Multi-sensor-based control in Intelligent Parking applications

David Perez Morales

THÈSE DE DOCTORAT DE

L'ÉCOLE CENTRALE DE NANTES
COMUE **U**NIVERSITE **B**RETAGNE **L**OIRE

Ecole Doctorale N°601
Mathèmatique et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Automatique, productique et robotique*
Par

« David PÉREZ MORALES »

« Multi-sensor-based control in Intelligent Parking applications »

Thèse présentée et soutenue à NANTES, le 6 Décembre 2019
Unité de recherche : Laboratoire des Sciences du Numérique de Nantes

Rapporteurs avant soutenance :

Florent LAMIRAUX, Directeur de recherche CNRS, LAAS

Roland LENAIN, Directeur de recherche, IRSTEA

Composition du jury :

Présidente :	Isabelle FANTONI, Directrice de recherche CNRS, LS2N
Examinateurs :	Fawzi NASHASHIBI, Directeur de recherche Inria, Inria Paris
	Guillaume ALLIBERT, Maître de conférences, Université de Nice Sophia Antipolis
Dir. de thèse :	Philippe MARTINET, Directeur de recherche Inria, Inria Sophia Antipolis-Méditerranée
Co-encadrants :	Olivier KERMORGANT, Maître de conférences, Centrale Nantes
	Salvador DOMÍNGEZ QUIJADA, Ingénieur de recherche, LS2N

Acknowledgements

I would like to express my most sincere gratitude to my thesis director Philippe Martinet and co-supervisors Olivier Kermorgant and Salvador Domínguez for supporting and guiding me during this three-year journey. Their valuable feedback and advises were key for me to succeed in this endeavor.

I would like to thank as well to the jury members for showing interest in my work, even in the face of some rather complicated logistical problems for attending to my thesis defense.

I want to thank to the members of the ARMEN team for their friendliness and for the discussions (scientific or not) that took place along the past three years. I want to express my gratitude in particular to Arnaud Hamon for his support on the integration and experimentation with the vehicles and to Gaëtan Garcia for his help on certain experiments and demonstrations.

I want to thank as well to the Mexican National Council for Science and Technology (CONACYT) for providing the funding that allowed me the pursue this PhD.

Thanks to my parents Silverio and Leticia and to my brother Jorge for their never ending support and encouragement, even at a distance.

Thanks to Angie, Nallely and the rest of my friends that are on the other side of Atlantic Ocean. Thanks for their support, patience and comprehension.

Thanks to my friends Stephanie, Rafael, Vyshak, Catherine, Saman, Antoine and to my colleagues for the numerous discussions and memorable moments that we have shared.

Finally, I would like to thank to my girlfriend Marlène for her patience, love, support and encouragement.

Table of Contents

List of Figures

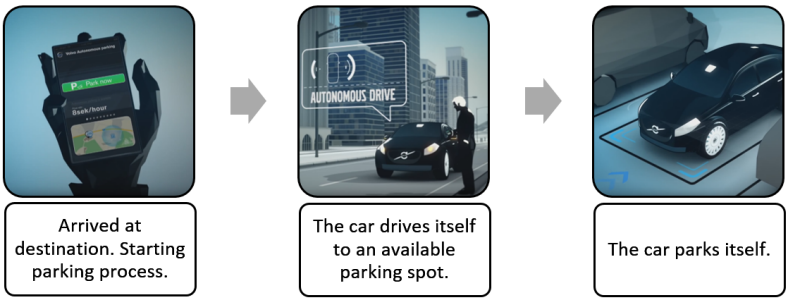
List of Tables

Introduction

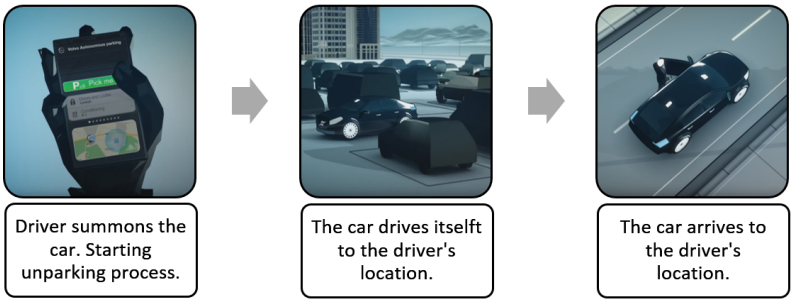
Vehicles and Intelligent Transportation systems have followed a quick mutation thanks to the growing use of electric technology and the emergence of autonomous navigation techniques. So much so that the hype created around self-driving vehicles even lead parts of the industry to overestimate their arrival [1] as commercial products. Even if often overlooked when talking about autonomous driving, the amount of maneuvering that encompass the different tasks that a driver would perform in a parking scenario (look for an empty parking spot, park, unpark, etc.) - even more so in crowded places, contribute in no small part to the complexity of developing a fully autonomous driving system [2].

In [3] it is estimated that 30% of the traffic in any city is people in cars searching for parking. A different study [4] says that 45% of total traffic on 7^{th} Avenue in Brooklyn is people looking for a parking space. According to [5], the burden of finding a free parking spot can be such that 100% of car owners in a couple of districts of Paris have abandoned their trip at least once. The statistics of a survey carried by the *University of Michigan Transportation Research Institute* shows that about 10,000 of 12 million traffic accidents occur while entering or leaving parking lots although it is affirmed that the actual number of parking related accidents is much more than the reported number [6]. Currently commercially available parking assistance systems, using mainly cameras and ultrasonic sensors to perceive the environment, have lead to a significant increase of driving comfort and provided cost savings from (avoidance of) accidents [7]. It is clear that advanced Intelligent Parking Systems would be of great use.

Even if many car manufactures already offer the first generation of automatic parking systems, it is still necessary to imagine the new generation. The most complete solution for the parking problem, and the dream of many, would be to be able to, once the driver arrives to the desired destination, leave the car at the entrance of the building or at the entrance of a dedicated parking lot (if any) and forget about it, letting the car park by itself safely and autonomously in an available (parallel, perpendicular or diagonal) parking spot (Figure 1a) and, whenever the user wants leave, be able to summon the car remotely to go to the driver's location and pick him/her up (Figure 1b). Therefore, the new generation of Intelligent Parking Systems requires the implementation of different advanced



(a) Car parking autonomously



(b) Car unparking autonomously

Figure 1: Autonomous Parking - Volvo Cars Innovations [8]

functionalities: move and look for an empty place autonomously in an encumbered environment, check the parking maneuver compatibility, perform the parking and unparking maneuvers, navigate autonomously from the parking spot to the place to pickup the user and obviously to communicate with the user and the parking exploitation company.

The mobile robotics research axe of the LS2N's ARMEN team is involved on the development of autonomous vehicles applications. As part of its efforts, the team participates in the ANR project VALET. This project studies a full Valet Parking System to be implemented at a city level. Thus, the work done during the PhD falls under the umbrella of the ANR project VALET, aiming to address the parking related tasks.

As such, the main objective of the PhD is to revisit the tasks of parking and other maneuvers with the multi-sensor based approach - an approach that has proven its efficacy and robustness in many different robotic applications, and compare them with already existing approaches. In order to attain the objectives it is necessary to understand the nonholonomic constraints inherent to car-like robots and the implications they have when doing path planning - a typical approach followed in the literature for solving motion problems, or while controlling the vehicle. Moreover, since (as the name implies) sensor-

based control approaches rely heavily on the sensor information available, the perception problem has to be taken into account. Regarding the control problem, a common approach in robotics is to make use of the task function approach [9] which allows to specify the task to be performed as an output function associated to the state equation of the robot. Nevertheless, due to safety and comfort reasons, control techniques that are capable of explicitly dealing with constraints, such as Model Predictive Control (MPC), are of interest for our application.

Thanks to the work done during the PhD thesis, two different frameworks were developed. The first framework, using a Multi-Sensor-Based Control (MSBC) approach, allows to formalize different parking and unparking operations in a single maneuver with either backward or forward motions. Building upon the first one and by using an MPC strategy, a Multi-Sensor-Based Predictive Control (MSBPC) framework was developed, allowing the vehicle to park autonomously (with multiple maneuvers, if required) into perpendicular and diagonal parking spots with both forward and backward motions and into parallel ones with backward motions in addition to unpark from parallel spots with forward motions. These frameworks have been tested extensively using a robotized Renault ZOE with positive outcomes and now they are part of the autonomous driving architecture being developed at LS2N.

In the first chapter an overview of the state-of-the-art parking techniques is presented, followed by related works on sensor-based control applied to car-like robots, an introduction and key aspects of Model Predictive Control (MPC) strategies and an overview of the relevant perception systems for intelligent vehicles. Chapter 2 introduces the modeling and notation for car-like robots and the multi-sensor framework as well as the considered perception system. The following chapter presents the work done on the MSBC approach for performing parking and unparking operations in a single maneuver. The MSBPC for parking in multiple maneuvers is presented in Chapter 4. Finally, general conclusions and perspectives for future work are discussed.

Publications

Journal Articles

D. Pérez-Morales, O. Kermorgant, S. Domínguez-Quijada, and P. Martinet, "Multi-Sensor-Based Control For Autonomous Parking", *IEEE Transactions on Intelligent Vehicles.*, *Submitted*, 2019.

Conference Proceedings

D. Pérez-Morales, O. Kermorgant, S. Domínguez-Quijada, and P. Martinet, "Automatic Perpendicular and Diagonal Unparking Using a Multi-Sensor-Based Control Approach", *in 2018 15th International Conference on Control, Automation, Robotics and Vision*, Singapore, Singapore: IEEE, Nov. 2018, pp. 783–788.

D. Perez-Morales, O. Kermorgant, S. Dominguez-Quijada, and P. Martinet, "Laser-Based Control Law for Autonomous Parallel and Perpendicular Parking", *in Second IEEE International Conference on Robotic Computing*, Laguna Hills, USA, 2018, pp. 64–71.

Peer-reviewed workshops

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 Workshop on Planning, Perception and Navigation for Intelligent Vehicles at IEEE/RSJ IROS*, Madrid, Spain, 2018.

——, "Autonomous Perpendicular And Parallel Parking Using Multi-Sensor Based Control", *in 9th Workshop on Planning, Perception and Navigation for Intelligent Vehicles at IEEE/RSJ IROS*, Vancouver, Canada, 2017.

Chapter 1

Related work

In this chapter an overview of the work related to Intelligent Parking applications is presented. We start by defining the relevant classifications that exist in the literature to categorize self-driving cars in general and Parking and Maneuvering Assistance Systems (PMAS) in particular as well as an overview of the current industrial state. Afterwards, we review the state-of-the-art on parking techniques with a particular focus on path planning approaches - a popular choice in recent years. Then, the basic concepts of sensor-based control are introduced as well as the few works available on the literature applied to car-like robots. Then, the basics of Model Predictive Control (MPC) strategies and their essential properties are given as well as some examples of applications exploiting predictive techniques followed by a key framework that combines sensor-based control (i.e. Visual Servoing (VS)) and MPC strategies. Finally, the perception issue is briefly introduced alongside an overview of the key characteristics of the different sensors typically considered for autonomous vehicles.

1.1 Relevant classifications

When talking about self-driving cars, probably the two most relevant classifications that appear in the literature are, on the one hand, the one provided by SAE International [10] and, on the other hand, the one given by the *International Organization of Motor Vehicle Manufacturers* (OICA by its French acronym) [11], with the latter being based on the former. Both define six levels of automation (from level 0 to 5) where the lower level is used to define a *driver only* level (i.e. no active systems are involved) while the highest one (level 5) is used to denote *full automation* - the vehicle is able to drive itself under all conditions with no human intervention. When the vehicle is able to control either the longitudinal or the lateral motion of the vehicle one is talking about a system of level 1 but if the car is able to perform both tasks one refers to level 2 or higher. The difference between the higher levels (2 and above) lies on the requirement (or lack of thereof) of a

driver with level 2 requiring the driver to supervise the vehicle at all times while level 3 requiring the driver to be in position to retake control of the vehicle (if the system deems it necessary) but not actively supervising it. As for the fourth level, no driver is required in defined use cases.

Considering that our application has a well defined scope, the classification of Parking and Maneuvering Assistance Systems (PMAS) defined in the Handbook of Intelligent Vehicles [7] is of relevance as well. In this one, two main categories are defined: *Passive* and *Active* systems.

Passive PMAS refer to those that do not interact with the actuators of the vehicle (i.e they do not perform any control) but rather are used to warn the driver about nearby objects, inform if a given parking spot is suitable for the vehicle or provide video feeds from one or multiple cameras (often with auxiliary lines superimposed) to help the driver to best steer the vehicle into the parking spot. As the reader might have already guessed, passive PMAS overlap with the level 0 of driving automation defined above.

Regarding active PMAS, there are three subcategories: *semiautomatic*, *fully automatic* and *autonomous* parking assistance. Similarly to the previously defined levels of driving automation, the difference among these systems lies on the level of involvement of the driver.

Semiautomatic parking assistance overlaps with level 1 systems since this type of systems only control the steering wheel while the driver has to control the longitudinal motion of the vehicle. To the best of my knowledge, the first commercial offer of this type of assistant was made by Toyota on 2003 in Japan under the name of Intelligent Parking Assist. [12]. Many other car manufacturers (Ford, Lincoln, PSA Peugeot Citroën, BMW, Volkswage, Nissan, Mercedes-Benz, Land Rover, among others) [13] [14] [15] currently offer (or had offered) this type of systems.

Fully automatic systems refer to those that are capable of performing the whole parking maneuver but, depending on whether or not the driver is expected to actively supervise the vehicle, they can be part of either level 2 or 3 systems. The first commercial offers were provided by BMW [16] and Tesla Motors [17]. Many other manufacturers have working prototypes.

The most advanced subcategory, autonomous parking assistance, refers to systems that do not require human intervention nor supervision and, depending on whether they can work under any circumstance or only for defined use cases they can be considered either level 5 or 4 systems. Currently there are no commercial offers that have reached

this category although Audi [18], BMW [19] and Tesla Motors [20] have shown working prototypes. It should be noted that, Tesla's situation on this regard is a special case given that they have early access programs for some of their features such as Enhanced Summon [2] thus being (to some extent) commercially available but not considered as a finished feature.

1.2 State-of-the-art parking techniques

As it can be seen in [6], the literature related to automatic parking is quite extensive, having many different control approaches available. Despite the fact that the automobile industry has already started to roll out some commercial implementations of active parking assistants capable of actively controlling acceleration, braking and steering [21], the research interest in the topic remains strong.

The different control approaches available in the literature can be divided into two categories [22]: one based on stabilizing the vehicle to a target point, with fuzzy control being the most widely investigated from this category and with one of the earliest works on automatic parking [23] using sinusoidal control functions; the other is based on path planning, where geometric collision-free path planning approaches based on the non-holonomic kinematic model of a vehicle have been of special interest in recent years [24]–[26]. Relatively typical parking control theories are fuzzy control, neural network control and control method based on linear inequalities [6]. As for combined planned/reactive approaches, a pioneering work in the matter is reported in [27], where a behavior based architecture is used to combine the contributions of the different planned and reactive behavior towards a common goal: parking while avoiding collision.

In [28] a neuro-fuzzy sensor based controller is presented. The controller is able to decide about the motion direction at each time interval by processing the information obtained from the sensors. The key idea of their scheme is to construct a mapping between sonar sensors measurements and the turning angle in order to plan and carry out the sensor based maneuvers required for parking and, since the mapping is directly established between the sensor measurements and the turning angle there is no need to deal with radial imprecision and angular uncertainty.

In [29] a technique for integrating metric and semantic maps for vision-only automated parking is reported. Said maps are built and maintained using the data obtained from the cameras mounted on the car. In the semantic maps, two types of semantic labels are

considered: static and dynamic. From those maps, the car used to collect the data to build them, was able autonomously drive inside the parking lot and park in a predefined parking slot with a relatively good accuracy, although the approach used to realize the parking tasks is not specified.

Expanding slightly the scope of the state-of-the-art, one can find works on docking of electric cars for recharging purposes [30] as well as for academic mobile robots [31]. In [30], the pose of the vehicle is reconstructed using a infrared perception system and a known infrared pattern on the charging base in order to stabilize the pose of the vehicle at the desired value. As for [31], a given path is deformed in a reactive fashion using sensor data in order to avoid collision and/or to reach a desired (perceived) docked pose; experimental results have been shown for collision avoidance and docking using a differential mobile robot towing a cart [31] and, using the same approach, collision avoidance results have been shown for a car-like robot in [32].

As show in [33], path planning approaches for automated vehicles have been heavily investigated in recent years.

In [34] and [25] techniques to perform, respectively, parallel and parking maneuvers where the path planning step is addressed by a combination of admissible collision-free circulars arcs and tangent straight lines are presented. In both techniques, the problem of stabilizing the vehicle at a desired position and orientation is seen as an extension of the path tracking problem. Saturated controllers are proposed to achieve a quick steering without chattering when performing the parking maneuvers.

Following similar geometric principles, the approach reported in [35] relies on three sub-paths: two circles and a straight line, each one tangent to the next one. This work addresses forward and backward perpendicular parking maneuvers and backward parallel ones.

A path planning approach that considers clothoids (a curve whose curvature varies linearly with its arc length) for parking into parallel spots, for both forward and backwards motions, can be found in [26]. The use of clothoids is considered in order to account for the limitations on the steering velocity of the vehicle, answering to the admissibility constraints of car-like vehicles.

Similarly, to plan smooth paths, in [36] the authors make use of clothoid-like curves. Parking into non-parallel parking spots, in addition to parallel ones was considered in this work although only for backward maneuvers.

The description of a complete Autonomous Valet Parking System architecture is given

in [37]. The authors localize the vehicle in a known environment by means of vision approaches while the obstacle detection is performed using ultrasonic sensors. Inside this environment, the car is able to park autonomously in a previously taught parked pose from a learned (or in the vicinity of it) departure position. If the vehicle finds itself out of the known path, new local paths that bring the vehicle to the usual one are computed using geometric methods like the ones reported in [38].

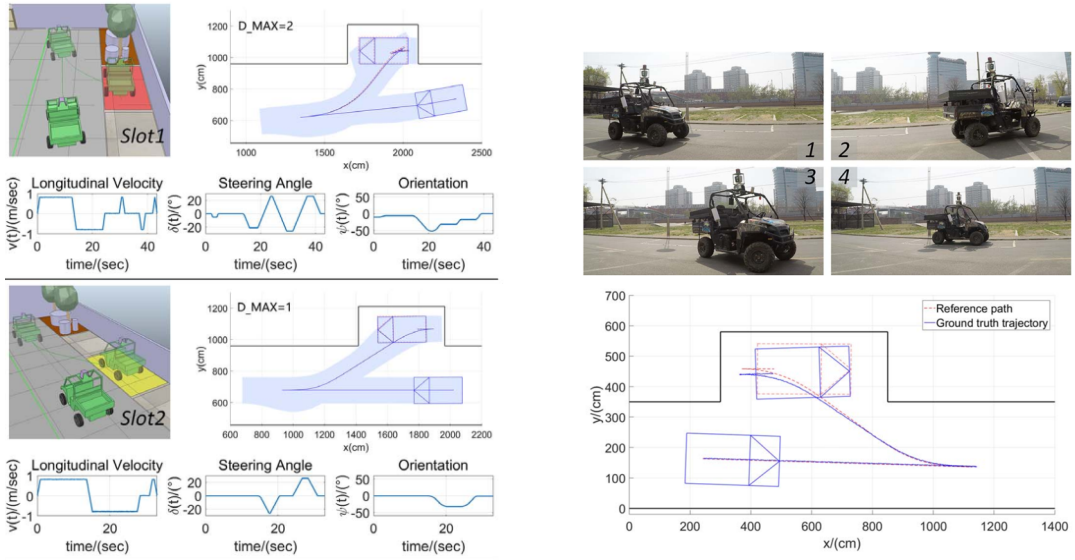
The heuristic exploration-based planning approach reported in [39] makes use of the knowledge of the vehicle's orientation during the exploration phase in order to improve the planning efficiency in maneuvering scenarios when compared to common random-sampling methods and general heuristic search methods. The (rather short) computation time claimed by the authors could allow this technique to be used for online replanning.

From the RRT-based path planning approaches, one can highlight [40]–[42]. These three works use the same BiRRT* backbone planner with the difference being on how the sampling is guided. In [40] a Hybrid Curvature (HC) steering function based on clothoids is introduced in order to ensure continuity on the steering angle while the vehicle is moving in one direction but allows curvature discontinuities where a change of (longitudinal) direction occurs. In order to ensure that the curvatures respect the actuator limits regarding the maximum acceleration, in [41] the sampling is guided using Hybrid Curvature Rate (HCR) steering functions. The HCR steering function is based on cubic spirals. Finally, in [42] the sampling is guided by a Convolutional Neural Network (CNN). In this case, the CNN has been trained with simulated data. Its goal is to predict the future poses of the vehicle given the current driving situation in order to guide the planner towards an optimal solution.

An (often) overlooked point is that, solving the motion planning problem does not equal to solving the maneuvering one. Due to the nonholonomic constraints that the vehicle is subject to, small discrepancies between the model and the real platform, noise in the sensor measurements, localization errors, external disturbances, etc. could potentially produce non-negligible discrepancies between the planned path and the actual path performed by the vehicle, especially as the amount of changes of longitudinal direction increases.

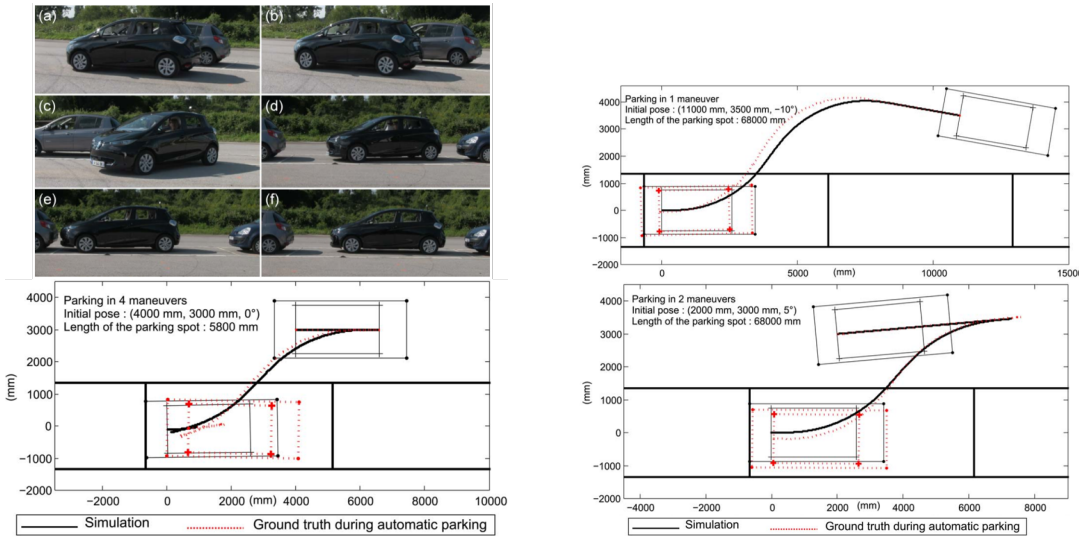
Taking as an example Fig. 1.1 one can consider the planning problem to be addressed successfully and, looking at the simulations results (Fig. 1.1a), the maneuvering problem seems to be solved as well but, when one looks at the experimental results (Fig. 1.1b) it is clear that the vehicle doesn't follow exactly the planned path. Similar outcomes can

be seen in Fig. 1.2 where the discrepancies between the planned and the actual path are non-negligible.



(a) (b)

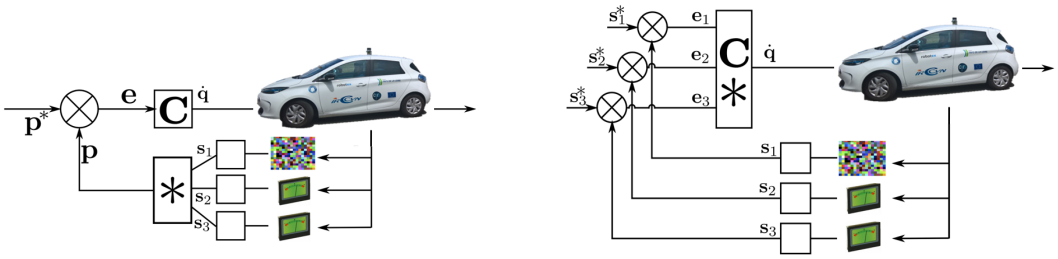
Figure 1.1: Results presented in [36]: (a) simulations and (b) real experimentation



(a) (b)

Figure 1.2: Real experimentation results presented in [26]: (a) parking in four maneuvers, (b) parking in one and two maneuvers

1.3 Sensor-based control applied to car-like robots



(a) Sensor data are used to estimate the position of the robot

(b) Sensor data are directly integrated in the control loop

Figure 1.3: Block diagrams of (a) classical position control and (b) multi-sensor-based control

Contrary to the original use of sensors in classical position control where their data are used to provide position feedback (Fig. 1.3a), sensor-based control techniques integrate the sensor data directly in the control loop (Fig. 1.3b). As such, is not the robot anymore that should reach a given desired position but instead is the sensor that should perceive a given desired value (e.g. a camera that should view a certain image, an ultrasonic sensor that should measure a given desired distance, etc.).

Sensor-based control techniques based on the task function approach [9] (i.e. VS [43], [44]) have proven its efficacy in many robotics applications (manipulation, grasping, parallel robotics, mobile robotics, etc.) although few works are available in the literature on car-like robot applications. A key benefit of these techniques is their great robustness against modeling and calibration errors, particularly when then pose of the robot doesn't need to be reconstructed (i.e. Image-Based Visual Servoing (IBVS)) and, since they rely only in locally perceived information, they do not suffer from localization issues. On the contrary, given that the task is expressed in the sensor space, the trajectory of the robot depends heavily on the choice of sensor features and parametrization (Fig. 1.4), therefore it is difficult to predict its behavior without testing it beforehand.

A visual servoing approach for path reaching and following is reported in [45] requiring only a few path features. Real experimentation using a car-like robot and comparisons between Position-Based Visual Servoing (PBVS) and IBVS are presented, showing that, if the control law is well designed and the features remain visible, the IBVS approach is generally better, having smoother control signals and being more robust against calibration errors.

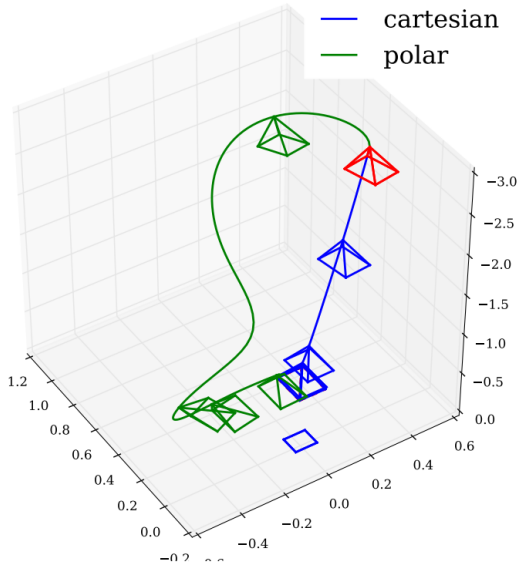


Figure 1.4: Trajectories of a flying camera using an image point as the sensor feature parameterized using Cartesian and polar coordinates

In [46] the navigation task of the car-like robot is performed autonomously using vision while avoiding the obstacles detected by a 2D LiDAR. The path to follow consist of an ordered set of key images that have been previously acquired during the teaching phase.

