 5w^+({}^2h_4, x_3 - v_{\max}, x_3) \\ &+ 5w^+(({}^2\mathbf{u}_1 \times {}^2\mathbf{u}_1^*)(3), 0.75, 1), \forall i \in \{1, 2, 4, 5\} \end{aligned} \quad (57)$$

The weight related to the orientation error are:

$$w_{u_1} = w(\max(|e_7|, |e_8|), 0, a_2) \quad (58)$$

the remaining weights are defined as follows:

$$\begin{aligned} w_7 &= 4w^-(e_9, \frac{1}{2}\rho_m^+, \frac{1}{4}\rho_m^+) \\ &+ (2 - w^+(s_3, \frac{1}{4}w_{sp} - d_4, \frac{1}{2}w_{sp} - d_4)) \\ &+ (1 - w^-(\max({}^3h_5, {}^6h_5), -\frac{3}{4}v_{\max}, -\frac{3}{2}v_{\max})) \end{aligned} \quad (59)$$

$$\begin{aligned} w_9 &= (1 - 0.9w^+(e_{12}, -0.5, 0)) \\ &\times w^+({}^m h_1, \frac{1}{4}w_{sp} - d_4, \frac{1}{2}w_{sp} - d_4) \\ &+ \left(2w^-(7h_3, d_4, d_2) w^+({}^m h_1, -d_5, -d_3) \right. \\ &\times (1 - w_{u_1}) w^-(8h_3, d_4, d_2)) \\ &+ \frac{1}{2}(1 - w(h_{\max}, \frac{1}{4}w_{sp} - d_4, \frac{1}{2}w_{sp} - d_4, -\frac{3}{4}v_{\max}, -\frac{3}{2}v_{\max})) \end{aligned} \quad (60)$$

$$\begin{aligned} w_{10} &= (1 - w({}^m h_1, \frac{1}{4}w_{sp} - d_4, \frac{1}{2}w_{sp} - d_4, d_1, 0)) \\ &\times w^-(s_3, d_3, 0) \\ &\times 5w^+(\max({}^4h_4, {}^5h_4) + d_3, -\frac{3}{4}v_{\max}, -\frac{3}{8}v_{\max}) \end{aligned} \quad (61)$$

$$\begin{aligned} w_{12} &= ((0.01 + w^+(e_9, d_2, d_6) + 0.1w_{u_1}) \\ &\times (0.05 + w^+(e_{12}, 0, 2v_{\max}))) \end{aligned} \quad (62)$$

REFERENCES

- [1] W. Wang, Y. Song, J. Zhang, and H. Deng, "Automatic parking of vehicles: A review of literatures," *Int. 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 *IEEE Int. Conf. on Vehicular Electronics and Safety*, Beijing, China, 2016, pp. 61–66.
- [3] D. Gonzalez, J. Perez, V. Milanese, and F. Nashashibi, "A Review of Motion Planning Techniques for Automated Vehicles," *IEEE Trans. on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2016.
- [4] P. Petrov, F. Nashashibi, and M. Marouf, "Path Planning and Steering control for an Automatic Perpendicular Parking Assist System," in *Workshop on Planning, Perception and Navigation For Intelligent Vehicles*, Hamburg, Germany, 2015, pp. 143–148.
- [5] P. Petrov and F. Nashashibi, "Saturated Feedback Control for an Automated Parallel Parking Assist System," in *Int. Conf. on Control, Automation, Robotics and Vision*, Marina Bay Sands, Singapore, 2014, pp. 577–582.
- [6] H. Vorobieva, N. Minoiu-Enache, S. Glaser, and S. Mammari, "Geometric Continuous-Curvature Path Planning for Automatic Parallel Parking," in *IEEE Int. Conf. on Networking, Sensing and Control*, Evry, France, 2013, pp. 418–423.
- [7] Y. Yi, Z. Lu, Q. Xin, L. Jinzhou, L. Yijin, and W. Jianhang, "Smooth path planning for autonomous parking system," in *IEEE Intelligent Vehicles Symp.*, 2017, pp. 167–173.
- [8] H. Banzhaf, L. Palmieri, D. Nienhuser, T. Schamm, S. Knoop, and J. M. Zollner, "Hybrid curvature steer: A novel extend function for sampling-based nonholonomic motion planning in tight environments," in *IEEE Int. Conf. on Intelligent Transportation Systems*, Yokohama, 2017, pp. 1–8.
- [9] 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 Intelligent Vehicles Symp.*, Changshu, China, 2018, pp. 326–333.
- [10] C. Chen, M. Rickert, and A. Knoll, "Path planning with orientation-aware space exploration guided heuristic search for autonomous parking and maneuvering," in *IEEE Intelligent Vehicles Symp.*, Seoul, Korea, 2015, pp. 1148–1153.
- [11] H. Banzhaf, P. Sanzenbacher, U. Baumann, and J. M. Zollner, "Learning to Predict Ego-Vehicle Poses for Sampling-Based Nonholonomic Motion Planning," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1053–1060, 2019.
- [12] X. Zhang, A. Liniger, A. Sakai, and F. Borrelli, "Autonomous parking using optimization-based collision avoidance," in *IEEE Conf. on Decision and Control*. IEEE, 2018, pp. 4327–4332.
- [13] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *IEEE Trans. on Control Systems Technology*, 2020.