In an attempt to minimize the issues caused by degraded localization performance in urban environments, sensor-based control approaches have been recently investigated in the context of navigation [47] and dynamic obstacles avoidance [48].

Regarding maneuvering tasks (e.g. parking, unparking, half turn), few things have been investigated [49].

1.4 Model Predictive Control

The term Model Predictive Control (MPC) does not refer to a specific control strategy but rather to a wide variety of control methods that explicitly use a model of the process in order to obtain the control signal by minimizing an objective function [50]. The origins of this type of strategies can be tracked back to [51], initially proposed for industrial processes. MPC is considered a mature strategy for linear and slow systems while nonlinear, hybrid or very fast processes were considered beyond the realm of MPC in no small part due to computational demands. Nonetheless, thanks to advances in computing capabilities and the associated cost reduction, MPC techniques have been investigated

for many different (previously considered as unfeasible) applications in recent years [52] - e.g. automotive, health care, finance, etc.

The basic ideas of this type of techniques are to explicitly use a model to predict the output of the process at future time instants up to a given horizon, the computation of control sequence that minimizes a given objective function and the use of a receding strategy in order to displace the horizon towards the future and, at each step applying the first control signal of the computed sequence (Fig. 1.5).

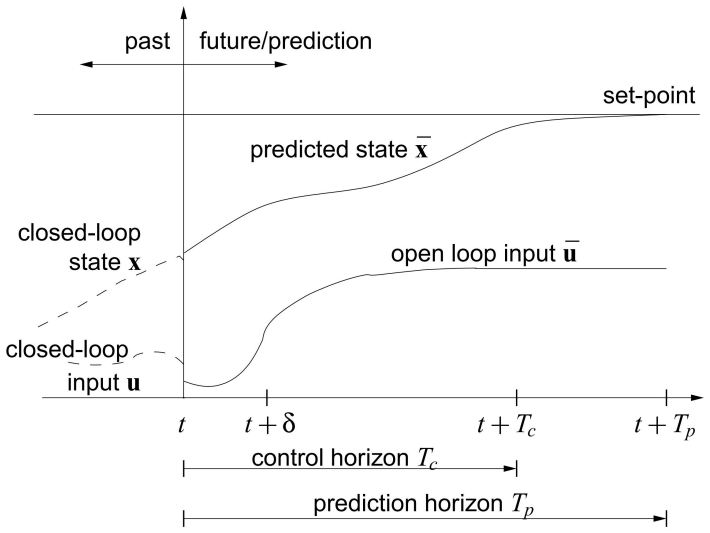


Figure 1.5: Temporal diagram of the finite-horizon prediction [53]

A key benefit of MPC strategies is that, when using a finite-horizon, it is possible to explicitly handle constraints on the states, inputs and outputs. On the contrary, from a theoretical point of view, when using a nonlinear model the control problem is a non-convex Non-Linear Program (NLP) thus the solution is rather complex and there is no guarantee that the global optimum can be found [50].

In spite of the challenges that nonlinear problems pose for MPC strategies, for differential mobile robots, such techniques have been proven to be feasible for stabilizing the robot at a desired pose [54] and for performing trajectory tracking [55]. Furthermore, in [56] it was shown that it is possible to generate a reactive walking pattern for a humanoid robot in real time using a Nonlinear Model Predictive Control (NMPC) strategy.

In [57]–[59] predictive strategies were developed to compensate for the delay in the response of the steering angle of off-road car-like vehicles when following a given path by exploiting the knowledge of future curvature values along the path. A single robotized

tractor was used to validate the approach in [57], [58] while in [59] multiple robots in a given (potentially varying) formation were considered.

In recent years the use of MPC as part of Artificial Intelligence (AI) algorithms has been investigated [60], [61]. In [60] an efficient (both in computation time and memory usage) differentiable MPC is presented and used as a policy class for reinforcement and imitation learning. As for [61], based on an MPC strategy, the dynamics of the agent are learned with a model-based reinforcement learning algorithm and afterwards the learned dynamics are used to initialize a model-free learner to improve the final results. The authors claim that the hybrid approach is able reach comparable results to purely model-free learners but requiring significantly less samples.

Regarding the choice of the control horizon, it has been shown in [62] that, for a car-like robot described in natural coordinates, the minimal control horizon required in order to control its full state has to be equal to 3 (equal to the degree of nonholonomy - 1).

The framework presented in [63] formalizes the IBVS subject to constraints (workspace limitations, visibility constraints and actuator limitations) with a predictive control strategy by writing the IBVS task as a nonlinear optimization problem. On the one hand, thanks to the predictive strategy, the different considered constraints can be explicitly handled. On the other hand, the well-known robustness against noise and modeling errors of sensor-based control techniques is preserved as well as the local asymptotic stability property of such type of approaches. Additionally, it is shown that a local model based on the interaction matrix is enough to predict the evolution of the sensor features and thus allows to perform successfully VS tasks while satisfying the considered constraints. Given that this framework links MPC and sensor-based control strategies, it will be key for the development of Chapter 4 where a novel approach for performing parking/unparking maneuvers (starting from virtually any initial position in the workspace) in a reactive fashion is presented.

1.5 Perception

In order to perceive the environment, autonomous vehicles have to be equipped with several sensors. The most relevant types of exteroceptive sensors for intelligent vehicles (Fig. 1.6) are radar, laser, vision and ultrasonic [64]. Each type of sensors has different inherent characteristics and price points that make them more or less suitable for certain tasks.

(a) Radar

(b) Laser

(c) Ultrasonic

(d) Vision

Figure 1.6: Examples of exteroceptive sensors for intelligent vehicles

Radars can provide information about the distance, velocity and angle of detection of perceived objects. These type of sensors are in general rather robust against adverse weather conditions and the detection range (at least for long-range radars) is often the longest (up to 200 m) although the Field Of View (FOV) is often very narrow.

Laser scanners or Light Detection And Ranging (LiDAR)s take measurements according to the time-of-flight principle. The detection range for this type of sensors is usually around 100 m although the newest versions are capable of reaching 300 m. The lateral resolution and FOV is considerably better for this sensors than for radars although in general are less robust against adverse weather conditions. With enough layers, the amount of detail provided can rival vision systems (not considering the color) although increasing layers usually means a rather significant increase in price. Often in both academic and industrial research setups, the LiDARs used have moving parts to achieve a lager horizontal FOV (i.e. rotating LiDARs) although recently developed solid sate LiDARs aim to improve the durability of the sensor and reduce its cost at the expense of FOV [65].

Ultrasonic sensors are the most widely adopted in the industry because of their low cost, lightweight, low power consumption and low computation requirements compared to other ranging sensors. On the other hand they have the shortest range and are able only to tell whether or not there is an object in the detection range and at which distance

with no information on the angle, size or shape.

Vision systems are getting more popular in the industry as time goes by due to their reasonable cost and the amount of information they can provide, allowing to detect and classify traffic signs, pedestrians, vehicles, etc. Depending on the type of wavelength perceived, cameras can be classified as visible (VIS) or infrared (IR) for passive ones or as (active) Time-of-Flight (ToF) if the system uses the time-of-flight principle to measure distance by emitting near infrared (NIR) light pulses [65]. Nevertheless, the amount of information comes at the cost of relatively high computational processing requirements. Depending on the task to accomplish one could consider monocular vision systems (mostly for detection and classification) or stereo vision systems when depth information is required and the range of ToF is not enough. The performance of these systems is often very dependent on the environment and visibility conditions.

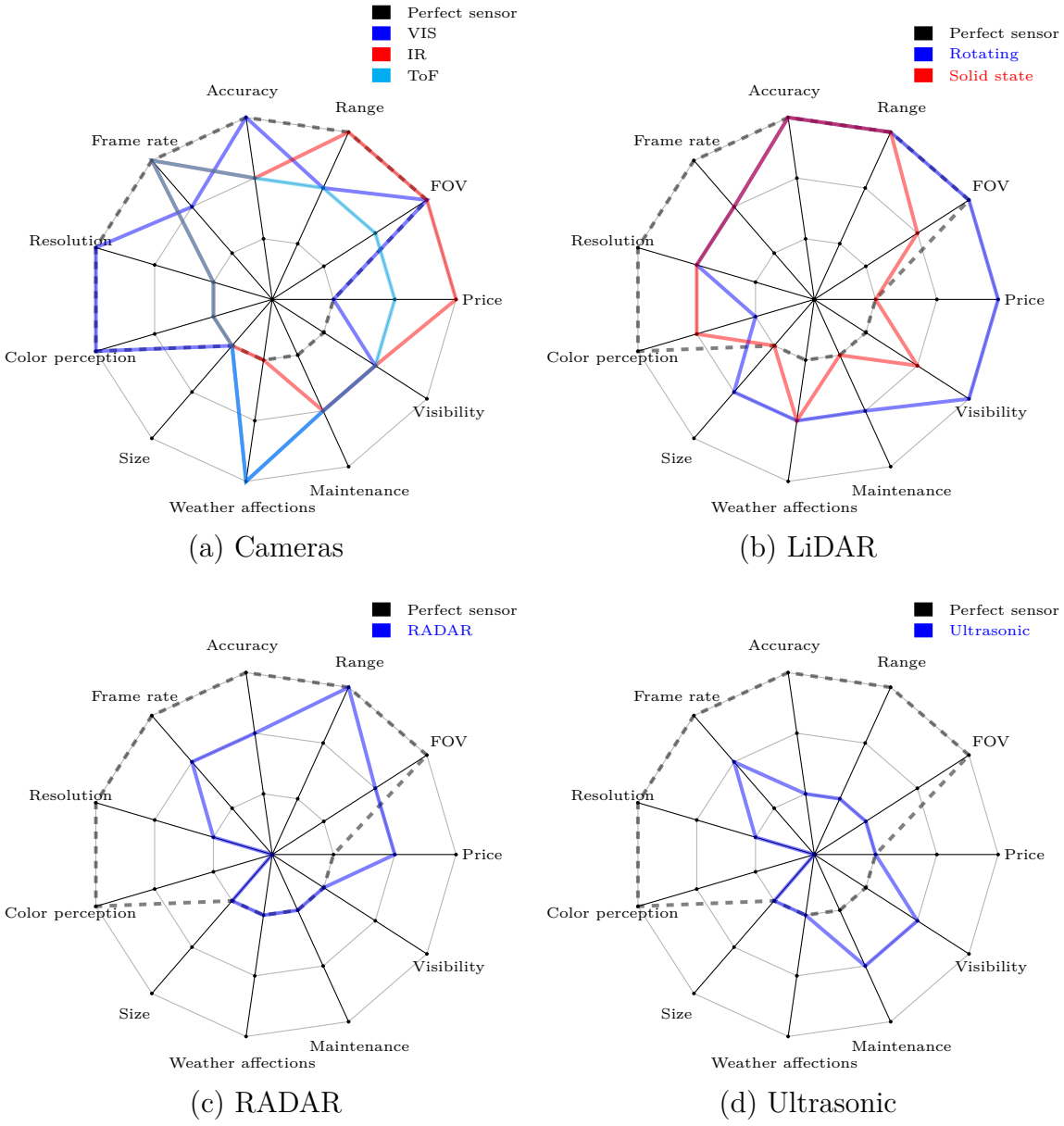
The key characteristics of different sensors typically used in intelligent vehicles applications are well summarized in [65] (Fig. 1.7) and compared against a *perfect* sensor. It can be clearly seen that no sensor can be considered *perfect* although the closest one, depending on the perception requirements, could be argued to be VIS cameras with both LiDAR variants following closely.

Regarding the perception algorithms available in the literature to extract parking spots from sensors embedded in the vehicle, we now give an overview of different works using different (combinations) of sensors.

Having knowledge of the parking area stored in a digital map - including parking spots, the authors of [66] simply use the sensor data (four 2D LiDARs) to build an occupancy grid. Using this occupancy grid and an occupancy Bayes' formula, the state (free or occupied) of a given parking spot is determined.

Considering a system where the user selects the desired parking spot by touching a display showing a rear view image, the authors of [67] define a Region Of Interest (ROI) in which the system looks for a parking spot. The parking spot is extracted from a bird's-eye view using a Hough transform.

In [68] an algorithm that fuses vision data from a bird's-eye view (built from four cameras, one on each side of the vehicle) and ultrasonic sensors is reported. The work focuses on the detection and tracking of parking spots in underground and indoor environments. Due to this, only rectangular parking spots are considered. Additionally, pillar information (typical in such environments) is exploited to improve the performance of the approach.



(a) Cameras (b) LiDAR

(c) RADAR (d) Ultrasonic

Figure 1.7: Comparison of the main features of different sensors typically used in intelligent vehicles according to [65].

Using as well bird's-eye view images, in [69] an algorithm capable of recognizing free parking spots of any type (perpendicular, diagonal and parallel) is presented. The authors use a height map constructed by means of dense motion-stereo vision in order to distinguish between road and obstacles. To extract the road markings, which are assumed to have a fixed thickness, a probability map with a salient feature is used and, finally, the parking spot recognition is performed by means of a Bayesian classifier.

Using 2D LiDAR information, [70] presents an algorithm capable of extracting and tracking empty perpendicular parking spots from the free space between two detected vehicles. The authors assume that the already parked vehicles have an L-shape and as such, they use rectangular and round corner detection as the basis of the approach.

In [71], the data of a 3D LiDAR with 32 layers is used to map the environment using a Simultaneous Localization and Mapping (SLAM) approach. From this map, the already parked vehicles are extracted using a CNN. By fusing the map and the detected vehicles, the authors estimate the topology of parking lots and infer vacant parking spots in a graph-based approach.

1.6 Conclusion

Following the classifications introduced in this chapter, we are aiming to reach a level 4 and therefore an autonomous parking assistance system level although, considering that the work done during the PhD thesis was focused on the control issue, for the most part, we depend on external perception algorithms in order to tell whether or not our system is actually able to reach the desired level.

Regarding the overview of the state-of-the-art parking technique, an often overlooked point has been identified when working with nonholonomic mobile robots: solving the motion planning problem does not equal to solving the maneuvering one.

We have also seen that, in spite of the interesting properties that sensor-based control approaches bring to the table and the fact that it has been proved effective in many different robotic applications, few work has been done on car-like robot applications. Additionally, a key framework that combines sensor-based control and MPC strategies [63] was introduced.

Regarding the perception issues, we have introduced most relevant exteroceptive sensors for intelligent vehicles applications as well as the main benefits and drawbacks of each of them. Furthermore, some works on parking spot extraction using different (combinations of the) sensors considered were presented.

Chapter 2

Modeling and notation

In this chapter we define the modeling considered for the car-like robot as well as the multi-sensor-based modeling formalism which later will allow to define the required task and constrained sensor features and explain how they relate to the vehicle's velocity. Additionally, the perception system configuration is defined, an example of how an empty parking spot could be extracted from the sensor data is given as well as how, by using virtual sensors the definition of the different sensor features can be more direct. Finally, the parametrization used for modeling straight lines (crucial for our application) is given.

2.1 Car-like robot model and notation

Given that parking applications are generally executed at low-speed motions, a kinematic model can be considered as accurate enough.

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. Furthermore, all the sensors are assumed to be calibrated.

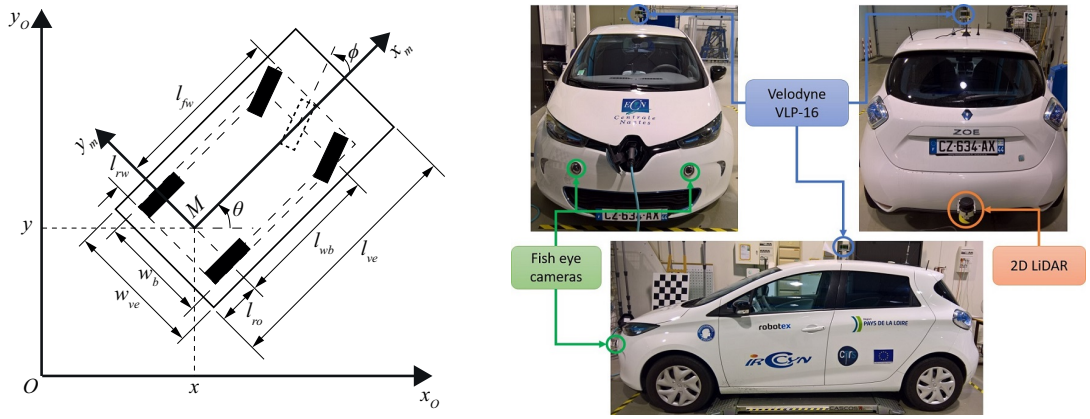
Considering the well-known kinematic model of a car with rear-wheel driving [72]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ \tan\phi/l_{wb} \\ 0 \end{bmatrix} v_{x_m} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \dot{\phi}, \tag{2.1}$$

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

$$\mathbf{v}_m = [v_{x_m}; \dot{\theta}_m], \tag{2.2}$$

where v_{x_m} and $\dot{\theta}_m$ are, respectively the longitudinal (along x_m) and rotational velocities



(a) (b)

Figure 2.1: (a) Kinematic model diagram for a car-like rear-wheel driving robot. (b) Robotized Renault ZOE used for real experimentation

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:

$$\dot{\theta}_m = \frac{v_{x_m} \tan \phi}{l_{wb}}. \tag{2.3}$$

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

$$\mathbf{v}_r = [v_{x_m}; \phi] \tag{2.4}$$

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

$$\rho_m = \frac{l_{wb}}{\tan \phi}. \tag{2.5}$$

The vehicle used for experimentation and simulation, represented by its bounding rectangle in Fig. 2.1a, is a Renault ZOE (Fig. 2.1b). Its relevant dimensional parameters are presented in Table 2.1..

Table 2.1: 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

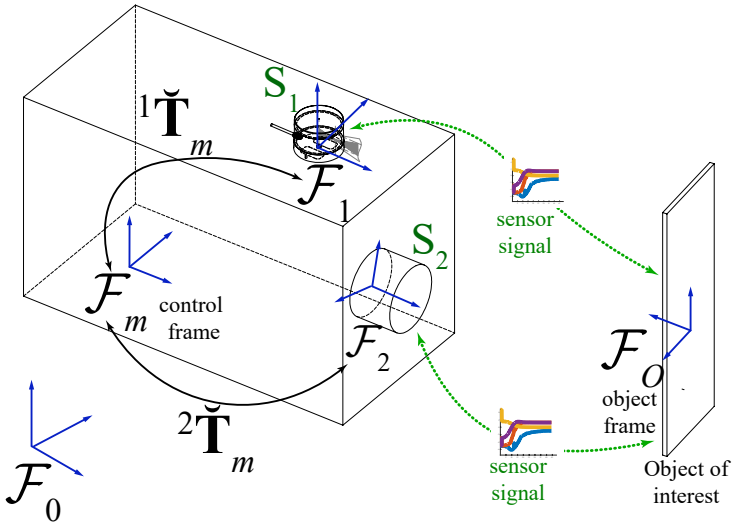


Figure 2.2: Multi-sensor model

2.2 Multi-sensor modeling

For the sake of clarity, the considered multi-sensor modeling (detailed in [73]) is recalled in this subsection.

Let us consider a robotic system equipped with k sensors (Fig. 2.2) that provide data about the environment. Each sensor S_i gives a signal (sensor feature) $\mathbf{s}_i$ of dimension $\mathbb{d}_i$ with $\sum_{i=1}^{k} \mathbb{d}_i = \mathbb{d}$.

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

$$\dot{\mathbf{s}}_i = \breve{\mathbf{L}}_i \breve{\mathbf{v}}_i = \breve{\mathbf{L}}_i {}^{i}\breve{\mathbf{T}}_m \breve{\mathbf{v}}_m \tag{2.6}$$

where $\breve{\mathbf{L}}_i$ is the interaction matrix [74] of $\mathbf{s}_i$ ($\dim(\breve{\mathbf{L}}_i) = \mathbb{d}_i \times 6$) and ${}^{i}\breve{\mathbf{T}}_m$ is the 3D screw transformation matrix that expresses the sensor twist $\breve{\mathbf{v}}_i$ (which is expressed in its

corresponding frame $\mathcal{F}_i$) with respect to the robot twist $\breve{\mathbf{v}}_m$ (expressed in the control frame $\mathcal{F}_m$).

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

$$\dot{\mathbf{s}} = \breve{\mathbf{L}}_{\mathbf{s}} \breve{\mathbf{v}}_m \tag{2.7}$$

with:

$$\breve{\mathbf{L}}_{\mathbf{s}} = \breve{\mathbf{L}} \breve{\mathbf{T}}_m \tag{2.8}$$

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

Planar world assumption

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

$${}^i\check{\mathbf{T}}_m = \begin{bmatrix} \cos({}^m\theta_i) & \sin({}^m\theta_i) & x_i \sin({}^m\theta_i) - y_i \cos({}^m\theta_i) \\ -\sin({}^m\theta_i) & \cos({}^m\theta_i) & x_i \cos({}^m\theta_i) + y_i \sin({}^m\theta_i) \\ 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

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 with origin at the point M (Fig. 2.1a) and, assuming that there is no slipping nor skidding, the robot twist $\check{\mathbf{v}}_m$ can be further reduced to (2.2). Thus it is possible to write:

$$\dot{\mathbf{s}} = \mathbf{L}_{\mathbf{s}} \, \mathbf{v}_m \tag{2.10}$$

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

It is worth noting that this planar world assumption is not a limitation of the presented approach but rather a tool that allows simplifying the different equations and matrices

taking advantage of some of the properties of the our current configuration (parallel z axes). For a more general case one could consider the following expression:

$$\dot{\mathbf{s}} = \mathbf{L_s}\,\mathbf{v}_m + \breve{\mathbf{L}}_{\mathbf{s}}^{unc}\,\breve{\mathbf{v}}_m^{unc}$$

where $\breve{\mathbf{v}}_m^{unc}$ is composed of the uncontrolled motions (extracted from $\breve{\mathbf{v}}_m$), $\breve{\mathbf{L}}_{\mathbf{s}}^{unc}$ being its corresponding interaction matrix.

2.3 Perception

2.3.1 Perception system configuration

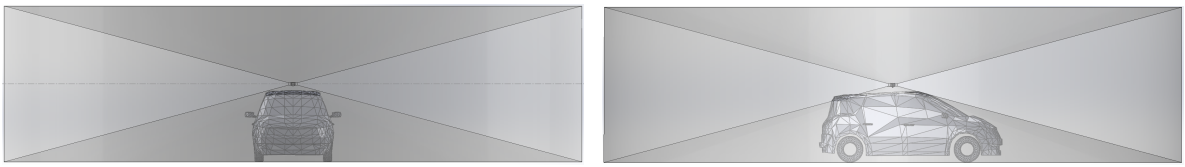
When dealing with parking applications one could expect to have obstacles all around the vehicle at, potentially, close distances. Similarly, if one was interested in detecting the markings on the ground to extract the parking spots, it would be necessary to be able to perceive all around the vehicle.

The vehicle used for experimentation (Fig. 2.1b) has been equipped with several sensors (Velodyne VLP-16, SICK LMS151, GNSS, etc.) to observe its environment, a computer to process the data and actuators that can be computer-controlled. Since,as mentioned above, our application requires exteroceptive information from all around the vehicle at potentially close distances, the VLP-16 and SICK LMS151 were the sensors chosen to work with.

Nevertheless, from the sensors' placement one can quickly tell that there non-negligible blindspots with the considered perception system. The LMS151 can only perceive on the rear of the vehicle while the perception cone of the VLP-16 doesn't allow it to detect relatively low nearby obstacles (Fig. 2.3) particularly on the sides and on the rear.

In fact, if a car is placed close to the side, the VLP-16 would not able to detect the closest portions of it as shown in Fig. 2.4. In order to ensure a proper detection, the distance between the cars would have to be increased as shown in Fig. 2.5.

To try to eliminate the previously mentioned blindspot, one could consider adding some low-cost sensors LiDARs as shown in Fig. 2.6. In the configuration proposed in Fig. 2.6a the additional sensors would be placed below the side mirrors while keeping the LMS151 for perception on the rear. The configurations shown in Figs. 2.6b and 2.6c would not require keeping the LMS151 since the perception on the rear would be already covered. The second configuration would provide the best coverage on the rear with a



(a) (b)

Figure 2.3: Perception cone of the VLP-16 placed on top of the vehicle: (a) front and (b) side views.

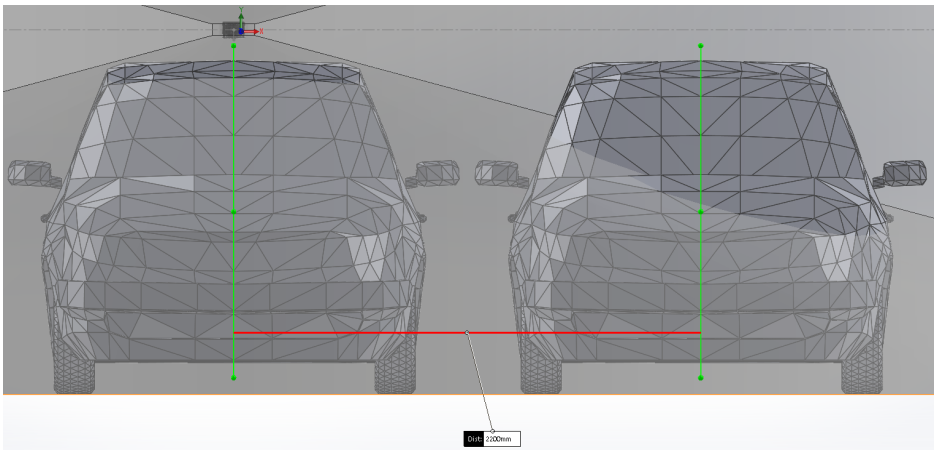


Figure 2.4: Vehicle to detect placed at a distance of 2.2 m (center to center)

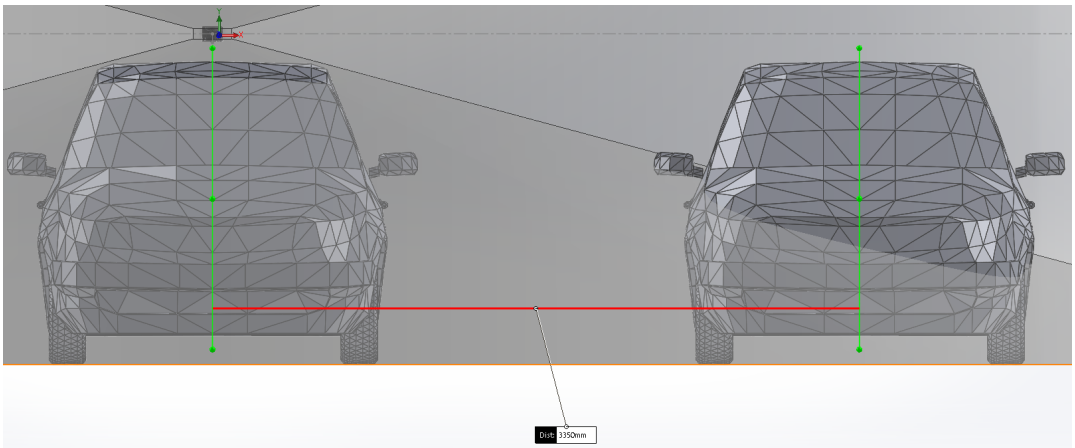
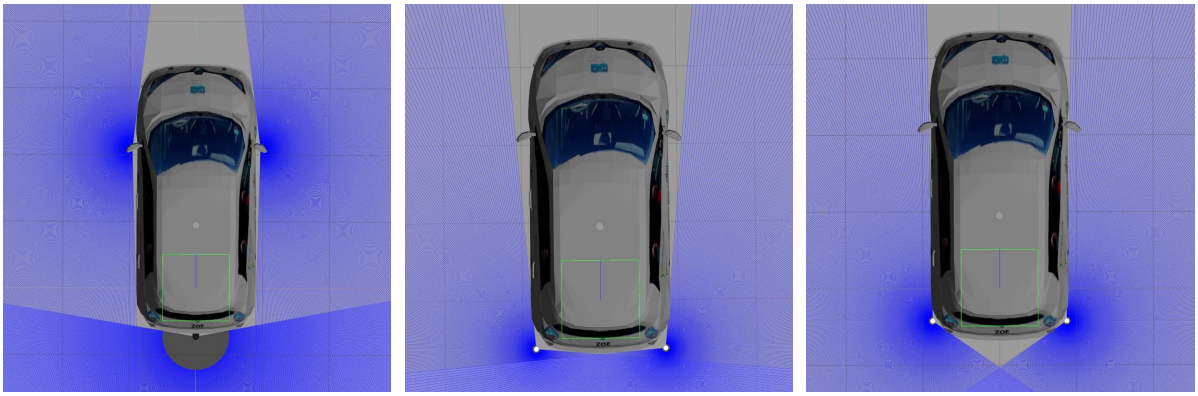


Figure 2.5: Vehicle to detect placed at a distance of 3.35 m (center to center)

very good one on the sides but having the drawback of the sensors being relatively far from the vehicle's body. In the third configuration the sensors would be very close to the body of the vehicle at the expense of a slightly worse perception on the rear, although



(a) (b) (c)

Figure 2.6: Perception capabilities of different sensor configurations (perception cone of VLP-16 omitted for the sake of clarity)

comparable to the one with a single LMS151 on the rear.

Nevertheless, due to different technical and practical reasons, no additional LiDARs where installed in the vehicle.

2.3.2 Sensory data processing

For safety reasons, the perception focuses on the detection of parked cars, which can be approximated by boxes considering that, when viewed from the top, they have a rectangular-like shape.

It should be noted that, since the focus of the work was on the control rather than on the perception problem, the algorithm presented in this section is meant to be considered as an example of how one could perform an online extraction of the required sensor features rather than the only way to do it (e.g [69]). For example, 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.

Because the two considered sensors provide information of a very similar nature, the data can be fused by simply storing the data provided by the LMS151 as a point cloud and then transforming the point cloud from LMS151's frame to the VLP-16's frame so it can be added to the point cloud provided by the latter sensor. For this, it is assumed that

the time difference between the data provided by each sensor is reasonably small, i.e. the data are sufficiently synchronized.

The complete point cloud obtained from the two sensors is first filtered with a couple of crop boxes. The first crop box keeps only the data that are close enough to the car to be relevant in a parking application and that does not represent the floor and afterwards and the second one is used to filter out the points that belong to the car's body (self-collision sensor readings). Then, an Euclidean Cluster Extraction algorithm is used to have each obstacle represented as a cluster. The orientation of each cluster is extracted by fitting a line model to the points belonging to the contour of the cluster using a RANSAC algorithm. The orientation of the bounding box is equal to the orientation of the fitted line. Then, the rotated bounding box of the cluster is computed using the previously found orientation. We have found from experience that this method gives better results than the one included (getOBB) in the Point Cloud Library (PCL) [75] which is based on the eigenvectors of the cluster.

2.3.3 Parking spot extraction

The (sufficiently large) free space between two parked vehicles is considered as a parking spot. The parked vehicles are represented by red rectangles while the parking spot is represented by a green one in Fig. 2.7.

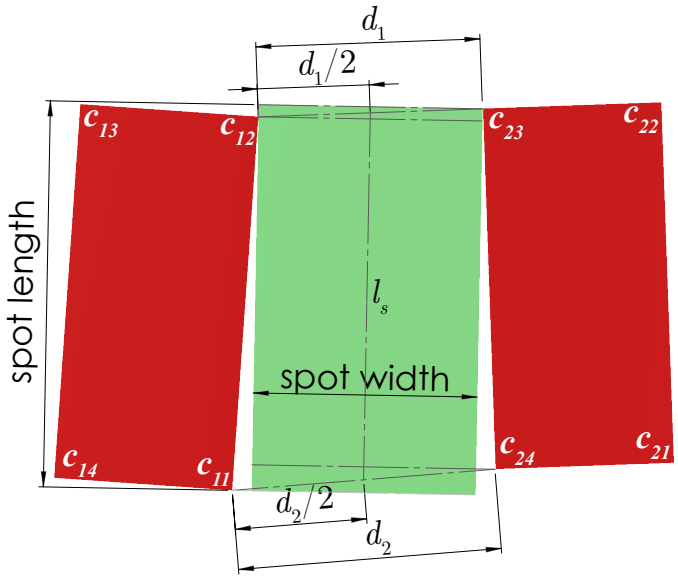
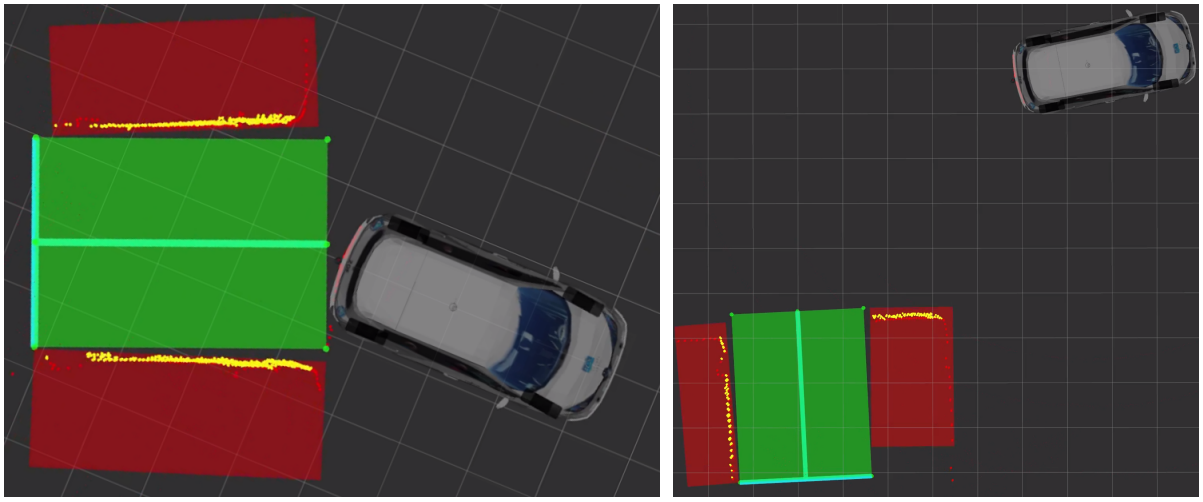


Figure 2.7: How to extract an empty parking spot (rectangular shape)



(a) (b)

Figure 2.8: Parking spot extraction from real data: (a) Already parked vehicles not perfectly aligned, (b) the parked vehicles are not completely visible

First of all, it is necessary to find the two minimum distances between the points defined by the corners of the obstacles with the constraint that the four points that define the two distances have to be different. In Fig. 2.7 the two minimum distances are d_1, defined by c_{12} and c_{23}, and d_2, defined by c_{11} and c_{24}.

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

Following this approach, the estimated parking spot adapts automatically in size and orientation to the free space between the two cars even when they are not perfectly parked (Fig. 2.8a) and/or one of them is not completely visible (Fig. 2.8b).

2.3.4 Use of virtual sensors

Assuming that the sensory data can be consistently transformed from a given frame to another, it is possible to use virtual sensors placed at will. The relevant sensors features

are thus transformed and expressed in the most convenient coordinate frame, simplifying the sensor features definitions and their interaction matrices and leading to a generic formulation.

The usefulness of virtual sensors can be exemplified as follows: if the car is parking into a perpendicular spot with a backward motion (Fig. 2.8a), the risk of collision with the obstacle on the left is the highest for the car's rear left corner, therefore it would be convenient to have a virtual sensor (S_6) placed on said corner to measure directly the distance to the line that defines the left boundary of the parking spot. Indeed, thanks to the use of virtual sensors, different physical sensors may be used as long as the necessary sensor features can be computed.

The virtual sensor placement can be seen in Fig. 2.9. The placement of S_1 and S_2 correspond, respectively, to the VLP-16 and LMS151. 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 orange rectangle whose width and length are equal to w_b and $l_{rw} + l_{fw}$ respectively. S_3 to S_8 have the same orientation as $\mathcal{F}_m$. All the virtual sensors are fed using the data extracted from the free parking spot.

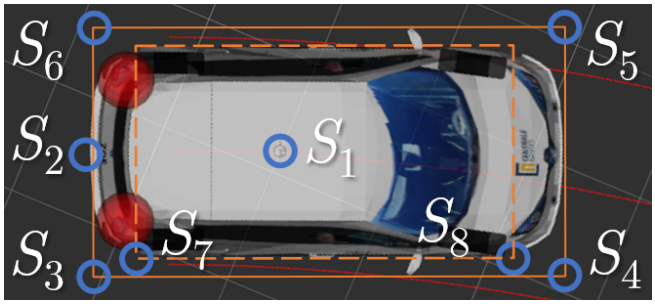


Figure 2.9: General sensors' configuration

2.3.5 Line parametrization

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

$$ {}^i p_f = [{}^i X_f; {}^i Y_f; {}^i Z_f; {}^i W_f] \tag{2.11a} $$

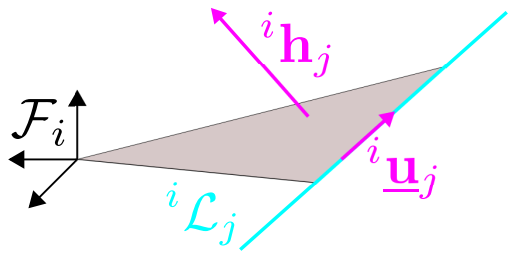


Figure 2.10: Geometric interpretation of a line's normalized Plücker coordinates

$$^ip_g = [^iX_g; ^iY_g; ^iZ_g; ^iW_g], \tag{2.11b}$$

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 [76]:

$$^i\mathcal{L}_j = [^i\underline{\mathbf{u}}_j; ^i\mathbf{h}_j] \tag{2.12}$$

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, it can be seen (Fig. 2.10) that $^i\mathbf{h}_j$ 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 [77]:

$$^i\mathbf{u}_j = {}^iW_f[^iX_g; ^iY_g; ^iZ_g] - {}^iW_g[^iX_f; ^iY_f; ^iZ_f] \tag{2.13a}$$

$$^i\mathbf{r}_j = [^iX_f; ^iY_f; ^iZ_f] \times [^iX_g; ^iY_g; ^iZ_g] \tag{2.13b}$$

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 the dissertation 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:

$$\mathbf{s}_{i,j} = \left[^i\underline{u}_{j,1}; ^i\underline{u}_{j,2}; ^ih_j\right], \tag{2.14}$$

$$\check{\mathbf{L}}_{i_{\mathcal{L}_j}} = \begin{bmatrix} 0 & 0 & {}^{i}\underline{u}_{j,2} \\ 0 & 0 & -{}^{i}\underline{u}_{j,1} \\ -{}^{i}\underline{u}_{j,2} & {}^{i}\underline{u}_{j,1} & 0 \end{bmatrix} \tag{2.15}$$

2.4 Conclusion

In this section we have established the models considered both for the vehicle and for the multi-sensor-based framework. Moreover, it was given an example of how one could extract an empty parking spot considering the perception system configuration used as well as how the use of virtual sensors can simplify the expression of the different required sensor features. Finally, the parametrization of straight lines passing through a pair of points using normalized Plücker coordinates was given.

Chapter 3

Multi-sensor-based control approach

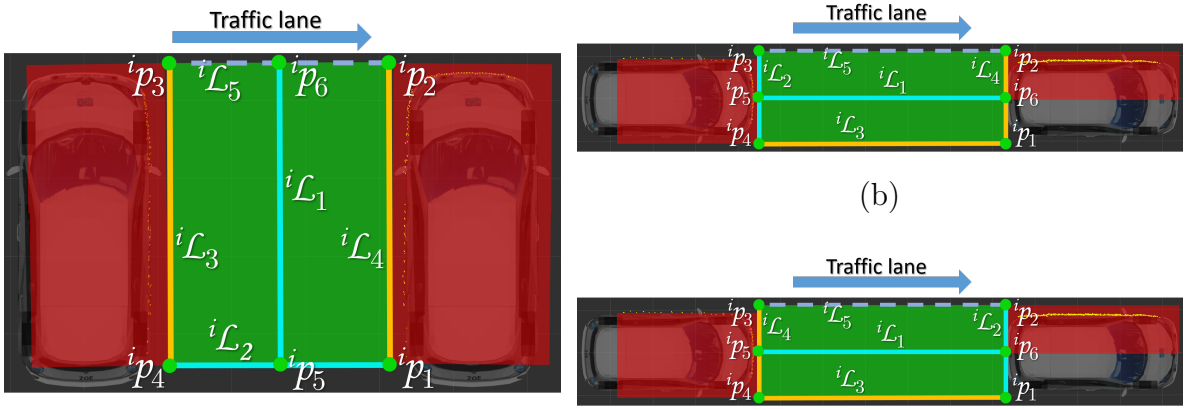
A natural goal for a human driver when parking, regardless of the type of parking spot, would be to try to make the vehicle's longitudinal axis to be collinear to the main axis of the parking spot (i.e. to be centered lateral-wise) and finish the maneuver at a certain distance from either the rear or front boundary of the parking spot (respectively for backward or forward parking maneuvers) while avoiding collision with surrounding obstacles during the whole maneuver. Following this idea, in this chapter we present an original Multi-Sensor-Based Control (MSBC) framework that allows to formalize parking operations. Using this common MSBC framework the vehicle is able to park autonomously into any type of parking (perpendicular, diagonal or parallel) with either backward or forward motions in one maneuver.

We assume that the vehicle is capable of perceiving surrounding free parking spots with its embedded sensors. Depending on the parking type, a certain set of sensor features is extracted from the sensor data and its distance to the corresponding desired features is minimized. Collision avoidance is ensured by imposing certain constraints on another set of sensor features. While it is not always possible to park in one maneuver without any path planning nor predictive control, we show that our approach performs well in a wide variety of configurations. The corresponding basins of attraction are also discussed to analyze cases where one maneuver is not enough.

3.1 Parking spots models

The presented MSBC relies on the perception of several points ${}^{i}p_a$ and lines ${}^{i}\mathcal{L}_j$ from several (virtual) sensors S_i which, as mentioned in the previous chapter, are obtained from the extracted empty parking spot.

Points ${}^{i}p_1$ to ${}^{i}p_4$ correspond to the corners of the parking spot while ${}^{i}p_5$ and ${}^{i}p_6$ are, respectively, the midpoints between $({}^{i}p_1, {}^{i}p_4)$ and $({}^{i}p_2, {}^{i}p_3)$ for the non-parallel cases (Fig. 3.1a) and between $({}^{i}p_3, {}^{i}p_4)$ and $({}^{i}p_1, {}^{i}p_2)$ for the parallel cases (Figs. 3.1b and 3.1c).



(a) (b) (c)

Figure 3.1: Parking spot models for non-parallel (b) and parallel (backward (b) and forward (c)) parking maneuvers

Table 3.1: Pair of points through which each line passes

Line	*Non-parallel*	*Parallel (backward)*	*Parallel (forward)*
${}^i\mathcal{L}_1$	$({}^ip_5, {}^ip_6)$	$({}^ip_5, {}^ip_6)$	$({}^ip_5, {}^ip_6)$
${}^i\mathcal{L}_2$	$({}^ip_1, {}^ip_4)$	$({}^ip_3, {}^ip_4)$	$({}^ip_1, {}^ip_2)$
${}^i\mathcal{L}_3$	$({}^ip_3, {}^ip_4)$	$({}^ip_1, {}^ip_4)$	$({}^ip_1, {}^ip_4)$
${}^i\mathcal{L}_4$	$({}^ip_1, {}^ip_2)$	$({}^ip_1, {}^ip_2)$	$({}^ip_3, {}^ip_4)$
${}^i\mathcal{L}_5$	$({}^ip_2, {}^ip_3)$	$({}^ip_2, {}^ip_3)$	$({}^ip_2, {}^ip_3)$

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 and front boundary of the parking spot, respectively, for backward and forward maneuvers. Lines ${}^i\mathcal{L}_3$ and ${}^i\mathcal{L}_4$ correspond to the lateral boundaries on the non-parallel cases. For the parallel case, ${}^i\mathcal{L}_3$ corresponds to the right side lateral boundary (regardless of the direction of the maneuver) while ${}^i\mathcal{L}_4$ corresponds to the front and rear boundary of the parking spot, respectively, for backward and forward maneuvers. The pair of points through which each line passes are defined in Table 3.1.

3.2 Interaction model

The interaction model considers the lines ${}^i\mathcal{L}_j$ (parametrized using normalized Plücker coordinates) and points ${}^ip_a$ introduced in Sec. 3.1.

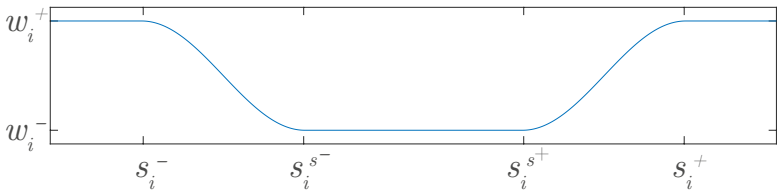


Figure 3.2: Weighting function w_i^t

3.2.1 Task

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 defined as:

$$\mathbf{s}^t = [s_1^t; \ldots; s_6^t] = [\mathbf{s}_{i_{\mathcal{L}_1}}; \mathbf{s}_{i_{\mathcal{L}_2}}] \tag{3.1}$$

with $\mathbf{s}^t$ obtained from S_1 for forward maneuvers and from S_2 for backward ones.

The corresponding interaction matrix $\check{\mathbf{L}}^t$ is computed by a 2nd order approximation [78] of the form:

$$\check{\mathbf{L}}^t = \frac{\check{\mathbf{L}}_{\mathcal{L}} + \check{\mathbf{L}}^*_{\mathcal{L}}}{2} \tag{3.2}$$

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.

In order to automatically adapt the influence of each task feature, a weighted error is considered. It is defined as follows:

$$\mathbf{e}^t_{\mathbf{W}} = \mathbf{W}\mathbf{e}^t \tag{3.3}$$

where $\mathbf{e}^t = \mathbf{s}^t - \mathbf{s}^{t^*}$ is the difference between the current sensor signal $\mathbf{s}^t$ and its desired value $\mathbf{s}^{t^*}$. The associated weighted task interaction matrix is defined as

$$\mathbf{L}^t_{\mathbf{W}} = \mathbf{W}^t\mathbf{L}^t_{\mathbf{s}} \tag{3.4}$$

with the associated weighting matrix $\mathbf{W}^t$ being defined as

$$\mathbf{W}^t = \mathrm{diag}(w_1^t, w_2^t, w_3^t, w_4^t, w_5^t, w_6^t) \tag{3.5}$$

where the values of w_3^t and w_6^t are constant while the rest of w_i^t ($w_i^t, i \in \{1, 2, 4, 5\}$) are computed using a smooth weighting function (Fig. 3.2) based on the one presented in [79].

As detailed in [80], the role of $\mathbf{W}^t$ 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.

Parking

Considering the definition of ${}^i\mathcal{L}_1$ and ${}^i\mathcal{L}_2$ (Sec. 3.2), a sensible choice would be for ${}^i\mathcal{L}_1^*$ to be collinear with the vehicle's longitudinal axis (x_m-axis) and ${}^i\mathcal{L}_2^*$ to be parallel to y_m-axis at a safe distance from either the rear or front boundary of the vehicle for, respectively, backward and forward maneuvers.

Unparking

Sensor-based control laws are usually defined so that the system reaches desired sensor features. In our case, the features are built on ${}^i\mathcal{L}_1$ and ${}^i\mathcal{L}_2$. While the desired values for those two lines are clearly defined when parking, multiple choices may be available when unparking. Indeed, it depends on how we want the car to be positioned at the end of the unparking maneuver. A sensible choice would be for ${}^i\mathcal{L}_1^*$ to be perpendicular to the vehicle's longitudinal axis (x_m-axis) and ${}^i\mathcal{L}_2^*$ to be parallel to y_m-axis, with each line being at a certain distance to the vehicle in order to be able to drive it outside of the parking spot.

3.2.2 Constraints

The set of constrained sensor features used for collision avoidance is denoted as $\mathbf{s}^c$ while its associated interaction matrix is denoted as $\check{\mathbf{L}}_{\mathbf{s}}^c$.

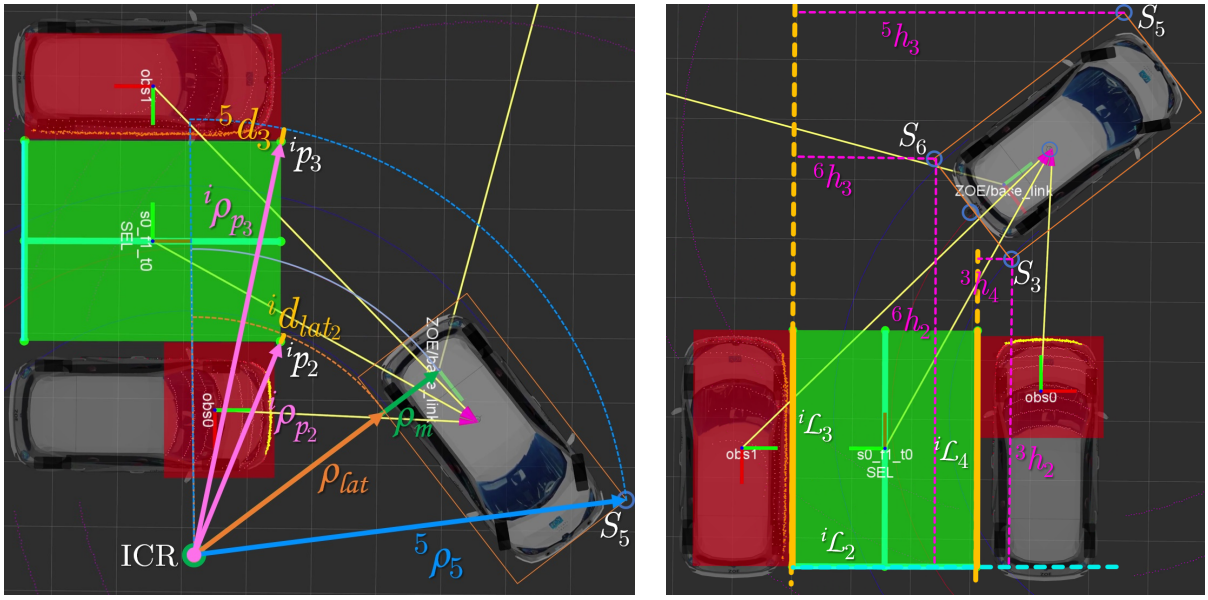
To define the constraints, three other types of features are considered in addition to ${}^i\mathcal{L}_j$: the X and Y coordinates of a given point ${}^ip_a({}^iX_a, {}^iY_a)$ and the differences of radii (shown in Fig. 3.3a):

$$ {}^id_{lat_a} = {}^i\rho_{p_a} - \rho_{lat} \tag{3.6} $$

$$ {}^id_a = {}^i\rho_i - {}^i\rho_{p_a} \tag{3.7} $$

where:

$$ {}^i\rho_{p_a} = \sqrt{({}^iX_a + x_i)^2 + ({}^iY_a + y_i - \rho_m)^2} \tag{3.8} $$



(a) Differences of radii: ${}^{i}d_{lat_2}$ for ${}^{i}p_2$ from sensor S_i and ${}^{5}d_3$ for ${}^{i}p_3$ from sensor S_5. All the radii define concentric arcs with center at ICR

(b) Illustration of the physical meaning of different ${}^{i}h_j$ features

Figure 3.3: Example of what different features used as constraints for collision avoidance mean physically.

$$ {}^{i}\rho_i = \sqrt{{x_i}^2 + (\rho_m - y_i)^2} \tag{3.9} $$

$$ \rho_{lat} = |\rho_m| - \frac{w_{ve}}{2} \tag{3.10} $$

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

$$ \check{\mathbf{L}}_{i_{X_a}} = \begin{bmatrix} -1 & 0 & {}^{i}Y_a \end{bmatrix} \tag{3.11} $$

$$ \check{\mathbf{L}}_{i_{Y_a}} = \begin{bmatrix} 0 & -1 & -{}^{i}X_a \end{bmatrix} \tag{3.12} $$

while the interaction matrices for both (3.6) and (3.7) is defined as:

$$ \check{\mathbf{L}}_{i_d} = \begin{bmatrix} 0 & \frac{{}^{i}\varrho_y}{{}^{i}\rho_{p_a}^2} & \frac{{}^{i}X_a\,{}^{i}\varrho_y}{{}^{i}\rho_{p_a}^2} \end{bmatrix} \tag{3.13} $$

with ${}^{i}\varrho_y = -|{}^{i}Y_a + y_i - \rho_m|$. The interaction matrices associated to the rest of the features

used as constraints can be deduced from the third row of (2.15).

Unlike $\mathbf{s}^t$, due to the nature of each type of parking maneuver, the exact definition of the set of sensor features $\mathbf{s}^c$ required for collision avoidance is different for each parking/unparking case although, in general, as shown in Fig. 3.3b constraints on ${}^i h_j$ are used to avoid going outside the boundaries of the parking spot when S_i is inside it, constraints on ${}^i X_a$ are used to stop the vehicle before colliding with ${}^i p_a$ (typically the corners that define ${}^i\mathcal{L}_5$) if said point cannot be avoided, constraints on ${}^i d_{lat_a}$ and ${}^i d_a$ (Fig. 3.3a) are used to keep the vehicle's path away from the obstacle and, constraints on ${}^i Y_a$ are used to keep a certain minimum lateral distance to point ${}^i p_a$. The exact definition of the set of constrained sensor features $\mathbf{s}^c$ is now given case by case.