- [14] H. Vorobieva, S. Glaser, N. Minois-Enache, and S. Mammar, "Automatic parallel parking in tiny spots: Path planning and control," *IEEE Trans. on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 396–410, 2015.
- [15] K. Min and J. Choi, "A control system for autonomous vehicle valet parking," in *Int. Conf. on Control, Automation and Systems*, Gwangju, South Korea, 2013, pp. 1714–1717.
- [16] 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 Int. Conf. on Intelligent Transportation Systems*, Las Palmas, Spain, 2015, pp. 1791–1796.
- [17] 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 Intelligent Vehicles Symp.*, Seoul, South Korea, 2015, pp. 77–82.
- [18] D. Pérez-Morales, O. Kermorgant, S. Domínguez-Quijada, and P. Martinet, "Laser-Based Control Law for Autonomous Parallel and Perpendicular Parking," in *IEEE Int. Conf. on Robotic Computing*, Laguna Hills, USA, 2018, pp. 64–71.
- [19] A. De Luca, G. Oriolo, and C. Samson, "Feedback control of a nonholonomic car-like robot," in *Robot Motion Planning and Control*, 1998, pp. 171–253.
- [20] O. Kermorgant and F. Chaumette, "Dealing with constraints in sensor-based robot control," *IEEE Trans. on Robotics*, vol. 30, no. 1, pp. 244–257, 2014.
- [21] F. Chaumette and S. Hutchinson, "Visual servo control, part I : Basic Approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. December, pp. 82–90, 2006.
- [22] B. Přibyl, P. Zemčík, and M. Čadik, "Camera Pose Estimation from Lines using Plücker Coordinates," in *Proceedings of The British Machine Vision Conf.* British Machine Vision Association, 2015, pp. 45.1–45.12.
- [23] N. Andreff, B. Espiau, and R. Horaud, "Visual Servoing from Lines," *Int. Journal of Robotics Research*, vol. 21, no. 8, pp. 679–699, 2002.
- [24] 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.
- [25] M. Morari and E. Zafiriou, *Robust Process Control*. Englewood Cliffs, New Jersey: Prentice Hall, 1989.
- [26] R. Findeisen and F. Allgower, "An Introduction to Nonlinear Model Predictive Control," in *Benelux Meeting on Systems and Control*, Veldhoven, no. January, 2002, pp. 119–141.
- [27] G. Allibert, E. Courtial, and F. Chaumette, "Visual Servoing via Non-linear Predictive Control," in *Visual Servoing Via Advanced Numerical Methods*, 2010, pp. 375–393.
- [28] O. Thari and Y. Mezouar, "On the efficient second order minimization and image-based visual servoing," in *IEEE Int. Conf. on Robotics and Automation*, 2008, pp. 3213–3218.
- [29] D. Pérez-Morales, O. Kermorgant, S. Domínguez-Quijada, and P. Martinet, "Multi-Sensor-Based Predictive Control for Autonomous Backward Perpendicular and Diagonal Parking," in *Workshop on Planning, Perception and Navigation For Intelligent Vehicles At IEEE/RSJ Iros*, Madrid, Spain, oct 2018, pp. 173–180.
- [30] I. A. Sucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robot. Autom. Mag.*, vol. 19, no. 4, pp. 72–82, 2012.
- [31] S. G. Johnson, "The NLOpt nonlinear-optimization package."
- [32] D. Kraft, "A Software Package for Sequential Quadratic Programming," Institut fuer Dynamik der Flugsysteme, Oberpfaffenhofen, Tech. Rep., 1988.



David Pérez-Morales graduated from CETYS Universidad Tijuana, Mexico in 2014. In 2016 he received the MSc degree in advanced robotics from École Centrale de Nantes, France and, from the same institution, in 2019 he received the PhD degree in robotics. His studies were financed by the Mexican National Council for Science and Technology (CONACYT) from 2014 to 2019. He is currently a research engineer in LS2N. His research interests include mobile robotics, in particular self driving cars and sensor-based control.



Olivier Kermorgant graduated from École Centrale Paris in 2004. From 2008 to 2011 he was with the Lagadic group at Inria Rennes where he received the Ph.D. degree in signal processing from University of Rennes in 2011. He then joined the Ocean Systems Lab at Heriot-Watt University, Edinburgh, as a Research Assistant. He was Assistant Professor at University of Strasbourg from 2012 to 2014. Since 2015 he has been Associate Professor at Centrale Nantes. His research interests include sensor-based control, disturbance rejection and optimization.



Salvador Domínguez-Quijada graduated from Universidad de Valladolid, Spain in 1999. From 2000 to 2013 he was with Cartif research center Foundation at Valladolid and received the Ph.D. degree in Social Robotics from University of Valladolid, Spain in 2007. He was granted with an internship at Fraunhofer Research Institute IAIS in Sankt Augustin, Germany in 2011. In 2013 he joined the former IRCCyN now called LS2N in Centrale Nantes, France. His research interests include autonomous navigation in general and self driving cars in particular.



Philippe Martinet graduated from the CUST, Clermont-Ferrand, France, in 1985 and received the Ph.D. degree in electronics science from the Blaise Pascal University, France, in 1987. From 1990 to 2000, he was assistant Professor with CUST. From 2000 until 2011, he has been a Professor with Institut Français de Mécanique Avancée (IFMA), Clermont-Ferrand. In 2006, he was a visiting professor at the Sungkyunkwan university in Suwon, South Korea. In September 2011, he moves to Ecole Centrale de Nantes and LS2N. In November 2017, he moves to Inria Sophia Antipolis as Research director. His research interests include visual servoing of robots, multi-sensor-based control, force vision coupling, autonomous guided vehicles, modeling, identification and control of complex machines. From 1990, he is Author and Coauthor of more than three hundred forty references.