Parking

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, a collision free maneuver can be ensured by constraining sensor features observed by the sensors placed at the rear corners of the vehicle. As such, $\mathbf{s}^c$ is defined as:

$$\mathbf{s}^c = [s_1^c; \ldots; s_{10}^c] = [\mathbf{s}_3; \mathbf{s}_6] \tag{3.14}$$

with

$$\mathbf{s}_3 = [{}^3 h_2; {}^3 h_4; {}^3 X_2; {}^3 Y_2; {}^3 d_{lat_2}] \tag{3.15a}$$

$$\mathbf{s}_6 = [{}^6 h_2; {}^6 h_3; {}^6 X_3; {}^6 Y_3; {}^6 d_3]. \tag{3.15b}$$

Forward non-parallel case Analogously to the previous case, this time a collision free maneuver can be ensured by constraining sensor features observed by the sensors placed at the front corners of the vehicle. As such, $\mathbf{s}^c$ is defined as:

$$\mathbf{s}^c = [s_1^c; \ldots; s_8^c] = [\mathbf{s}_4; \mathbf{s}_5] \tag{3.16}$$

with

$$\mathbf{s}_4 = [{}^4 h_2; {}^4 h_3; {}^4 X_3; {}^4 Y_3] \tag{3.17a}$$

$$\mathbf{s}_5 = [{}^5h_2; {}^5h_4; {}^5Y_2; {}^5d_2]. \tag{3.17b}$$

Backward parallel case Even if for this type of maneuvers the rear side of the vehicle has to enter first into the parking spot, there is risk of collision almost all-around the vehicle thus the constrained sensor features should be observed by most of the sensors although giving special attention to the right side of the vehicle. 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, $\mathbf{s}^c$ is defined as follows:

$$\mathbf{s}^c = [s_1^c; \ldots; s_{10}^c] = [\mathbf{s}_3; \mathbf{s}_4; \mathbf{s}_6; \mathbf{s}_7; \mathbf{s}_8] \tag{3.18}$$

with

$$\mathbf{s}_3 = [{}^3h_2; {}^3X_2; {}^3Y_2; {}^3d_{lat_2}] \tag{3.19a}$$

$$\mathbf{s}_4 = [{}^4Y_2; {}^4d_2; {}^4Y_4] \tag{3.19b}$$

$$\mathbf{s}_6 = {}^6h_2 \tag{3.19c}$$

$$\mathbf{s}_7 = {}^7h_3 \tag{3.19d}$$

$$\mathbf{s}_8 = {}^8h_3. \tag{3.19e}$$

Forward parallel case Similar to the backward parallel case, there is risk of collision almost all-around the vehicle but mostly on its right side and, additionally, the boundary ${}^i\mathcal{L}_3$ would as well often coincide with a curb. Therefore, $\mathbf{s}^c$ is defined as follows:

$$\mathbf{s}^c = [s_1^c; \ldots; s_7^c] = [\mathbf{s}_3; \mathbf{s}_4; \mathbf{s}_5; \mathbf{s}_7; \mathbf{s}_8] \tag{3.20}$$

with

$$\mathbf{s}_3 = {}^3d_{lat_3} \tag{3.21a}$$

$$\mathbf{s}_4 = [{}^4h_2; {}^4X_3; {}^4Y_3] \tag{3.21b}$$

$$\mathbf{s}_5 = {}^5h_2 \tag{3.21c}$$

$$\mathbf{s}_7 = {}^7h_3 \tag{3.21d}$$

$$\mathbf{s}_8 = {}^8h_3. \tag{3.21e}$$

Unparking

Forward non-parallel case This type of unparking maneuver would have to be performed when the vehicle was parked with a backward non-parallel maneuver. As such,

the constrained sensor features $\mathbf{s}^c$ are mostly the same:

$$\mathbf{s}^c = [s_1^c; \ldots; s_7^c] = [\mathbf{s}_3; \mathbf{s}_6] \tag{3.22}$$

with

$$\mathbf{s}_3 = [{}^3h_2; {}^3Y_2; {}^3d_{lat_2}] \tag{3.23a}$$

$$\mathbf{s}_6 = [{}^6h_2; {}^6h_3; {}^6Y_3; {}^6d_{lat_3}]. \tag{3.23b}$$

Backward non-parallel case Analogously to the previous case, this type of unparking maneuver would have to be performed when the vehicle was parked with a forward non-parallel maneuver. As such, the constrained sensor features $\mathbf{s}^c$ are mostly the same as its parking counterpart:

$$\mathbf{s}^c = [s_1^c; \ldots; s_6^c] = [\mathbf{s}_4; \mathbf{s}_5] \tag{3.24}$$

with

$$\mathbf{s}_3 = [{}^3h_2; {}^3h_3; {}^3Y_3] \tag{3.25a}$$

$$\mathbf{s}_6 = [{}^6h_2; {}^6h_4; {}^6Y_2]. \tag{3.25b}$$

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 ${}^3d_{lat_3}$ 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 the appendix.

3.3 Control

From the kinematic model of the task and constrained sensor features, a purely reactive control law is performed. The core of the control law is formulated in Quadratic

Programming (QP) form [81] with only inequality constraints:

$$\begin{aligned} \mathbf{v}_m = (v_{x_m}; \dot{\theta}_m) = \quad & \text{argmin} ||\mathbf{L}_{\mathbf{W}}^t \mathbf{v}_m + \lambda \mathbf{e}_{\mathbf{W}}^t||^2 \\ & \text{s.t. } \mathbf{A}\mathbf{v}_m \leq \mathbf{b} \end{aligned} \tag{3.26}$$

with:

$$\mathbf{A} = [\mathbf{L}_{\mathbf{s}}^c; -\mathbf{L}_{\mathbf{s}}^c] \tag{3.27}$$

$$\mathbf{b} = [\alpha(\mathbf{s}^{c^+} - \mathbf{s}^c); -\alpha(\mathbf{s}^{c^-} - \mathbf{s}^c)] \tag{3.28}$$

where α is a constant gain, λ is the control gain and $[\mathbf{s}^{c^-}, \mathbf{s}^{c^+}]$ is the interval in which $\mathbf{s}^c$ should remain. Since the constraints are used for collision avoidance, only one side of the interval $[s_i^{c^-}, s_i^{c^+}]$ is considered for each feature.

In practice, the control input of the robotized vehicle is given by (2.4). Since the mapping (2.3) is nonlinear, a general purpose nonlinear solver is used to compute (3.26) with $\mathbf{v}_r$ as optimization variables. Considering ϕ as optimization variable instead of $\dot{\theta}_m$ also allows to take into account the steering wheel joint limits $\phi_{\max}$ by bounding the output with an additional constraint:

$$|\phi| < \phi_{\max} \tag{3.29}$$

Similarly, the longitudinal velocity is bounded in the solver:

$$|v_{x_m}| < v_{\max} \tag{3.30}$$

where $v_{\max}$ is an adaptive saturation value imposing a deceleration profile based on the velocity profile shown in [34] as the vehicle approaches the final pose. Furthermore, to avoid large changes in the control signals at the current iteration n that may cause uncomfortable sensations for the passengers or surrounding witnesses and, to consider to some extent the dynamic limitations of the vehicle, the control signals are saturated as well by some increments with respect to the previous control signals (at iteration $n-1$) as shown below:

$$(v_{n-1} - \Delta_{dec}) \leq v_n \leq (v_{n-1} + \Delta_{acc}) \tag{3.31a}$$

$$(\phi_{n-1} - \Delta_{\phi}) \leq \phi_n \leq (\phi_{n-1} + \Delta_{\phi}). \tag{3.31b}$$

As we can see, the sensor-based control is defined in a generic way up to (3.26), then the actual vehicle model is considered in order to add mechanical (steering limits) and practical (control continuity) constraints. The main advantage of a reactive control in our case is that it adapts to changes in the environment and to non-perfect response of the vehicle. Indeed, the low-level control may not be able to follow perfectly the $\mathbf{v}_r$ setpoint.

It is known that sensor-based control laws are Local Asymptotically Stable (LAS) in a neighborhood of $\mathbf{e} = 0$ when $\mathbf{W}^t\mathbf{L}_\mathbf{s}^t$ and $\mathbf{W}^t\widehat{\mathbf{L}_\mathbf{s}^t}$ are full rank and $\mathbf{L}_\mathbf{s}^t$ is sufficiently well estimated [73], which is the case for our system . Furthermore, the LAS property is maintained even when dealing with more that one task if the optimum is inside the bounds [82]. Additionally, it has been proved that inequality constraints (which, when active, may be seen as additional tasks) do not imply an unstable behavior [83]. If the objective is compatible with the inequality constraints (which is our case) then the control is still LAS. On the opposite, if the control objective violates some inequalities then the system is known to converge to the boundary of the inequalities.

All the stability properties mentioned in the previous paragraph consider holonomic systems. Since car-like vehicles are subject to non-holonomic constraints and do not satisfy Brockett's necessary condition for being stabilized via smooth time-invariant feedback [84], our goal is not to achieve asymptotic stability but rather practical stability, i.e. to bring the system to an arbitrarily small neighborhood of $\mathbf{e} = 0$. In order to do so, our control law attempts to minimize at best (in the least squares sense) both the weighted error of the task $\mathbf{e}_\mathbf{W}^t$ and the vehicle's twist $\mathbf{v}_m$ with the understanding that, due to the non-holonomic constraints, the final error may not be equal to zero even when the vehicle manages to get inside the parking spot. We illustrate the properties of our approach in the following sections.

3.4 Simulation results

For the results shown in this section, the parameters in Table 3.2 are considered. The value of $\phi_{\max}$ corresponds to the maximum steering angle of the real vehicle while the rest of the parameters were determined by empirical testing. Simulations are carried in MATLAB with a sampling time T_s= 0.1. MATLAB implementation uses the *fmincon* solver with a Sequential Quadratic Programming (SQP) algorithm.

Table 3.2: Control-related vehicle parameters

Parameteres	*Notation*	*Value*
Maximum steering angle	$\phi_{\max}$	$0.5236\,\text{rad}$
Maximum longitudinal velocity	$v_{\max}$	$\leq 0.556\,\text{m/s}$
Maximum acceleration increment	Δ_{acc}	$\text{sign}(v_{n-1})\ 0.2\,\text{m/s}\ T_s$
Maximum deceleration increment	Δ_{dec}	$\text{sign}(v_{n-1})\ 2.5\,\text{m/s}\ T_s$
Maximum ϕ increment	Δ_{ϕ}	$2^\circ\ T_s$

3.4.1 Parking

Individual cases - MATLAB simulations

To illustrate the performance of the proposed approach, several parking maneuvers are shown in this subsection. The initial position is marked with a colored (yellow, green and dark blue in the shown cases) square whose color depends on the final value of $||\mathbf{e}^t||$. An initial position marked in yellow (Fig. 3.4a) corresponds to a successful parking maneuver with a small final error ($||\mathbf{e}^t||$) while an initial position marked in dark blue (Fig. 3.5b) corresponds to an unsuccessful maneuver. The initial position of a successful maneuver but with a relatively large final error (Fig. 3.5a) is marked with a color lying in between dark blue and yellow.

Backward perpendicular cases are shown Figs. 3.4a, 3.5a and 3.5b. For these three maneuvers, the initial position is the same (8 m, 6.5 m). The initial orientation for each maneuver is, respectively 5°, 25° and 30°. In addition to the parking maneuver evolution, the evolution of $\mathbf{v}_r$, $\mathbf{e}^t$ and $\mathbf{s}^c$ can be seen in Fig. 3.4. The control signals have a generally smooth evolution (Fig. 3.4b) thanks to the considered bounds and saturation values. The (active) constraints imposed on $\mathbf{s}^c$ are satisfied at each time instant (Fig. 3.4d) ensuring a collision-free maneuver. The final task error $\mathbf{e}^t$ is very close to zero although, as it can bee seen in Fig. 3.4c, its evolution shows slight oscillations on certain errors e_i^t during the first 10 seconds of the maneuver due to the use of an interaction matrix of the form (3.2) [78]. These oscillations, instead of being a problematic behavior, are useful to maneuver successfully into the parking spot in one move in certain situations where an otherwise purely exponential decay of $\mathbf{e}^t$ without any oscillation would not be able to. As it is shown in Sec. 3.4.1, the aforementioned oscillations do not jeopardize the convergence of the proposed approach.

As it can be seen, the success or lack thereof of the parking maneuver depends on whether or not the car has enough space to maneuver (without changing the direction of

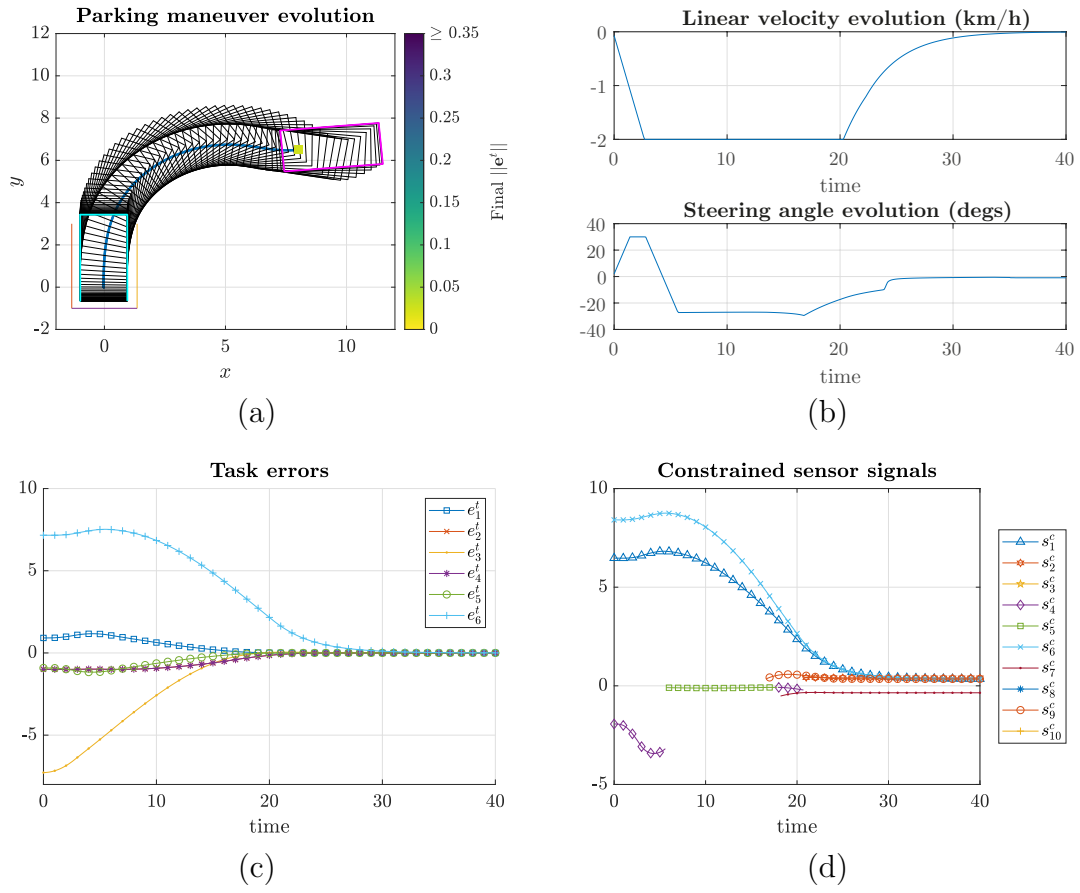


Figure 3.4: Backward ⊥ parking maneuver signals: (a) performed maneuver ($\theta_{T=0} = 5$), (b) control signals, (c) task error signal, (d) constrained sensor signals

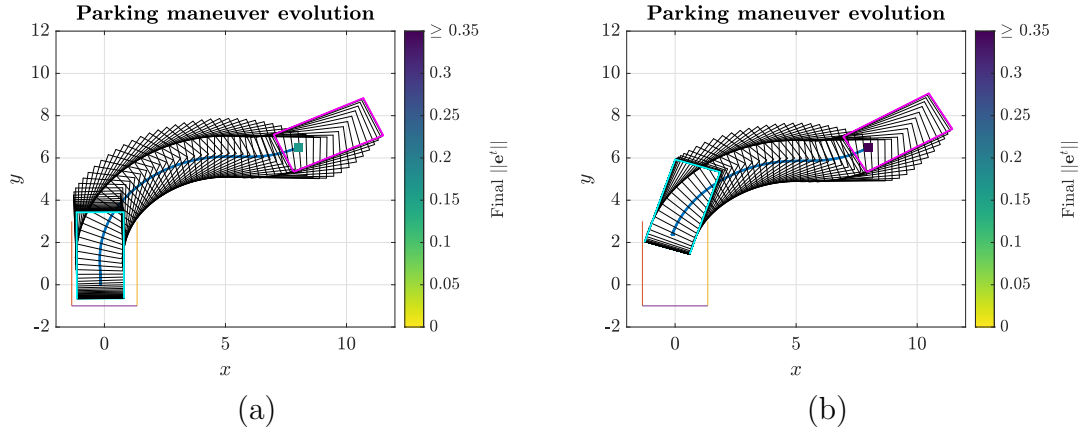


Figure 3.5: Backward ⊥ parking maneuvers. (a) $\theta_{T=0} = 25°$. (b) $\theta_{T=0} = 30°$

longitudinal motion) from its initial configuration to the desired one while satisfying the considered (control continuity, collision avoidance, etc) constraints.

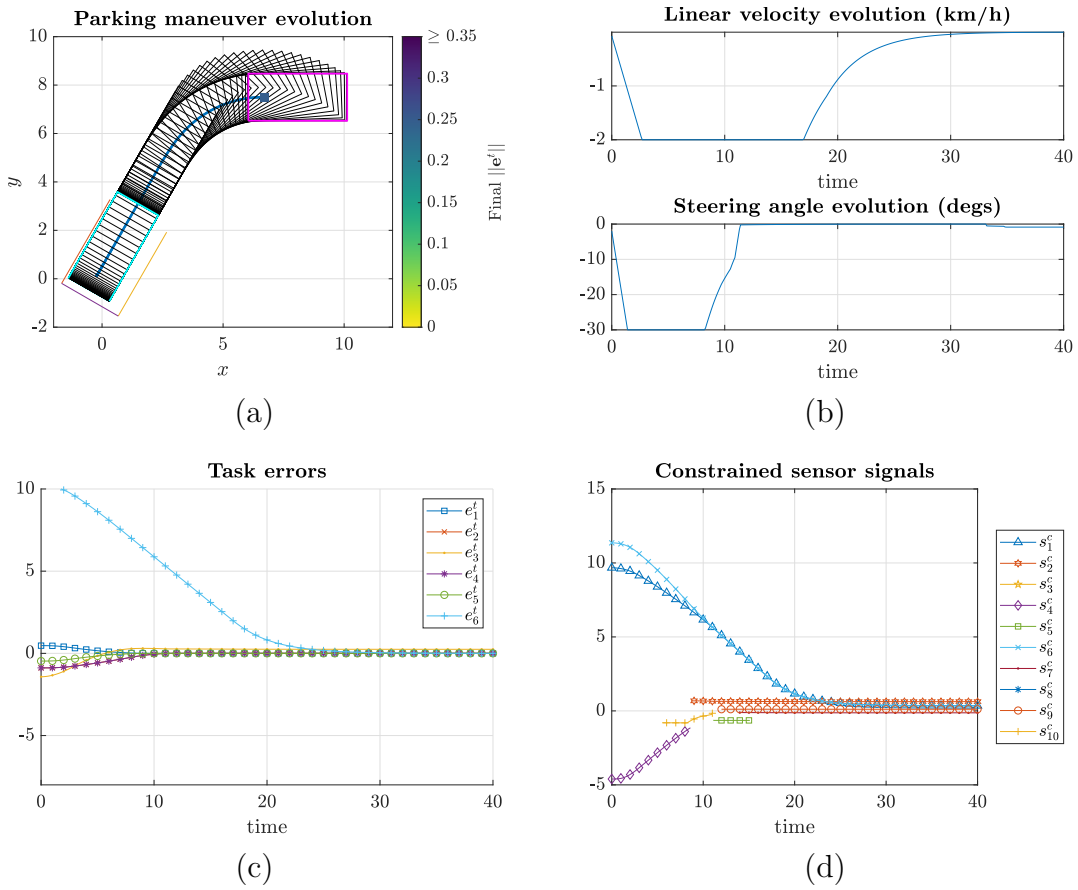
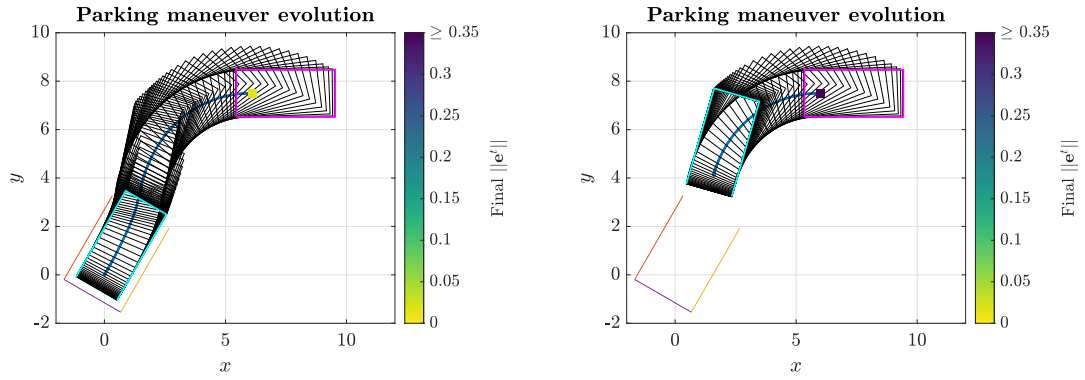


Figure 3.6: Backward diagonal parking maneuver signals: (a) performed maneuver (initial pose = (6.7 m, 7.5 m, 0°)), (b) control signals, (c) task error signal, (d) constrained sensor signals

Three backward diagonal cases are shown in Figs. 3.6a, 3.7a and 3.7b with the same initial orientation 0° while the initial position is, respectively, (6.7 m, 7.5 m), (6.1 m, 7.5 m) and (6 m, 7.5 m). It can be seen in Fig. 3.6a that the car is able to park successfully, satisfying the constraints at each time instant (Fig. 3.6d), with a simple maneuver but ends relatively far from the main axis of the parking spot (Fig. 3.6c) while in Fig. 3.7a the car has to perform a more complex maneuver in order to satisfy the constraints with the side effect of ending closer to parking spot's main axis. In Fig. 3.7b, the car gets trapped in local minima.



(a) (b)

Figure 3.7: Backward diagonal parking maneuvers. (a) Initial pose = (6.1 m, 7.5 m, 0°). (b) Initial pose = (6 m, 7.5 m, 0°)

As most experienced drivers would expect, parking with forward motions require more space to maneuver. Fig. 3.8a shows a case where the car is able to park successfully into a perpendicular spot with a forward motion, reaching a very close to zero final task error $\mathbf{e}^t$, with generally smooth control signals (Fig. 3.8b) while satisfying the constraints at every time instant (Fig 3.8d). Similarly to the backward case (Fig. 3.4), the car initially steers away from the parking spot, giving itself more room to maneuver.

If initial position of the vehicle was to be shifted downwards as in Fig. 3.9a, one would notice that even if the car manages to get inside the parking spot it does not manage to get properly aligned. In order to be able to end properly aligned, the vehicle's initial position would have to be shifted considerably to the left as shown in Fig. 3.9b so that the vehicle would have enough room to steer away from the parking spot at the beginning to later be able to reach the desired parked pose.

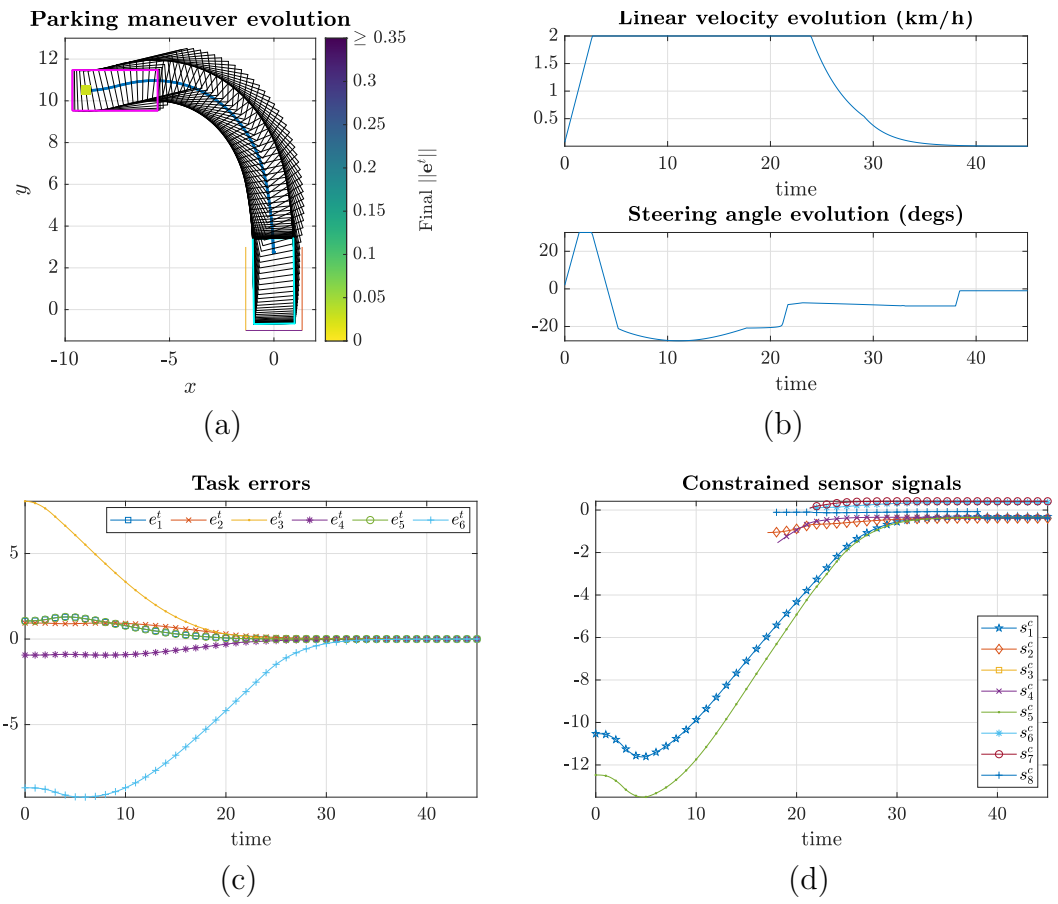


Figure 3.8: Forward perpendicular parking maneuver signals: (a) performed maneuver (initial pose = (−9 m, 10.5 m, 0°)), (b) control signals, (c) task error signal, (d) constrained sensor signals

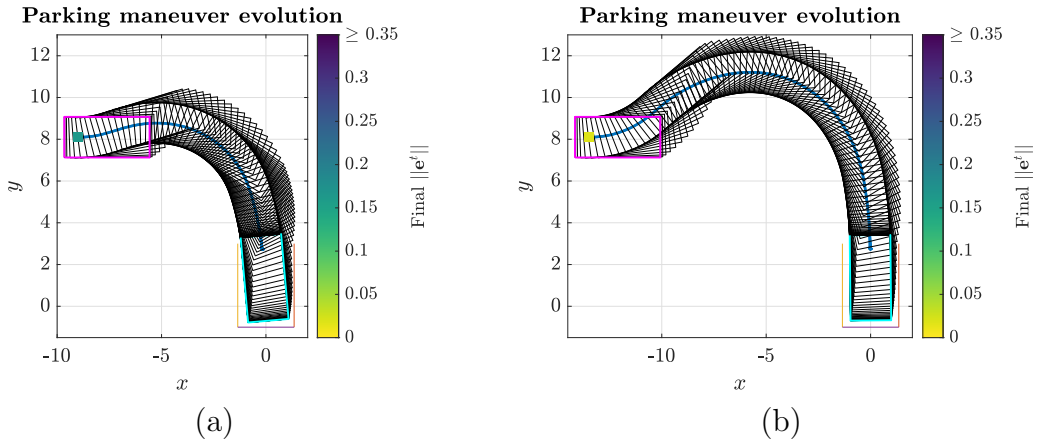


Figure 3.9: Forward perpendicular parking maneuvers. (a) Initial pose = (−9 m, 8.1 m, 0°). (b) Initial pose = (−13.5 m, 8.1 m, 0°)

As most drivers would expect, when dealing with diagonal parking spots, the vehicle needs less space to maneuver than with perpendicular spots. In Fig. 3.10a a case where the vehicles parks successfully into a diagonal parking spot with a forward motion satisfying the collision avoidance constraints at every time instant (Fig. 3.10d), reaching a very close to zero final task error $\mathbf{e}^t$. In this case, the controller does not steer the vehicle away form the parking spot (Fig. 3.10b) since there is no need nor room to do so.

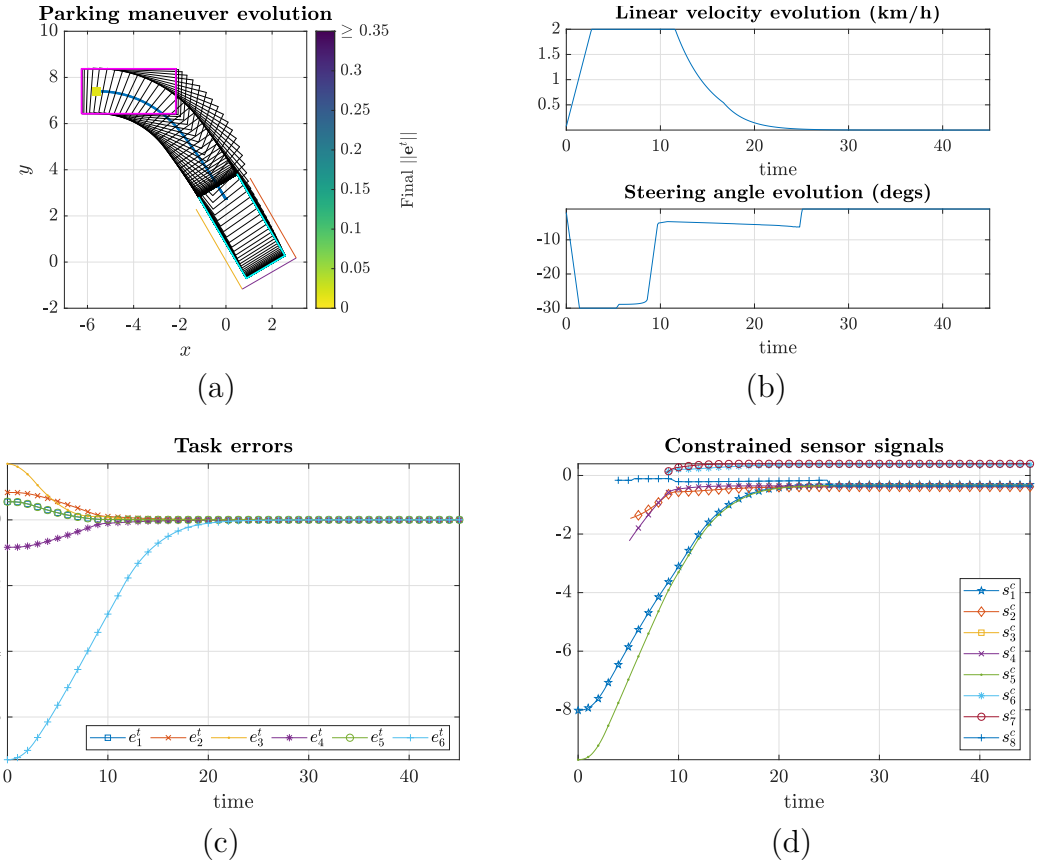
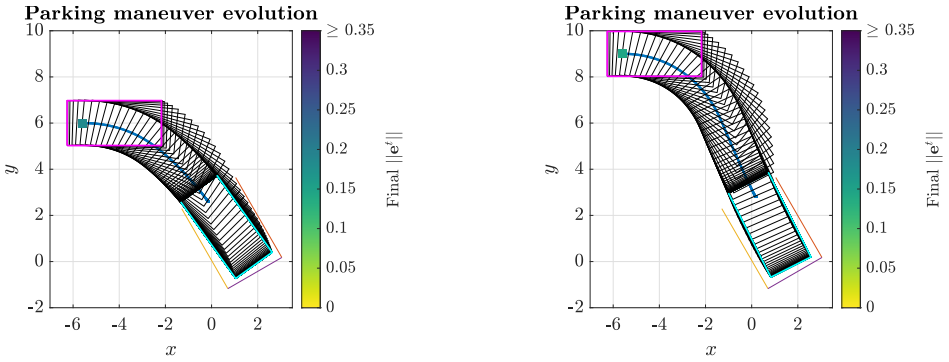


Figure 3.10: Forward diagonal parking maneuver signals: (a) performed maneuver (initial pose = $(-5.6\,\text{m},\ 7.4\,\text{m},\ 0°)$), (b) control signals, (c) task error signal, (d) constrained sensor signals

If the initial position was to be shifted downwards as in Fig. 3.11a, one would notice that, due to the collision avoidance constraints, the car would not able to align itself correctly with the parking spot even if it manages to get inside it. If instead of shifting the initial position downwards it would be shifted upwards as in Fig. 3.11b, the car would have to perform a more complex maneuver in order to get inside the parking spot although in the end not managing to aligned itself properly.



(a) (b)

Figure 3.11: Forward diagonal parking maneuvers. (a) Initial pose = (−5.6 m, 6 m. 0°). (b) Initial pose = (−5.6 m, 9 m, 0°)

A successful parking maneuver into a parallel spot with a collision-free (Fig. 3.12d) backward motion can be seen in Fig. 3.12a. The final task error $\mathbf{e}^t$ is very close to zero and the control signals are generally smooth (Fig. 3.12b). It can be noticed that in this case, unlike for non-parallel parking maneuvers, the controller does not steer the vehicle away from the parking spot.

When considering an initial position for the vehicle considerably farther (lateral-wise) from the parking spot as in Fig. 3.12a, the car still manages to park successfully. On the contrary, if the vehicle's initial position is closer (longitudinal-wise) to the parking spot, as in Fig. 3.13b the car would not have enough room to aligned itself properly with the parking spot even if it manages to get inside it.

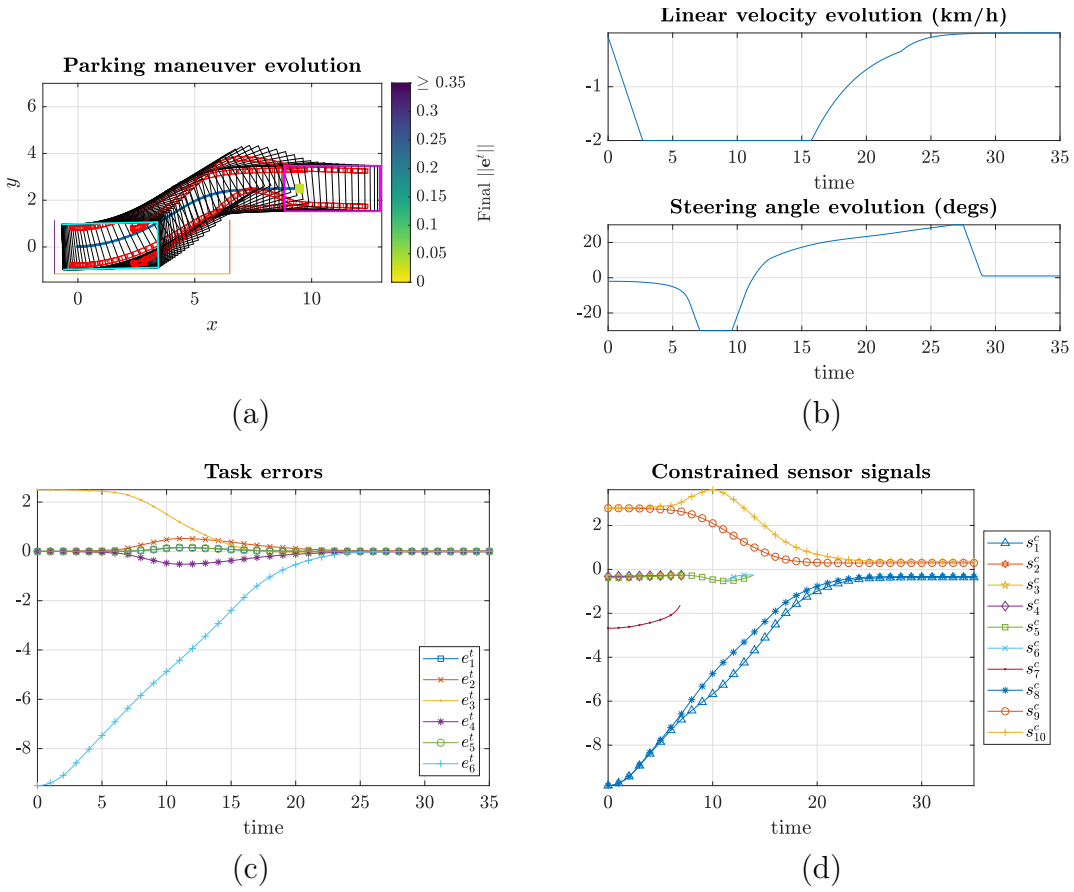


Figure 3.12: Backward ‖ parking maneuver signals: (a) performed maneuver (initial pose = (9.5 m, 2.5 m, 0°)), (b) control signals, (c) task error signal, (d) constrained sensor signals

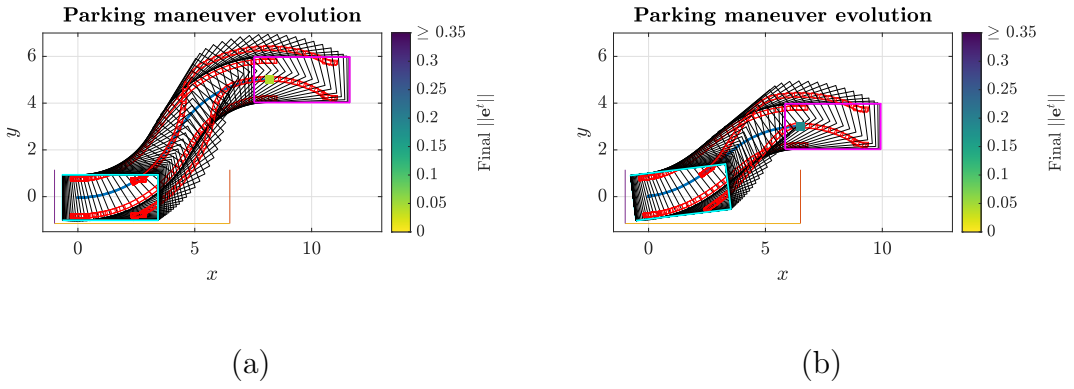


Figure 3.13: Backward ‖ parking maneuvers. (a) Initial pose = (8.2 m, 5 m, 0°). (b) Initial pose = (6.5 m, 3 m, 0°)

Parking into parallel spots with forward motions is the most complicated case and the least advisable. In fact, in order to be able to park in one maneuver the parking spot would have to be considerably longer than with backward motions. Nevertheless, a successful parking maneuver into a parallel spot with a collision-free (Fig. 3.12d) forward motion can be seen in Fig. 3.12a. The final task error $\mathbf{e}^t$ is very close to zero and the control signals are generally smooth (Fig. 3.12b).

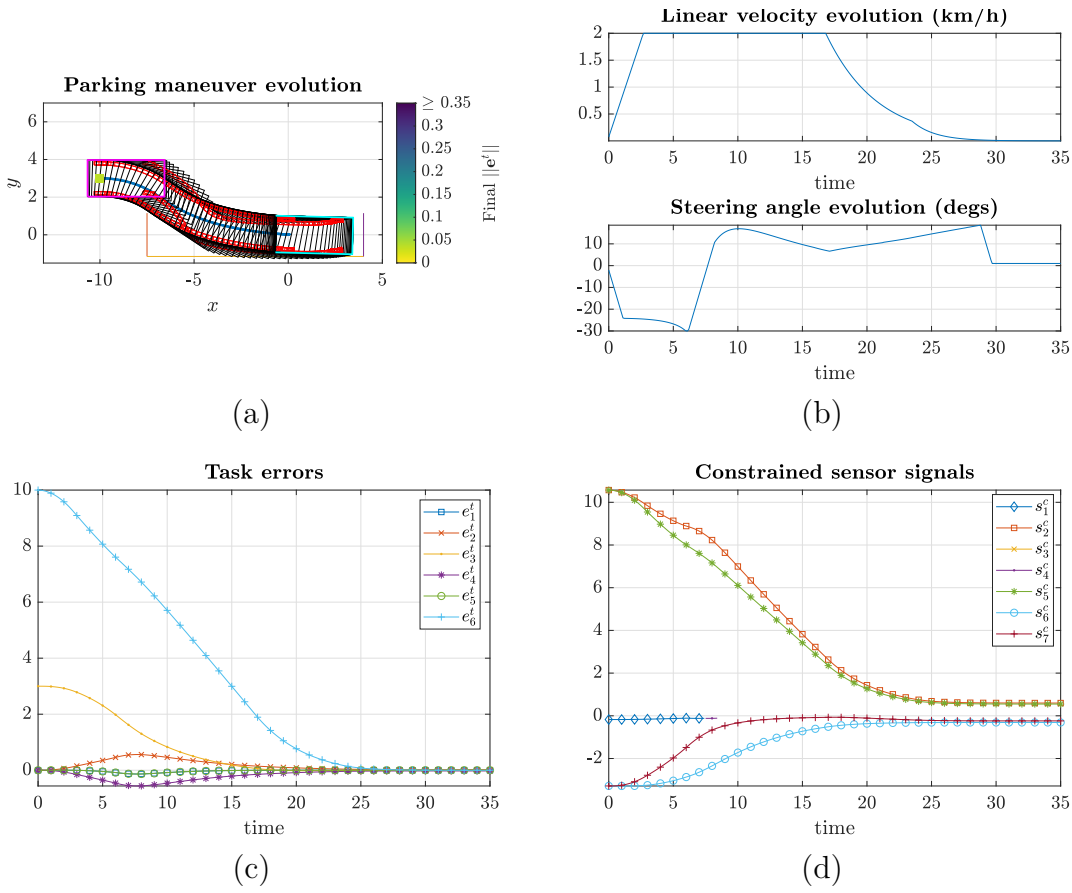
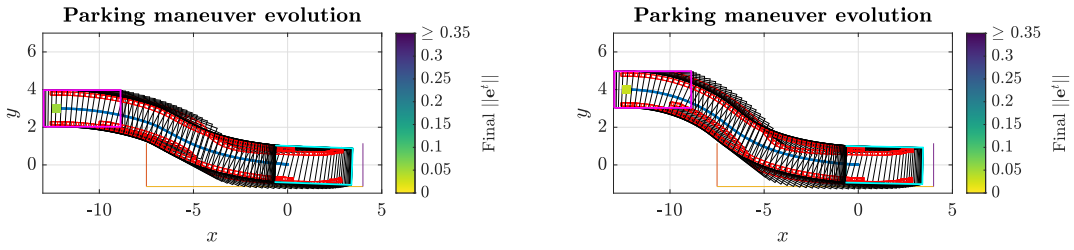


Figure 3.14: Forward ‖ parking maneuver signals: (a) performed maneuver (initial pose = (−10 m, 3 m, 0°)), (b) control signals, (c) task error signal, (d) constrained sensor signals

If one would shift the car's initial position to the left as in Fig. 3.15a, the final orientation of the vehicle would be slightly worse since by the moment the car would have to start steering to the right in order to enter to the parking spot, it would have already reached its maximum allowed longitudinal velocity thus not being able to turn as sharply when compared to Fig. 3.14a. In order to improve the final orientation of the vehicle, its initial position would have to be shifted upwards (Fig. 3.15b) so that the car would be

able to enter the parking spot with a more pronounced angle (similar to Fig .3.14a).



(a) (b)

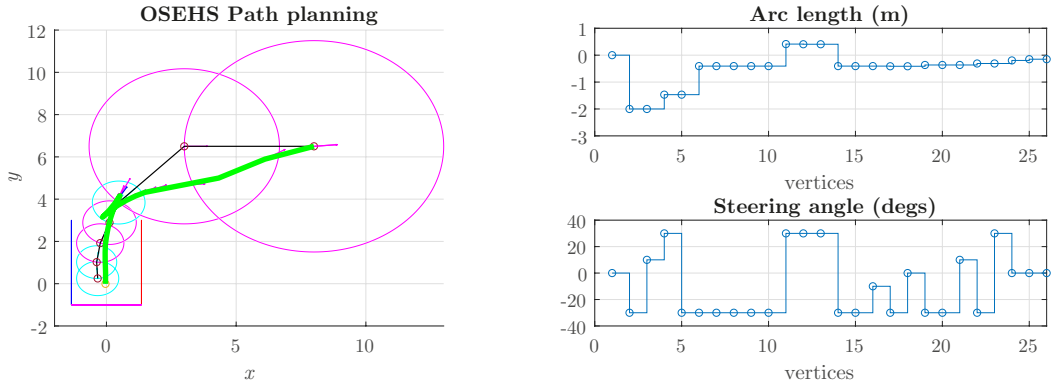
Figure 3.15: Forward $\|$ parking maneuvers. (a) Initial pose $= (-12.3\,\text{m},\ 3\,\text{m},\ 0°)$. (b) Initial pose $= (-12.3\,\text{m},\ 4\,\text{m},\ 0°)$

Comparison against a state of the art path planning approach

To put in perspective the performance of the proposed MSBC approach, it is compared against a state of the art path planning Orientation-Aware Space Exploration Guided Heuristic Search approach [39], whose authors claim that is fast enough (<100ms) to make online re-planning feasible during the execution of the maneuver. The OSEHS approach was chosen to compare against because it is generic enough to be able to deal with different types of parking scenarios and does not necessarily require to have knowledge about the free and occupied space of the whole environment beforehand.

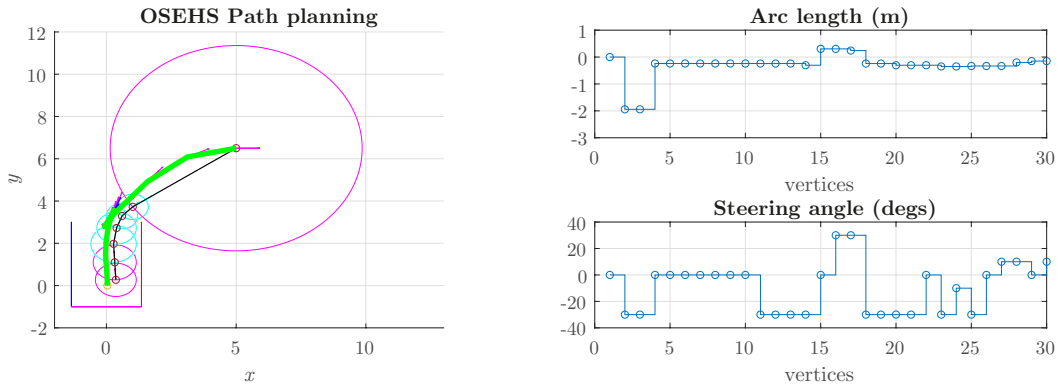
The results obtained from the implementation of the OSEHS algorithm for two different initial poses are presented in Figs. 3.16 and 3.17. Figs. 3.16a and 3.17a show the arc length that the vehicle has to travel between the current configuration and the next one with the sign of the arc length defining the direction of longitudinal motion: negative for backwards motions and positive for the forward case. Figs. 3.16b and 3.17b show the steering angle required to reach the next configuration from the current one.

Considering the two cases with a common initial pose (Figs. 3.4 and 3.16), it can be clearly seen that the OSEHS algorithm requires more maneuvers to park (3 vs 1) than our MSBC approach. Similarly, three maneuvers are required for the case shown in Fig. 3.17 while, as shown in Sec. 3.4.1, the MSBC approach is able to park in one maneuver. Furthermore, due to the existing trade-off between completeness and efficiency



(a) (b)

Figure 3.16: OSEHS Path planning approach results: three maneuvers are required. Initial pose = (8m, 6.5m, 5°)



(a) (b)

Figure 3.17: OSEHS Path planning approach results: three maneuvers are required. Initial pose = (5m, 6.5m, 0°).

in the OSEHS approach, the steering angle (Figs. 3.16b and 3.17b) isn't continuous having much more abrupt changes than our approach.

Exhaustive simulations

While the constrained sensor-based formalism is LAS when the desired position lies inside the constraints, the non-holonomy prevents from extending this property to the considered vehicle. Still, in order to assess the convergence of the proposed scheme, several exhaustive simulations were conducted for the different parking cases. To avoid over-cluttering the dissertation, for all the shown cases (Fig.3.18a-3.18f), the initial orientation of the vehicle

is equal to zero ($\theta_{T=0} = 0°$).

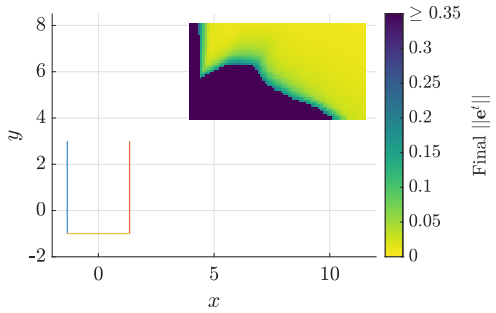
Since the exhaustive simulations are an aggregation of the results obtained from several simulations (like those shown in Figs. 3.4a, 3.5a and 3.5b), each figure consists of a parking spot (represented by 3 lines) adapted to each case and a scatter plot of the initial position of the vehicle, whose color depends (like in Sec. 3.4.1) on the final value of $||\mathbf{e}^t||$. The yellow portion of each scatter plot corresponds to the ROA, the green/blue portions can be interpreted as the boundaries of the ROA and the dark blue one represents the initial positions that are outside of the ROA.

The height of the forward non-parallel analysis windows (Fig. 3.18b and 3.18d) are considerably larger than for their backward counterparts (Fig. 3.18a and 3.18c) to better showcase the ROA's boundaries of the forward case. A similar reasoning applies for the length of parking spot on the parallel cases (Figs. 3.18e and 3.18f).

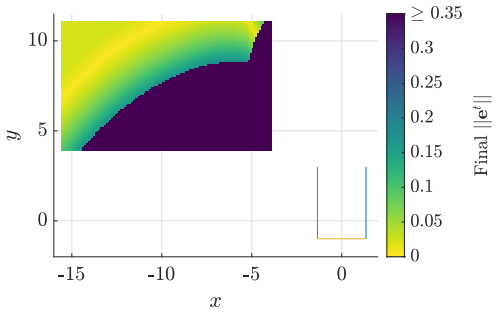
Despite of not being able to change the direction of travel of the vehicle during the maneuver, the ROAs are quite extended (especially for backward maneuvers). Keeping in mind the non-holonomic constraints to which the vehicle is subject to, the saturation values considered in the controller and the physical limits of the steering wheel (which translates to a certain minimum turning radius), the boundaries of the ROAs seem natural. For illustration purposes, consider as an example the backward perpendicular case (Fig. 3.18a). Since $\rho_{m_{\min}} = 4.483\,\text{m}$, it is to be expected that, for initial positions with $x < (\approx 4.4)$, the car is not able to park successfully and instead gets trapped in a local minima due to the nature of the controller used (proportional gradient-based).

Furthermore, it was expected for the ROAs to be smaller for forward maneuvers (even if the parking spot is considerably larger as in the forward parallel case) compared to the backward ones since, as most experienced drivers know, more space is needed to maneuver when parking with a forward motion. In practice, such exhaustive simulation results could be used to quickly assess the feasibility of a parking procedure from the locally perceived data.

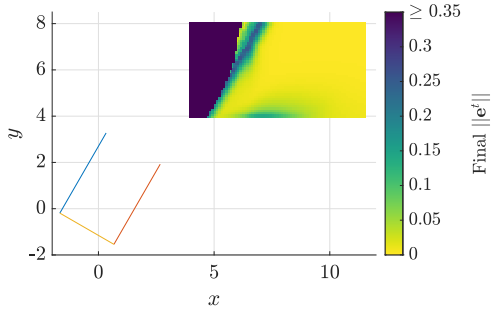
It should be noted that the diagonal green/blue line in the analysis window of (c) corresponds to situations where the car is able to park successfully with a simple maneuver but ending relatively far from the main axis of the parking spot (similar to Fig. 3.5a) while the yellow portion on the left side of said line corresponds to situations where the car has to perform a more complex maneuver in order to satisfy the constraints with the side effect of ending closer to parking spot's main axis



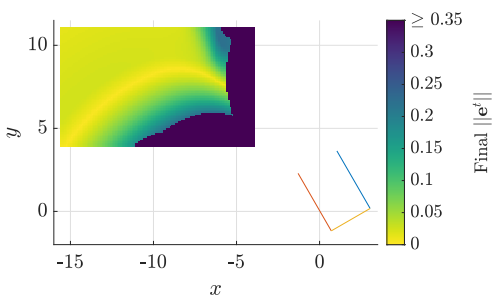
(a) Backward perpendicular case



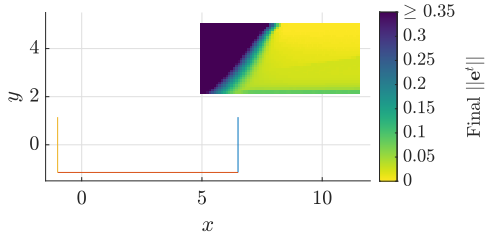
(b) Forward perpendicular case



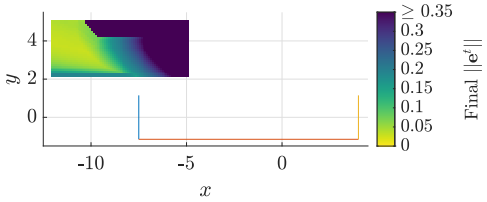
(c) Backward diagonal case



(d) Forward diagonal case



(e) Backward parallel case



(f) Forward parallel case

Figure 3.18: Exhaustive simulations for different types of parking maneuvers with a sampling step of 10cm for the initial position. For non-parallel cases: spot length = 4 m and width = 2.7 m. For (e): spot length = 7.5 m and width = 2.3 m. For (f): spot length = 11.5 m and width = 2.3 m. The color indicates the final error norm, ranging from dark blue for values above 0.35 (typically above ≈ 35 cm in position) to pure yellow for values below 0.03 (typically below ≈ 3 cm in position).

3.4.2 Unparking

Individual cases - MATLAB simulations

To illustrate the behavior of the presented approach when unparking, a few different maneuvers are shown. Similarly to the parking results, the final position is marked with a colored square whose color depends on the final value of $||\mathbf{e}^t||$.

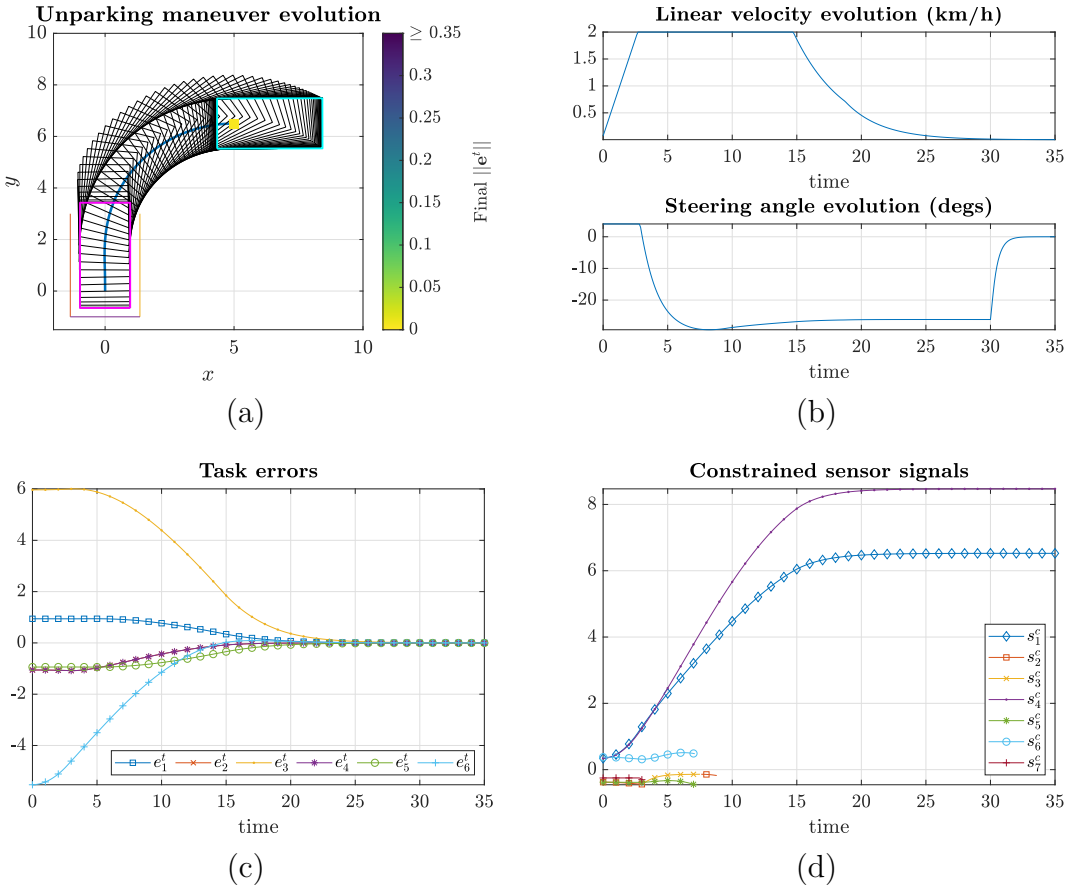
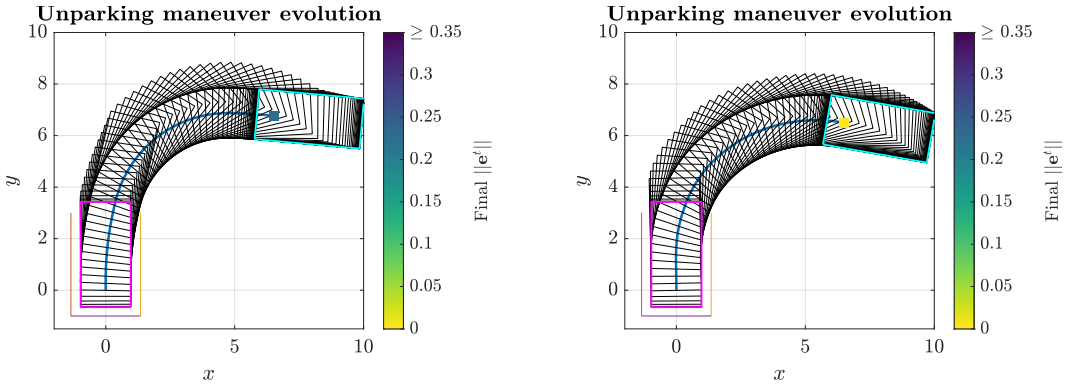


Figure 3.19: Forward ⊥ unparking maneuver signals: (a) performed maneuver (desired pose = (5 m, 6.5 m, 0°)), (b) control signals, (c) task error signal, (d) constrained sensor signals

It can be clearly seen that the vehicle is able to unpark successfully (Fig. 3.19c) from a perpendicular spot with a simple maneuver (Fig. 3.19a) with generally smooth controls signals (Fig. 3.19b) while satisfying the constraints at each time instant (Fig. 3.19d). If one was to shift the desired pose to the right, the vehicle would not be able to perfectly reach it (Fig. 3.20a) since, as it can be seen, in this case the vehicle tries to turn to the



(a) (b)

Figure 3.20: Forward $\perp$ unparking maneuvers. (a) Desired pose = (6.5 m. 6.5 m, 0°), (b) Desired pose = (6.5 m, 6.5 m, −10°)

right when still rather close to its initial position causing it to end farther from the desired pose even if it has more distance to travel. Keeping the same desired position as in Fig. 3.20a but this time with a desired orientation of −10°, it can be seen that the vehicle is now able to reach the desired pose (Fig. 3.20b).

An easy to reach case can be seen in Fig. 3.21a. Again, one can notice that the error converges to values very close to zero (Fig. 3.21c) with smooth control signals (Fig. 3.21b) while satisfying the constraints (Fig. 3.21d). If one was to shift the desired pose upwards, the vehicle would not be able to reach it (Fig. 3.22a) since it does not have enough space to maneuver in a single motion. If the desired pose was shifted again but this time to the right, the vehicle would now be able to reach the desired unpark pose since in this case it would have enough space to maneuver.

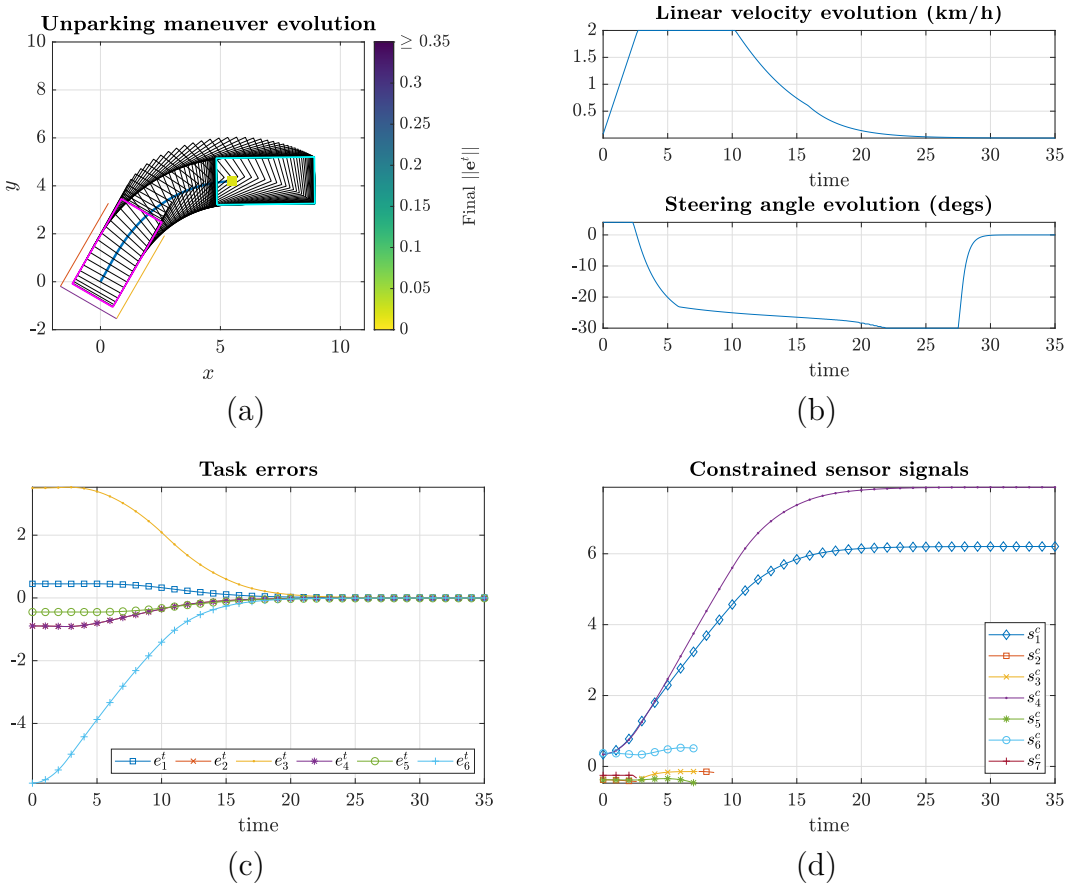


Figure 3.21: Forward diagonal unparking maneuver signals: (a) performed maneuver (desired pose = (5.5 m, 4.2 m, 0°)), (b) control signals, (c) task error signal, (d) constrained sensor signals

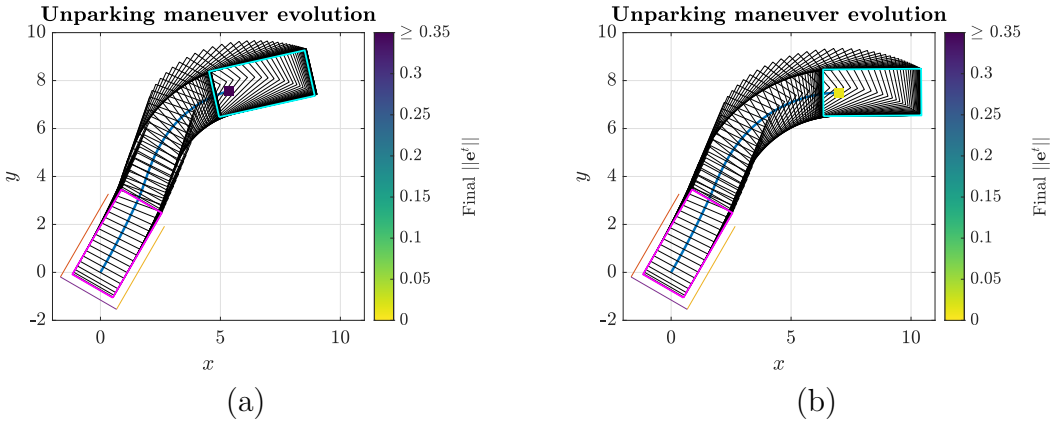


Figure 3.22: Forward diagonal unparking maneuvers. (a) Desired pose = (5.5 m, 7.5 m, 0°), (b) Desired pose = (7 m, 7.5 m, 0°)

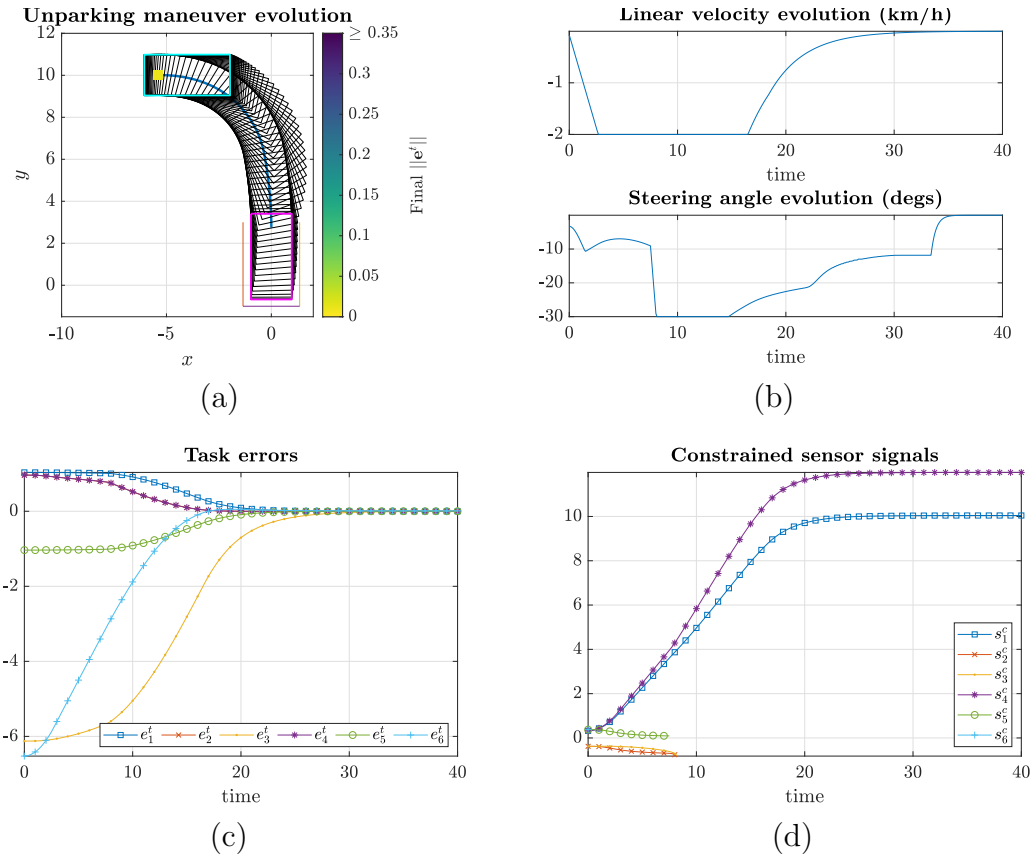
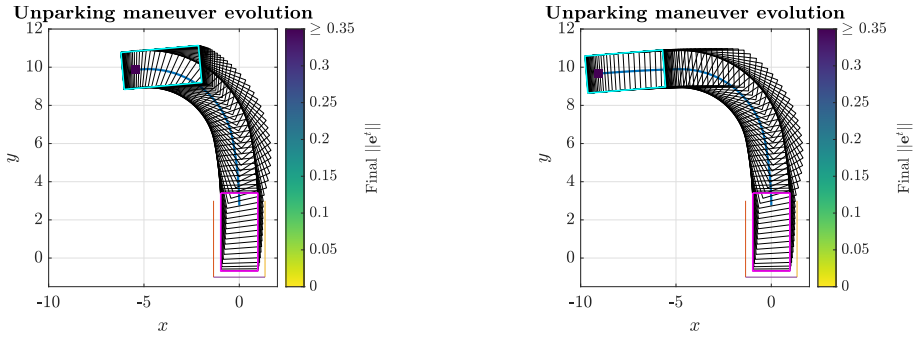


Figure 3.23: Backward $\perp$ unparking maneuver signals: (a) performed maneuver (desired pose $= (-5.4\,\text{m}, 10\,\text{m}, 0°)$), (b) control signals, (c) task error signal, (d) constrained sensor signals

When unparking from non-parallel spots with backward motions, the vehicle's steering angle is rather limited (due to the collision avoidance constraints) until most of its body has exited the parking spot. For this reason, generally the vehicle cannot reach desired unparked poses that are relatively close to the parking spot.

A successful unparking maneuver from a perpendicular parking spot with a backward motion can be seen in Fig. 3.23a. The final task error $||\mathbf{e}^t||$ is very close to zero (Fig. 3.23c), the control signals are generally smooth (Fig. 3.23b) and the constraints are satisfied during the whole maneuver (Fig. 3.23d).

If the desired pose was to be shifted downwards as in Fig. 3.24a, the vehicle would not be able to reach it due to the collision avoidance constraints and physical limits of the steering wheel. If the desired pose would now be shifted to the left as in Fig. 3.24b, the vehicle would still not be able to reach it. This behavior occurs due to the



(a) (b)

Figure 3.24: Backward ⊥ unparking maneuvers. (a) Desired pose = (−5.4 m. 8.1 m, 0°), (b) Desired pose = (−9 m, 8.1 m, 0°)

considered weighting approach, since as the vehicle's orientation approaches the desired one, the priority changes from position to orientation thus causing the car to drive in a mostly-straight line.

When unparking from a diagonal spot, as shown in Fig. 3.25a, the car can reach desired poses that are closer to the parking spot when compared to perpendicular cases. As it can be clearly seen, the final task error $||\mathbf{e}^t||$ is very close to zero (Fig. 3.23c), the control signals are generally smooth (Fig. 3.23b) and the constraints are satisfied during the whole maneuver (Fig. 3.23d). Nevertheless, if desired pose was to be shifted downwards as in Fig. 3.26a, the vehicle would not be able to reach it and, similarly to Fig. 3.24b, if the desired pose would to be shifted as well to the left as in Fig. 3.26b, the vehicle would still not be able to reach it.

A weighting approach that does not only depend on the current value of a given sensor feature but rather on the value of different sensor features could be considered to improve the behavior of the system when unparking.

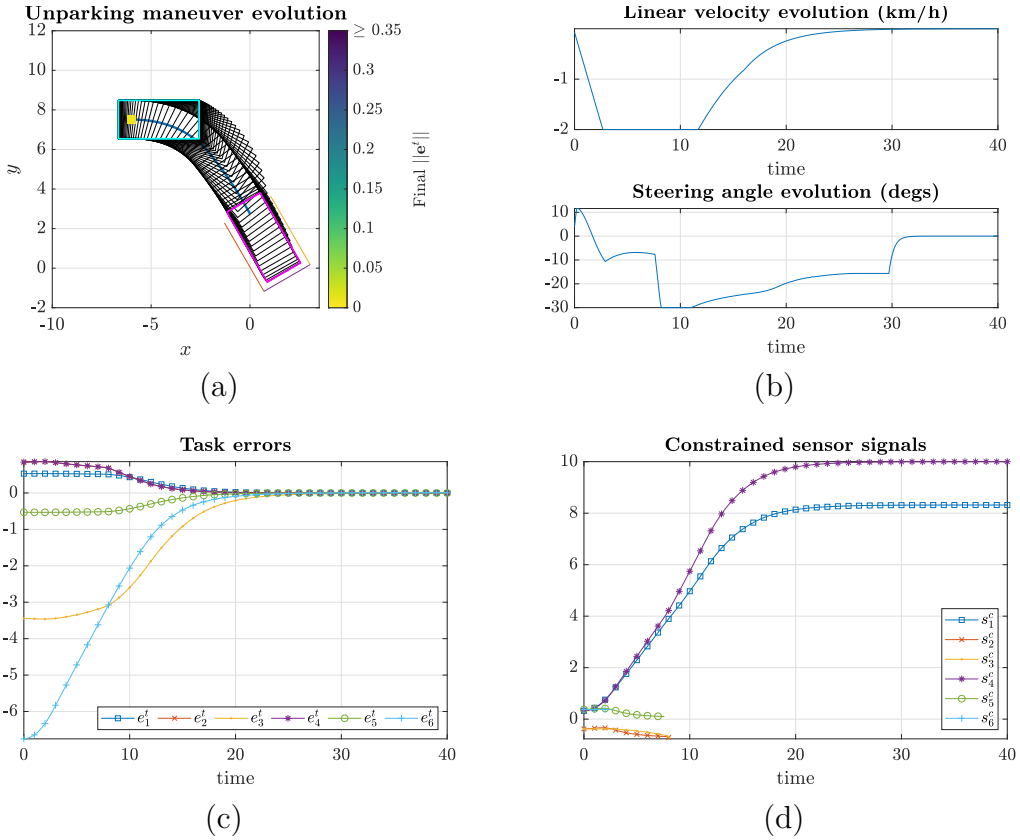


Figure 3.25: Backward diagonal unparking maneuver signals: (a) performed maneuver (desired pose = (−6 m, 7.5 m, 0°)), (b) control signals, (c) task error signal, (d) constrained sensor signals

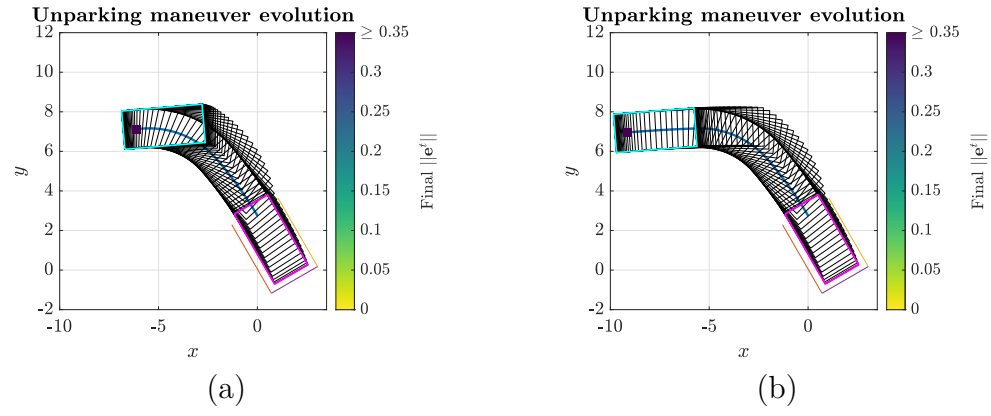
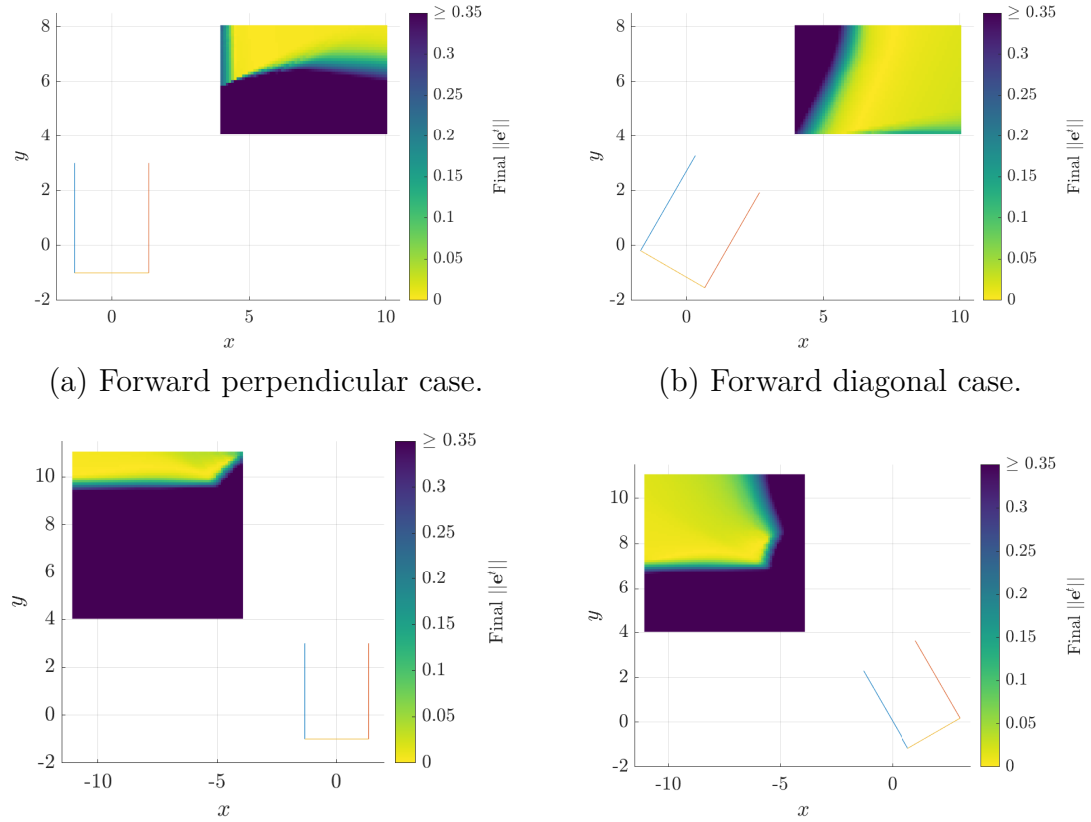


Figure 3.26: Backward diagonal unparking maneuvers. (a) Desired pose = (−6 m. 5.5 m, 0°), (b) Desired pose = (−9 m, 5.5 m, 0°)

Exhaustive simulations

To avoid over-cluttering of the dissertation, for the two shown cases (Figs. 3.27a-3.27b), the desired final orientation of the vehicle is 0°.



(a) Forward perpendicular case.

(b) Forward diagonal case.

(c) Backward perpendicular case.

(d) Backward diagonal case.

Figure 3.27: Exhaustive simulations. Final orientation = 0°. Parking spot length = 4 m and width = 2.7 m

Similarly to parking maneuvers, it was expected for the ROA to be smaller for perpendicular maneuvers compared to the diagonals ones since, as most experienced drivers know, more space is needed to maneuver when unparking from a perpendicular spot.

When comparing this ROAs of unparking maneuvers to their parking counterparts, one can notice that generally they are smaller when unparking. Additionally the ROAs' lower boundaries have a considerably more straight shape. This results can be explained, as mentioned above, by the considered weighting approach that induces a mostly-straight motion once the vehicle's orientation is near the desired one.

Fast prototyping environment

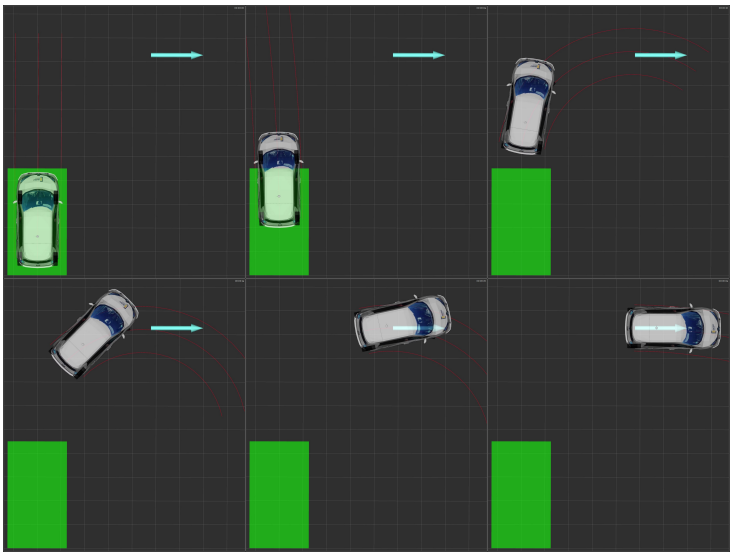


Figure 3.28: Forward perpendicular unparking maneuver in simulation using a homemade fast prototyping environment

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

As it can be seen in Figs. 3.28-3.29, the car is able to unpark successfully from the parking spot (represented by a green rectangle) in one motion while satisfying the constraints during the whole maneuver, with the evolution of the many different signals being very similar to the MATLAB cases.

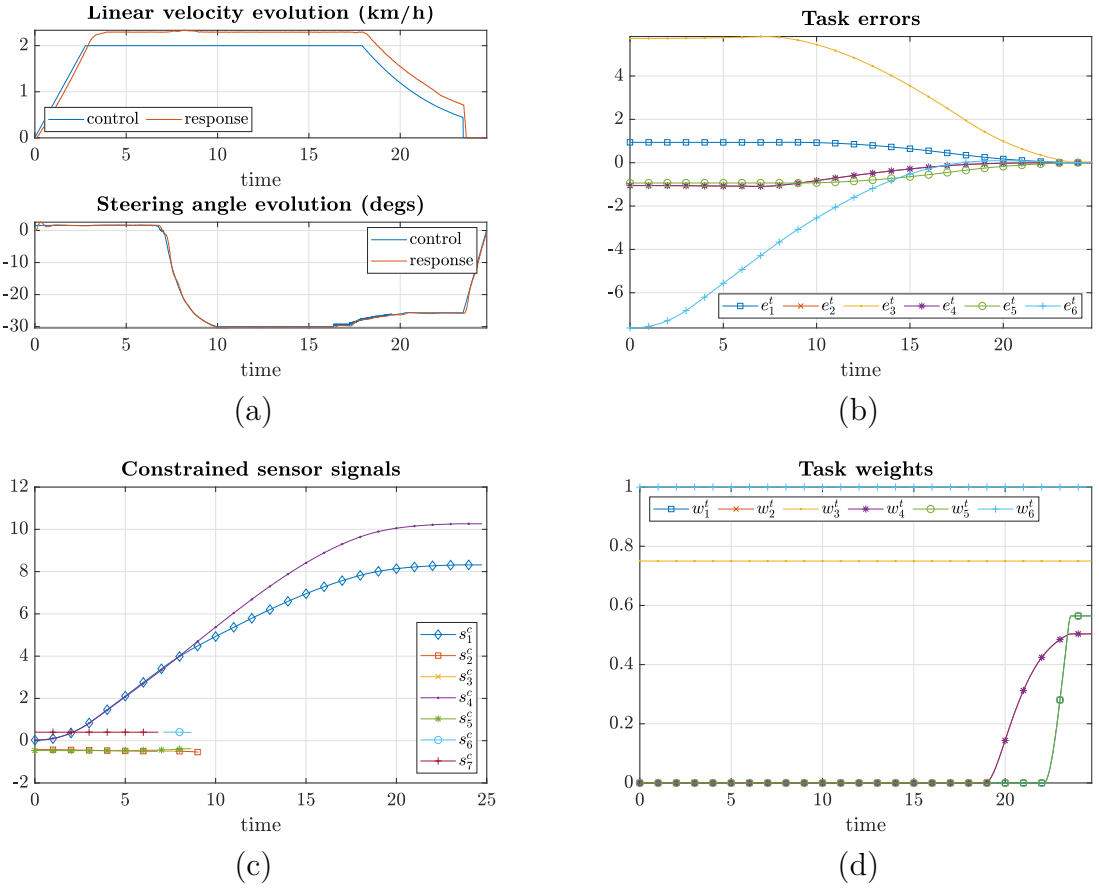


Figure 3.29: Forward perpendicular unparking maneuver signals

3.5 Real experimentation

Real experimentation was conducted for the backward perpendicular parking case (Figs. 3.33 and 3.30) with the controller running at 100 Hz. The controller is implemented in C++ using the solver NLopt [85] with a Sequential Least Squares Programming (SLSQP) algorithm[86].

3.5.1 Using online feature extraction

In this subsection we show the behavior of the system when the sensor features are extracted online using the approach described in Sec. 2.3. For this reason, no localization was necessary thus the vehicle's localization system was not enabled.

The vehicle was located approximately at (4.8 m, 7.9 m) with $\theta \approx 1.35°$ with respect to the parking spot at the beginning of the maneuver. The perceived length and width of

Figure 3.30: Experimental car parking in a perpendicular spot using online feature extraction

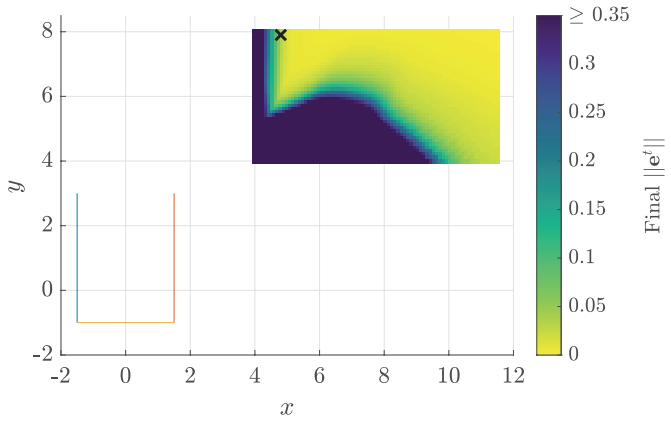


Figure 3.31: Backward $\perp$ case, spot length = 4 m and width = 3 m. Initial orientation = 1.35°

the parking spot were approximately 4 m and 3 m respectively. Taking into account these parameters, an adapted convergence analysis (Fig. 3.31) was conducted where it can be clearly seen that the initial position of the vehicle (denoted by a black $\times$) lies inside of the ROA.

It is obvious that the response of the system, particularly for the linear velocity (Fig. 3.32a), is less than ideal, since it oscillates and has a considerable initial overshoot. This behavior can be attributed to the low-level velocity controller, which at the time when this

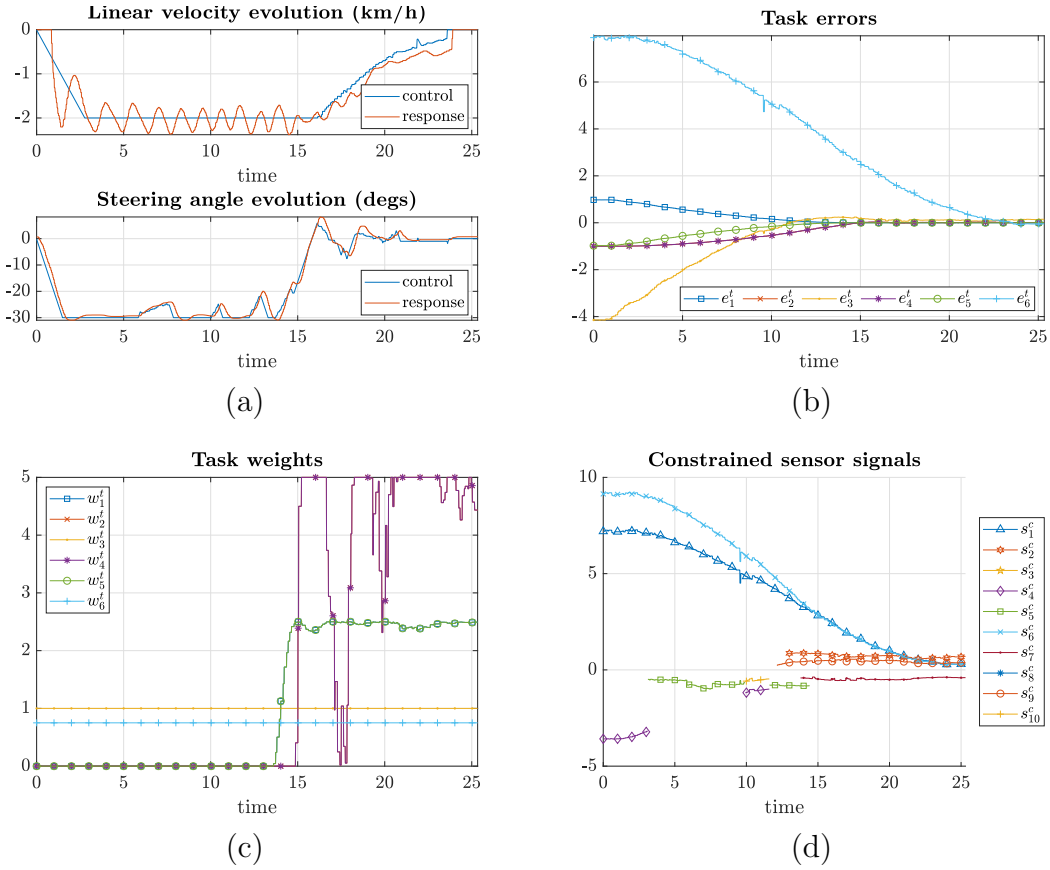


Figure 3.32: Real backward ⊥ parking maneuver signals

experiment was conducted still required considerable tunning to improve the performance at low speeds.

Regarding the sensor feature extraction, one can see in Figs. 3.32b and 3.32d that the different signals present some noise, most notably at around 10 s. Related to this, one can see that the task weights also present a noisy behavior (Fig. 3.32c), which is to be expected since they depend on the task error. The feature extraction could be improved by considering a better suited perception system (one with less blind spots) and/or a more sophisticated parking extraction algorithm.

Despite of the erratic response of the system in addition to the noise coming from the sensor features the vehicle was able to park successfully while satisfying the constraints during the whole maneuver (Fig. 3.32d), thus validating the results obtained from the corresponding convergence analysis. The fact that this approach is able to succeed in such adverse conditions is a testament of the well-known robustness of sensor-based control

strategies and opens the possibility of performing parking/unparking maneuvers even when no localization system is available.

3.5.2 Using virtually generated features

In this subsection we show the behavior of the system when the sensor features are virtually generated from the parking spots predefined in the map but subject to unmodeled 3D orientation and linear velocity disturbances caused by (in this case) a speed bump.

Figure 3.33: Experimental car parking in a perpendicular spot

The vehicle was located approximately at $(-6.7\,\text{m}, 8.1\,\text{m})$ with $\theta \approx 176°$ with respect to the parking spot at the beginning of the maneuver. The length and width of the parking spot were approximately 4.5 m and 2.8 m respectively. Taking into account these parameters, an adapted convergence analysis (Fig. 3.34) was conducted where it can be clearly seen that the initial position of the vehicle (denoted by a black $\times$) lies inside of the ROA.

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. 3.33). The estimated minimum and maximum values of roll and pitch (in degrees) were, respectively, $[-1.7, 0.65]$ and $[-2.02, 1.25]$.

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. 3.33, 3.35b) while satisfying the constraints during the whole maneuver (Fig. 3.35d) thus validating the results obtained from the corresponding convergence analysis.

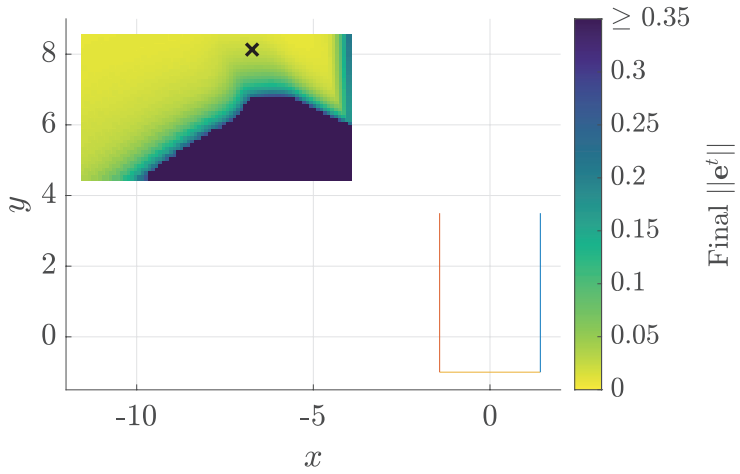


Figure 3.34: Backward ⊥ case, spot length = 4.5m and width = 2.85m. Initial orientation = 176°

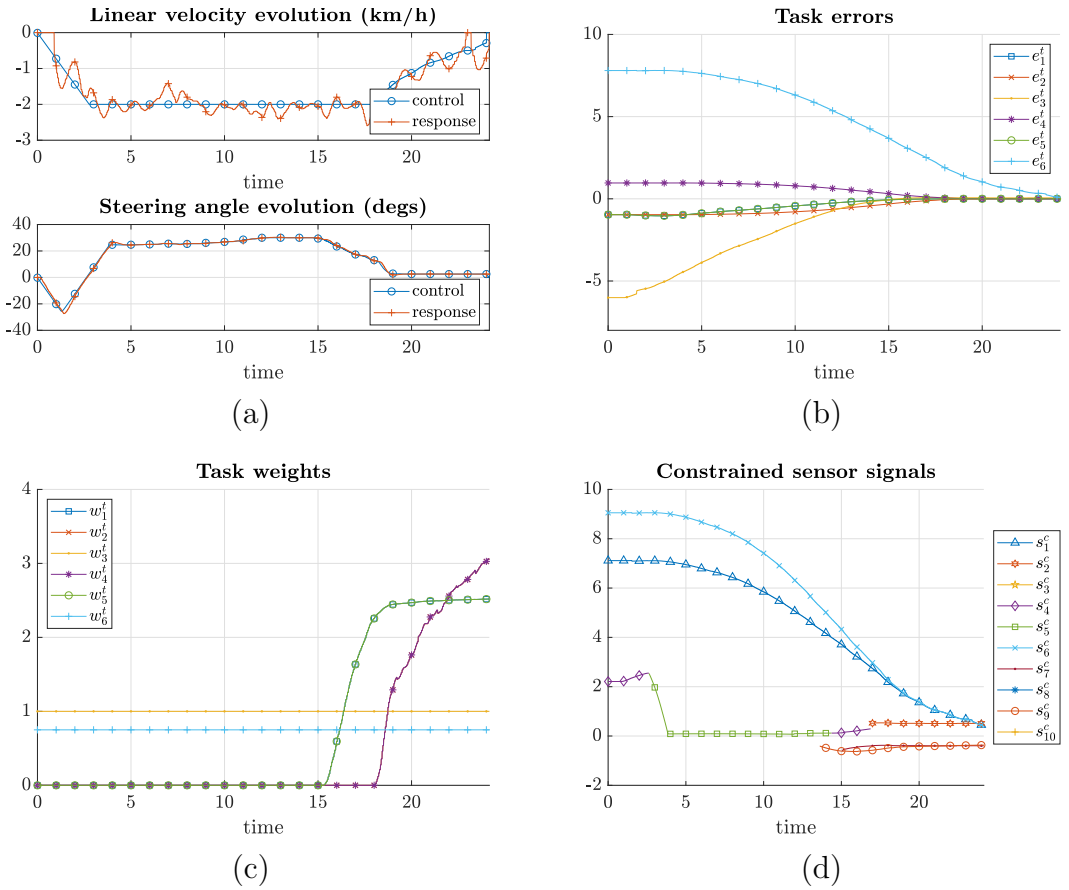


Figure 3.35: Real backward ⊥ parking maneuver signals

The linear velocity setpoint and response can be seen in Fig. 3.35a. Because of the low-level controller, the setpoint given by the parking controller is not perfectly followed, especially when the vehicle passes over the speed bump. Nevertheless, the error remains small enough so that the car manages to park successfully.

3.6 Conclusion

In this chapter we have formalized under a single common framework the six different parking cases: perpendicular, diagonal and parallel for both forward and backward motions. Additionally, we've shown how, with some minor changes (simplifications) on the interaction model, it is possible to perform unparking tasks using the same control approach as for parking tasks although. Nevertheless, the ROAs' of the unparking tasks could be enlarged by considering a different weighting approach.

The task definition is very similar, for both parking or unparking tasks, having just to consider slight adaptations on the interaction model for the different cases. Indeed, if one was interested in unconstrained parking/unparking maneuvers (e.g. because there are no obstacles in risk of collision), the differences among the different cases would be minimal.

Regarding the constraints required for collision avoidance, due to the nature of each of the maneuvers, the differences on the exact definition of the sensor features to use and on deactivation conditions are larger but the main idea remains the same across every case: avoid collision by imposing certain constraints on a set of sensor features (lines defining the boundaries of the parking spot, points at the corners of said spot, etc.).

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 (if online replanning is not feasible) as well as the classical compromise between completeness and computational efficiency when compared to exploration-based path planning techniques.

While a theoretical convergence domain cannot be proved due to non-holonomic constraints, it is possible to assess the convergence of the presented approach by means of exhaustive simulations. Additionally, these simulation results can be exploited to predict the feasability of a given parking maneuver as well as to estimate the final error which, for relatively large regions, may be less than 3 cm. Thanks to the convergence analyses performed, it has been proved that the presented MSBC approach is practically stable (at least in a relevant area for parking/unparking applications). Moreover, the situations

where the vehicle gets trapped in local minima are easily explainable by the nature of the controller used and the constraints acting on the system.

As it can ben seen in Sec. 3.5, the linear velocity setpoint given by the parking controller is not perfectly followed by the low-level controller (Figs. 3.32a and 3.35a). Even if this behavior is not related to the presented technique, it serves to show the robustness of our approach against model-plant mismatches. The first case was conducted with the initial prototype, where the response of the longitudinal velocity was not well identified. As for the second one, 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). Such successful results in spite of different unmodeled disturbances validate the conducted convergence analysis as well as the robustness and effectiveness of the presented approach

Chapter 4

Multi-sensor-based predictive control approach

As shown in the previous chapter, the MSBC framework is able to deal with parking tasks for any type of parking spot but only in one maneuver. To overcome this limitation, a Multi-Sensor-Based Predictive Control (MSBPC) has been explored.

The main idea of the approach presented in the previous chapter remains the same: 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 on 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.

The original MSBPC presented in this chapter allows 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. Additionally, regarding unparking maneuvers, the parallel case has been formalized as well following an analogous reasoning.

By extending the work presented in the previous chapter, 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/unpark 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 (e.g. drive the vehicle away from the parking spot) but that in the end allow to park/unpark successfully.

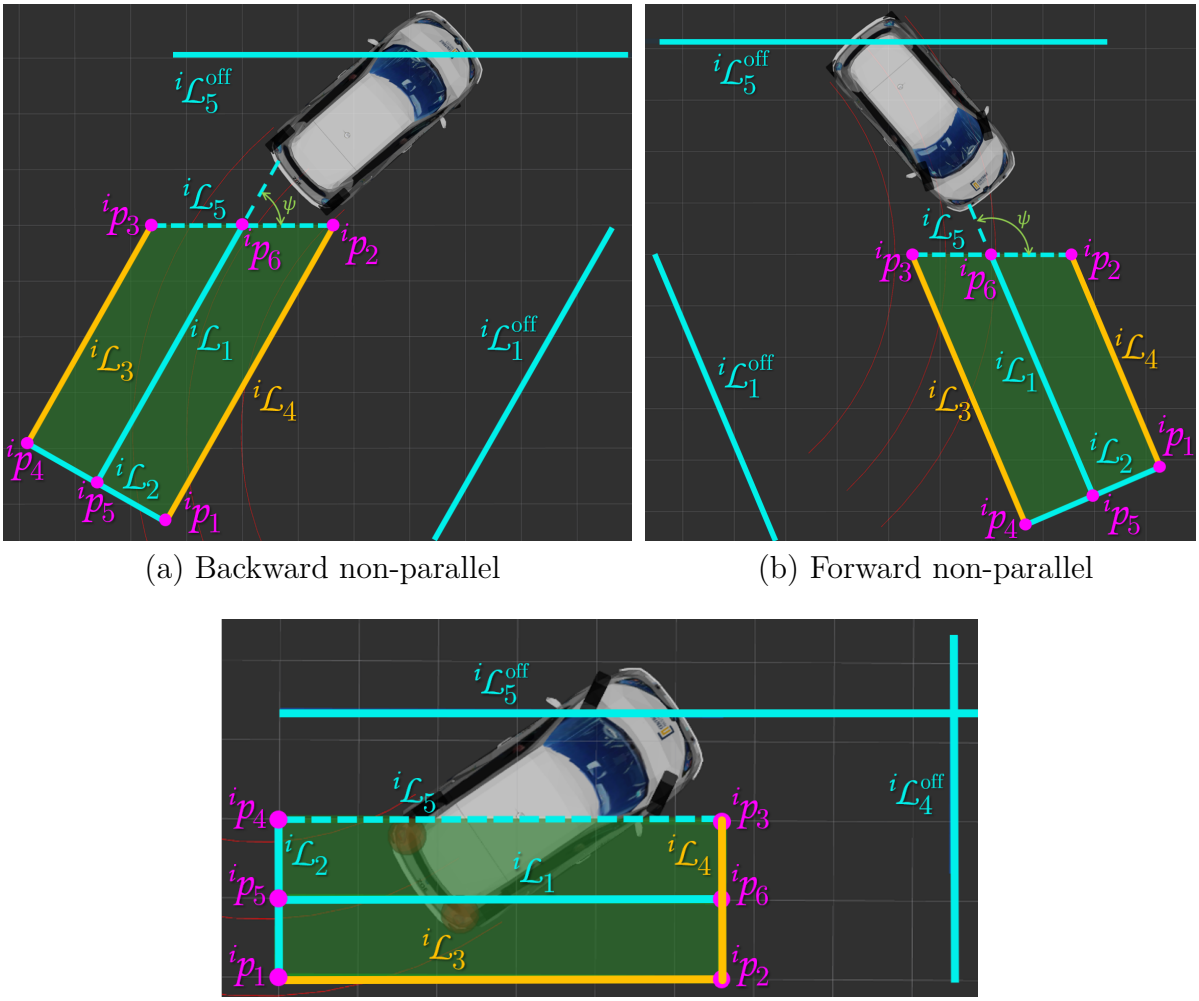
4.1 Parking spots models

For this chapter the parking spots considered are still defined by their corners (points ${}^{i}p_1$ to ${}^{i}p_4$) with the difference being that now we are not restraining them to be rectangular - particularly for diagonal cases where the parking spots are actually right-angled trapezoids (Figs. 4.1a and 4.1b). Furthermore, since for parallel parking spots only parking with backward motions is being considered, the enumeration of the points that define this type of parking spot has been revised (Fig. 4.1c) in an attempt to make it more consistent with the non-parallel cases. Moreover, in addition to the already defined lines we know consider as well ${}^{i}\mathcal{L}_1^{\text{off}}$, ${}^{i}\mathcal{L}_4^{\text{off}}$ and ${}^{i}\mathcal{L}_5^{\text{off}}$ which are simply offsets of, respectively, ${}^{i}\mathcal{L}_1$, ${}^{i}\mathcal{L}_4$ and ${}^{i}\mathcal{L}_5$ away from the parking spot.

The pair of points through which each line passes are defined in Table 4.1.

Table 4.1: Pair of points through which each line passes

Line	*Non-parallel*	*Parallel (backward)*
${}^{i}\mathcal{L}_1$	$({}^{i}p_5, {}^{i}p_6)$	$({}^{i}p_5, {}^{i}p_6)$
${}^{i}\mathcal{L}_2$	$({}^{i}p_1, {}^{i}p_4)$	$({}^{i}p_1, {}^{i}p_4)$
${}^{i}\mathcal{L}_3$	$({}^{i}p_3, {}^{i}p_4)$	$({}^{i}p_1, {}^{i}p_2)$
${}^{i}\mathcal{L}_4$	$({}^{i}p_1, {}^{i}p_2)$	$({}^{i}p_2, {}^{i}p_3)$
${}^{i}\mathcal{L}_5$	$({}^{i}p_2, {}^{i}p_3)$	$({}^{i}p_3, {}^{i}p_4)$



(a) Backward non-parallel

(b) Forward non-parallel

(c) Backward parallel

Figure 4.1: Parking spot models for backward (a) and forward (b) non-parallel types and for (c) parallel ones

4.2 Interaction model

4.2.1 Task sensor features

The set of task sensor features $\mathbf{s}^t$ corresponding to the positioning is composed of two opposing tasks subsets: the main task $\mathbf{s}_{\text{m}}^t$ - used to actually drive the car into the parking spot or towards the desired unparked pose, and the auxiliary task $\mathbf{s}_{\text{a}}^t$ - used to drive the

car away from the parking spot or from the desired unparked pose. It is defined as:

$$\mathbf{s}^t = [\mathbf{s}^t_{\mathrm{m}}; \mathbf{s}^t_{\mathrm{a}}] = [\mathbf{s}^t_{\mathrm{m},1}; \mathbf{s}^t_{\mathrm{m},2}; \mathbf{s}^t_{\mathrm{a,q^{off}}}; \mathbf{s}^t_{\mathrm{a},5^{\mathrm{off}}}] \tag{4.1}$$

with m = 1 and a = 2 for forward maneuvers (i.e. $\mathbf{s}^t_{\mathrm{m}}$ and $\mathbf{s}^t_{\mathrm{a}}$ are expressed in frames $\mathcal{F}_1$ and $\mathcal{F}_2$ respectively) while m = 2 and a = 1 for backward ones. Additionally, when dealing with non-parallel maneuvers, q = 1 (i.e. $\mathbf{s}^t_{\mathrm{a,q^{off}}} = \mathbf{s}^t_{\mathrm{a},1^{\mathrm{off}}}$) while q = 4 for parallel ones. The reasoning for this is that, in practice, often is necessary to perform motions that go against the final goal (e.g. drive the vehicle away from the parking spot) but that in the end will allow to have a successful maneuver.

The interaction matrix $\check{\mathbf{L}}^t_{\mathrm{m}}$ for the features observed by S_{m} is computed by a 2nd order approximation [78]. As it has been shown in the previous chapter, the use of an interaction matrix of the form (3.2) induces a rather interesting and useful behavior: considering parking tasks for illustration purposes, 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 $\check{\mathbf{L}}^t_{\mathrm{a}}$ for the features observed by S_{a} is computed at each iteration and is defined by (2.15).

Parking

For the main task, following the same reasoning as in the previous chapter, ${}^{\mathrm{m}}\mathcal{L}^*_1$ is to be collinear with the vehicle's longitudinal axis (x_m-axis) while ${}^{\mathrm{m}}\mathcal{L}^*_2$ 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. Regarding the auxiliary task, since its objective is to pull the vehicle away from the parking spot, the desired values ${}^{\mathrm{a}}\mathcal{L}^{\mathrm{off}^*}_{\mathrm{q}}$ and ${}^{\mathrm{a}}\mathcal{L}^{\mathrm{off}^*}_{5}$ 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.

Unparking

As mentioned in the previous chapter, the definition of the desired values for the features considered for the task depends on how we want the car to be positioned at the end of the

unparking maneuver. Nevertheless, a general sensible choice for the considered parallel case would be for ${}^{\mathrm{m}}\mathcal{L}_1^*$ to be collinear with the vehicle's longitudinal axis (x_m-axis) and ${}^{\mathrm{m}}\mathcal{L}_2^*$ to be parallel to y_m-axis at a certain distance from the vehicle in order to be able to drive it outside the parking spot. Regarding the auxiliary task, similarly to parking maneuvers, the desired values ${}^{\mathrm{a}}\mathcal{L}_{\mathrm{q}}^{\mathrm{off}^*}$ and ${}^{\mathrm{a}}\mathcal{L}_5^{\mathrm{off}^*}$ are chosen to be collinear with the x_m-axis

4.2.2 Constrained sensor features

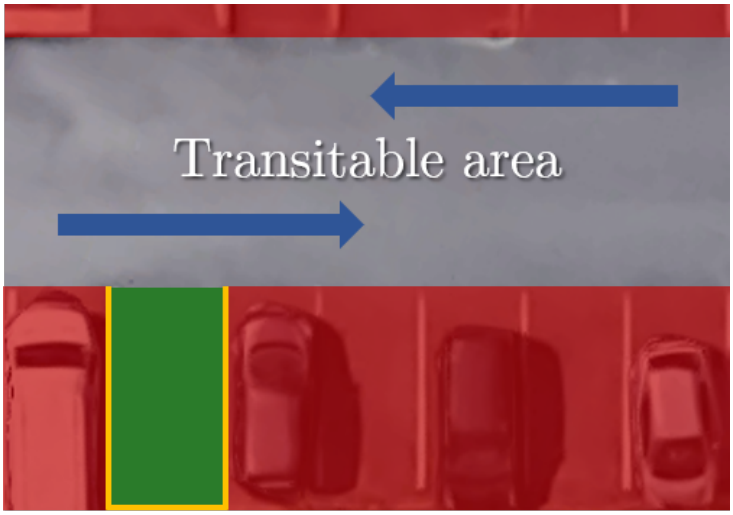


Figure 4.2: Example of a parking environment. The green rectangle denotes the parking spot into which the car should park. Red areas are considered forbidden zones, as such the vehicle should never go into them. Furthermore, it is considered that parking maneuvers can only start inside the transitable area and if no portion of the vehicle is inside any of the forbidden zones while unparking maneuvers can only start if the vehicle is in a parked pose (i.e. inside a parking spot).

Similarly to the MSBC approach, the set of constrained sensor features used for collision avoidance is denoted as $\mathbf{s}^c$. Nevertheless, since with this approach the vehicle is capable of change its longitudinal direction, a more complete (and realistic) set of constrained sensor features has to be taken into account.

Let us consider Fig. 4.2 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/exit 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. 4.2, 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/current 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.

The features defined in Sec. 3.2.2 are considered as well for the current approach. Similarly for the associated interaction matrices with the difference being that for the differences of radii (3.6) and (3.7) there is no need to define any interaction matrix and instead one can compute/predict such features directly from their definition.

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 $\acute{\mathbf{L}}^c_{\mathbf{s}}$. The exact definition of the set of constrained sensor features $\mathbf{s}^c$ is now given case by case.

Parking

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 $\mathbf{s}^c$ is defined as follows:

$$\mathbf{s}^c = [s^c_1; \ldots; s^c_{14}] = [\mathbf{s}^c_3; \mathbf{s}^c_4; \mathbf{s}^c_5; \mathbf{s}^c_6] \tag{4.2}$$

with

$$\mathbf{s}^c_3 = [{}^3h_2; {}^3h_4; {}^3h_5; {}^3X_2; {}^3Y_2; {}^3d_{lat_2}], \tag{4.3a}$$

$$\mathbf{s}^c_4 = [{}^4h_4; {}^4h_5], \tag{4.3b}$$

$$\mathbf{s}_5^c = [{}^5h_3; {}^5h_5], \tag{4.3c}$$

$$\mathbf{s}_6^c = [{}^6h_2; {}^6h_3; {}^6h_5; {}^6X_3]. \tag{4.3d}$$

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 $\mathbf{s}^c$ is defined as follows:

$$\mathbf{s}^c = [s_1^c; \ldots; s_{13}^c] = [\mathbf{s}_3^c; \mathbf{s}_4^c; \mathbf{s}_5^c; \mathbf{s}_6^c] \tag{4.4}$$

with

$$\mathbf{s}_3^c = {}^3h_5, \tag{4.5a}$$

$$\mathbf{s}_4^c = [{}^4h_2; {}^4h_3; {}^4h_5; {}^4X_3; {}^4Y_3], \tag{4.5b}$$

$$\mathbf{s}_5^c = [{}^5h_2; {}^5h_4; {}^5h_5; {}^5X_2; {}^5Y_2; {}^5d_2], \tag{4.5c}$$

$$\mathbf{s}_6^c = {}^6h_5. \tag{4.5d}$$

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, $\mathbf{s}^c$ is defined as follows:

$$\mathbf{s}^c = [s_1^c; \ldots; s_{18}^c] = [\mathbf{s}_3^c; \mathbf{s}_4^c; \mathbf{s}_5^c; \mathbf{s}_6^c; \mathbf{s}_7^c; \mathbf{s}_8^c] \tag{4.6}$$

with

$$\mathbf{s}_3^c = [{}^3h_2; {}^3h_5; {}^3X_3; {}^3Y_3; {}^3d_{lat_3}; {}^3X_4; {}^3Y_4], \tag{4.7a}$$

$$\mathbf{s}_4^c = [{}^4h_4; {}^4h_5; {}^4X_3; {}^4Y_3; {}^4d_3], \tag{4.7b}$$

$$\mathbf{s}_5^c = [{}^5h_4; {}^5h_5], \tag{4.7c}$$

$$\mathbf{s}_6^c = [{}^6h_2; {}^6h_5], \tag{4.7d}$$

$$\mathbf{s}_7^c = {}^7h_3, \tag{4.7e}$$

$$\mathbf{s}_8^c = {}^8h_3. \tag{4.7f}$$

Unparking

Given that the constrained sensor features considered for parking maneuvers already ensure that the vehicle will only evolve inside the allowed regions, when dealing with unparking maneuvers, it is possible to simply consider the same set of constrained sensor features as in its analogous parking task. Therefore, for forward parallel unparking maneuvers, $\mathbf{s}^c$ is the same as for backward parking ones.

Extension for dealing with moving obstacles

If the appropriate considerations are taken in the interaction model, sensor-based control strategies are capable of tracking moving targets [87], or as in our case, impose constraints on them in order to avoid collision. (2.7) can be modified to account for the velocity of a moving obstacle as follows:

$$\dot{\mathbf{s}} = \breve{\mathbf{L}}_{\mathbf{s}}(\breve{\mathbf{v}}_m - \breve{\mathbf{v}}_{\mathrm{mo}}) \tag{4.8}$$

where $\breve{\mathbf{v}}_{\mathrm{mo}}$ is 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 [88] 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 ${}^ip_{\mathrm{mo}}$ with a given orientation. As such, the constrained sensor features can be defined as:

$$\mathbf{s}^{c_{\mathrm{mo}}} = [s_1^{c_{\mathrm{mo}}}; \ldots; s_8^{c_{\mathrm{mo}}}] = [\mathbf{s}_3^{c_{\mathrm{mo}}}; \mathbf{s}_4^{c_{\mathrm{mo}}}; \mathbf{s}_5^{c_{\mathrm{mo}}}; \mathbf{s}_6^{c_{\mathrm{mo}}}] \tag{4.9}$$

with

$$\mathbf{s}_3^{c_{\mathrm{mo}}} = [{}^3X_{\mathrm{mo}}; {}^3Y_{\mathrm{mo}}], \tag{4.10a}$$

$$\mathbf{s}_4^{c_{\mathrm{mo}}} = [{}^4X_{\mathrm{mo}}; {}^4Y_{\mathrm{mo}}], \tag{4.10b}$$

$$\mathbf{s}_5^{c_{\mathrm{mo}}} = [{}^5X_{\mathrm{mo}}; {}^5Y_{\mathrm{mo}}], \tag{4.10c}$$

$$\mathbf{s}_6^{c_{\mathrm{mo}}} = [{}^6X_{\mathrm{mo}}; {}^6Y_{\mathrm{mo}}]. \tag{4.10d}$$

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 $({}^iX_{\mathrm{mo}}, {}^iY_{\mathrm{mo}})$ 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.

Constraints deactivation

Similarly to the MSBC approach, some constraints must be deactivated under certain conditions in order to be able to park successfully. The constraints deactivation conditions used to obtain the results presented in this work are detailed in the appendix.

4.3 Control

The presented MSBPC approach is based on the Visual Predictive Control (VPC) described in [63] with some modifications to impose an exponential decay of error related to the main task $\mathbf{e}_{\mathrm{m}}^t = \mathbf{s}_{\mathrm{m}}^t - \mathbf{s}_{\mathrm{m}}^{t^*}$. 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).

As a recall, the control input of the robotized vehicle is defined as:

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

with ϕ, considering (2.5), being mapped to $\dot{\theta}_m$ by

$$\dot{\theta}_m = \frac{v_{x_m}}{\rho_m}.$$

4.3.1 Structure

The control structure is based on the Internal-Model-Control (IMC) structure [89] (Fig. 4.3). The System block contains the robotized vehicle system and sensors whose input is

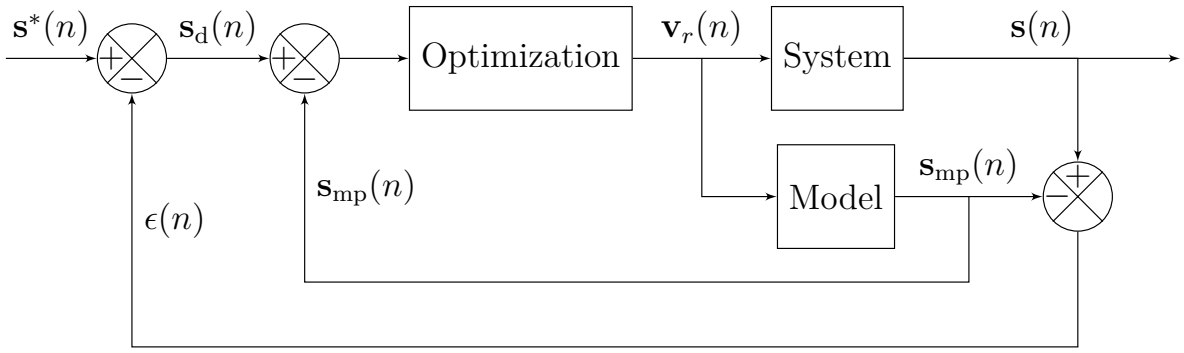


Figure 4.3: Control structure [63]

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:

$$\epsilon(n) = \mathbf{s}(n) - \mathbf{s}_\mathrm{mp}(n) \tag{4.11}$$

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. 4.3:

$$\mathbf{s}_\mathrm{d}(n) = \mathbf{s}^*(n) - \epsilon(n) = \mathbf{s}^*(n) - (\mathbf{s}(n) - \mathbf{s}_\mathrm{mp}(n)) \tag{4.12}$$

from where it is possible to deduce

$$\mathbf{s}_\mathrm{d}(n) - \mathbf{s}_\mathrm{mp}(n) = \mathbf{s}^*(n) - \mathbf{s}(n) \tag{4.13}$$

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. 3.2 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.

4.3.2 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:

$$|v_{x_m}| < v_{\max}, \tag{4.14a}$$

$$|\phi| < \phi_{\max}, \tag{4.14b}$$

where $v_{\max}$ is an adaptive saturation value imposing a deceleration profile based on the velocity profile shown in [34] 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:

$$|\dot{v}_{x_m}| \leq \dot{v}_{\max}, \tag{4.15a}$$

$$|\dot{\phi}| \leq \dot{\phi}_{\max}, \tag{4.15c}$$

$$|\ddot{v}_{x_m}| \leq \ddot{v}_{\max}, \tag{4.15b}$$

$$|\ddot{\phi}| \leq \ddot{\phi}_{\max}, \tag{4.15d}$$

$$|\dddot{\phi}| \leq \dddot{\phi}_{\max}. \tag{4.15e}$$

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

$$\mathbf{s}^c_{\min} \leq \mathbf{s}^c_{\mathrm{mp}} \leq \mathbf{s}^c_{\max} \tag{4.16}$$

and, if applicable,

$$\mathbf{s}^{c_{\mathrm{mo}}}_{\min} \leq \mathbf{s}^{c_{\mathrm{mo}}}_{\mathrm{mp}} \leq \mathbf{s}^{c_{\mathrm{mo}}}_{\max} \tag{4.17}$$

By writing the constraints (4.15) and (4.16) as nonlinear functions:

$$C(\mathbf{v}_r) \leq 0 \tag{4.18}$$

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

4.3.3 Mathematical formulation

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

$$\begin{aligned} &\min\ J(\mathbf{v}_r) \\ &\tilde{\mathbf{v}}_r \in \mathbb{C} \end{aligned} \tag{4.19}$$

with

$$J(\mathbf{v}_r) = \sum_{j=n+1}^{n+N_p} \Big([\mathbf{s}_\mathrm{d} - \mathbf{s}^t_\mathrm{mp}(j)]^\mathsf{T} \mathbf{Q}(j) [\mathbf{s}_\mathrm{d} - \mathbf{s}^t_\mathrm{mp}(j)] + \mathbf{v}_m(j-1)^\mathsf{T}\, \mathbf{R}(j-1)\, \mathbf{v}_m(j-1) \Big) \tag{4.20}$$

and

$$\tilde{\mathbf{v}}_r = \{\mathbf{v}_r(n), \mathbf{v}_r(n+1), \ldots, \mathbf{v}_r(n+N_c), \ldots, \mathbf{v}_r(n+N_p-1)\} \tag{4.21}$$

subject to

$$\mathbf{s}^t_\mathrm{mp}(j) = \mathbf{s}^t_\mathrm{mp}(j-1) + \mathbf{L}^t_\mathbf{s}(j-1) T_s \mathbf{v}_m(j-1) \tag{4.22a}$$

$$\acute{\mathbf{s}}^c_\mathrm{mp}(j) = \acute{\mathbf{s}}^c_\mathrm{mp}(j-1) + \acute{\mathbf{L}}^c_\mathbf{s}(j-1) T_s \mathbf{v}_m(j-1) \tag{4.22b}$$

and, if applicable,

$$\mathbf{s}^{c_\mathrm{mo}}_\mathrm{mp}(j) = \mathbf{s}^{c_\mathrm{mo}}_\mathrm{mp}(j-1) + \check{\mathbf{L}}^{c_\mathrm{mo}}_\mathbf{s}(j-1) T_s (\check{\mathbf{v}}_m(j-1) - \check{\mathbf{v}}_\mathrm{mo}(j-1)) \tag{4.23}$$

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 [53], particularly for a finite N_p. As mentioned previously, for sensor-based techniques, only local asymptotic stability can be demonstrated [74] when sensor data is kept in the sensor space. Nonetheless, as evoked in Section 1.4, it has been shown that the local asymptotic stability property is kept for the optimal MSBPC considered [90]. 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 to perform successfully sensor-based tasks while satisfying the considered constraints [63], [90]. The performance of the proposed approach is illustrated in Sec. 4.4.

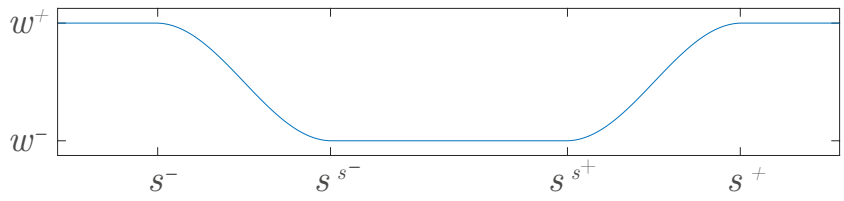


Figure 4.4: Generic weighting function $w(s)$

Weighting strategy

As mentioned in Sec.4.2.1, $\mathbf{s}_{\mathrm{m}}^t$ contains the features used to actually drive the car into the parking spot (or towards the desired unparked pose), thus the minimization of the error $\mathbf{e}_{\mathrm{m}}^t = \mathbf{s}_{\mathrm{m}}^t - \mathbf{s}_{\mathrm{m}}^{t^*}$ is what drives the vehicle towards the desired parked (unparked) pose, generally with a backward motion for backward maneuvers and forward ones for the forward case, while $\mathbf{e}_{\mathrm{a}}^t = \mathbf{s}_{\mathrm{a}}^t - \mathbf{s}_{\mathrm{a}}^{t^*}$ would be generally minimized with opposite directions of motion. 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 weighted matrix $\mathbf{Q}$, which remains constant along the prediction horizon. It is defined as:

$$\mathbf{Q} = \left[\begin{array}{c|c} Q_{\mathrm{m}}\,\mathbf{W}_{\mathrm{m}}^t & 0_{6\times 6} \\ \hline 0_{6\times 6} & Q_{\mathrm{a}}\,\mathbf{W}_{\mathrm{a}}^t \end{array}\right] \tag{4.24}$$

with $\mathbf{W}_{\mathrm{m}}^t = \mathrm{diag}(w_1^t, \ldots, w_6^t)$, $\mathbf{W}_{\mathrm{a}}^t = \mathrm{diag}(w_7^t, \ldots, w_{12}^t)$ and

$$Q_{\mathrm{a}} = \begin{cases} 0 & \text{if } \|\mathbf{s}_{\mathrm{m}_{\mathcal{L}_1}} - \mathbf{s}^*_{\mathrm{m},1_{\mathcal{L}_1}}\| < \epsilon_{\mathcal{L}_1} \text{ and } Q_{\mathrm{m}} > 0 \\ 1 - Q_{\mathrm{m}} & \text{otherwise} \end{cases} \tag{4.25}$$

where $\epsilon_{\mathcal{L}_1}$ is a small positive scalar value that serves to nullify Q_{a} (and consequently the influence of $\mathbf{s}_{\mathrm{a}}^t$) when the vehicle is almost collinear to ${}^{\mathrm{m}}\mathcal{L}_1$. The values of $w_i^t\ \forall i = \{1\text{–}12\}$ and Q_{m} are given by the weighting functions $w_i^t(\mathbf{s}, \mathbf{v}_r(n))$ and $Q_{\mathrm{m}}(\mathbf{s}, \mathbf{v}_r(n))$ which are composed of different combinations of a generic smooth weighting function w (Fig. 4.4). Due to the different nature of each of the considered cases, the exact definition of these functions varies slightly depending on the type of maneuver although their structure remains similar. For illustration purposes, the non-parallel backward parking case is now considered.

Let us consider a boolean variable `move_fwd` that serves as a hint of whether the

vehicle should move forward or not. In general `move_fwd` is set to true when the car starts moving forward and to false otherwise. If the vehicle is almost collinear to ${}^{\mathrm{m}}\mathcal{L}_1$ (i.e. $\|\mathbf{s}_{\mathrm{m}_{\mathcal{L}_1}} - \mathbf{s}^*_{\mathrm{m}_{\mathcal{L}_1}}\| < \epsilon_{\mathcal{L}_1}$) or near the desired parked pose (i.e. $\|\mathbf{e}^t_{\mathrm{m}}\| < \epsilon_{\mathrm{m}}$), `move_fwd` is set to false. Additionally, it's value is flipped if a local minima is detected.

The vehicle is considered to be at (or in the vicinity of) a local minima if $\|\mathbf{e}^t_{\mathrm{m}}\| \geq \epsilon_{\mathrm{m}}$ and $|v_{x_m}|$ has remained below a given threshold v_{th} during a minimum number of iterations `it` or $|v_{x_m}| < v_{\mathrm{th}}$ for at least `it` consecutive elements of a subset (from the beginning n up to $n + N_c$) of the optimal control sequence $\tilde{\mathbf{v}}_r$. As such, if these conditions are satisfied `move_fwd` should be flipped.

The value of the weight multiplier Q_{m} depends on `move_fwd` as shown in (4.26). The idea is to drive the vehicle generally with a backward motion if it starts to move while being on the right side of ${}^{\mathrm{i}}\mathcal{L}_1$ and with a forward motion if the vehicle starts on the left side of said line or if a local minima had been detected when moving in reverse, generally not driving farther away than ${}^{\mathrm{i}}\mathcal{L}_1^{\mathrm{off}}$.

$$Q_{\mathrm{m}} = \begin{cases} w(e^t_9)\, w({}^3h_5, {}^5h_5, {}^6h_5) & \text{if } \texttt{moving\_fwd} \\ w(e^t_3) & \text{otherwise.} \end{cases} \tag{4.26}$$

Regarding $\mathbf{W}^t_{\mathrm{m}}$, 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 $\mathbf{W}^t_{\mathrm{m}}$ can be defined as:

$$w^t_i = w(e^t_6)\ \forall i \in \{1, 2, 4, 5\}. \tag{4.27}$$

As for $\mathbf{W}^t_{\mathrm{a}}$, 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. Therefore, the diagonal elements of $\mathbf{W}^t_{\mathrm{a}}$ are given by:

$$w^t_7 = w^t_8 = w({}^5h_5, {}^6h_5)\, w({}^mh_1)\, w(s^t_9), \tag{4.28a}$$

$$w^t_9 = \Big(w(e^t_3) + \max\Big(w(e^t_7), w(e^t_8)\Big)\Big)\, w(e^t_9), \tag{4.28b}$$

$$w_{10}^t = w_{11}^t = w(s_9^t) + w(e_3^t) + w({}^5h_5), \tag{4.28c}$$

$$\begin{aligned} w_{12}^t = {} & w(e_9^t)\; w({}^mh_1) + w({}^3h_5, {}^5h_5, {}^6h_5)\; \max\left(w(e_7^t), w(e_8^t)\right) \\ & + w({}^5h_5)\,(1 - w(e_3^t))\; \left(1 - \max\left(w(e_7^t), w(e_8^t)\right)\right) w(e_9^t)\, w({}^3h_5). \end{aligned} \tag{4.28d}$$

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 $\dot{\theta}$ are regulated by means of the weighted matrix $\mathbf{R}$, which remains constant along the prediction horizon. It is defined as:

$$\mathbf{R} = \begin{bmatrix} Q_{\mathrm{m}} & 0 \\ 0 & 1 \end{bmatrix}. \tag{4.29}$$

4.4 Results

For the results shown in this section, the parameters in Table 4.2 are considered. Additionally, $\epsilon_{\mathcal{L}_1} = 0.125$ for non-parallel maneuvers while $\epsilon_{\mathcal{L}_1} = 0.035$ for parallel parking ones and $\epsilon_{\mathcal{L}_1} = 0.075$ for parallel unparking. The value of $\phi_{\max}$ corresponds to the maximum steering angle of the real vehicle while the rest of the parameters were determined by empirical testing, nevertheless some guidelines on how to tune them can be given:

- The maximum longitudinal velocity $v_{\max}$ and the maximum values of the constraints (4.15) 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 [53].

Table 4.2: Control-related vehicle parameters

Parameteres	*Notation*	*Value*
Sampling time	T_s	$0.1\,\mathrm{s}$
Control horizon	N_c	10 $(1\,\mathrm{s})$
Prediction horizon	N_p	25 $(2.5\,\mathrm{s})$
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	$\dddot{\phi}_{\max}$	$0.9\,\mathrm{rad/s^3}$
Threshold value for $\mathbf{e}_{\mathrm{m}}^{t}$	ϵ_{m}	0.1
Minimum velocity threshold	v_{th}	$0.06\,\mathrm{m/s}$
Iterations threshold	`it`	5

- 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.

Unless otherwise noted, simulations are carried out in MATLAB using the *fmincon* solver with a Sequential Quadratic Programming (SQP) algorithm. For real experimentation, the controller is implemented in C++ using the solver NLopt [85] with a Sequential Least Squares Programming (SLSQP) algorithm[86].

4.4.1 Parking simulation results

Individual cases - MATLAB simulations

To illustrate the performance of the proposed approach, a few examples of each type of parking situation are shown below. Similarly to Sec. 3.4.1, the initial position is marked with a colored (yellow and light green in the shown cases) square whose color depends on

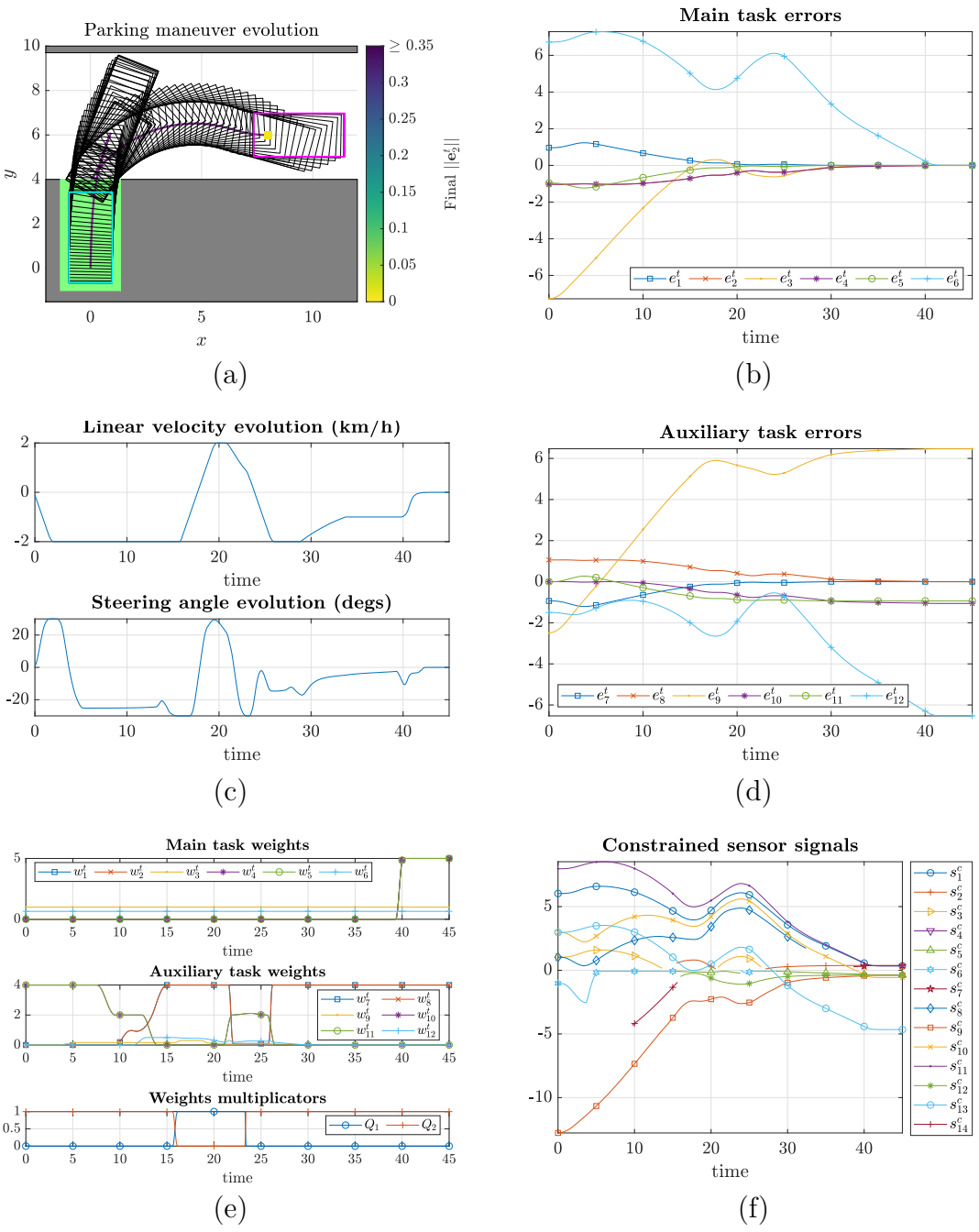
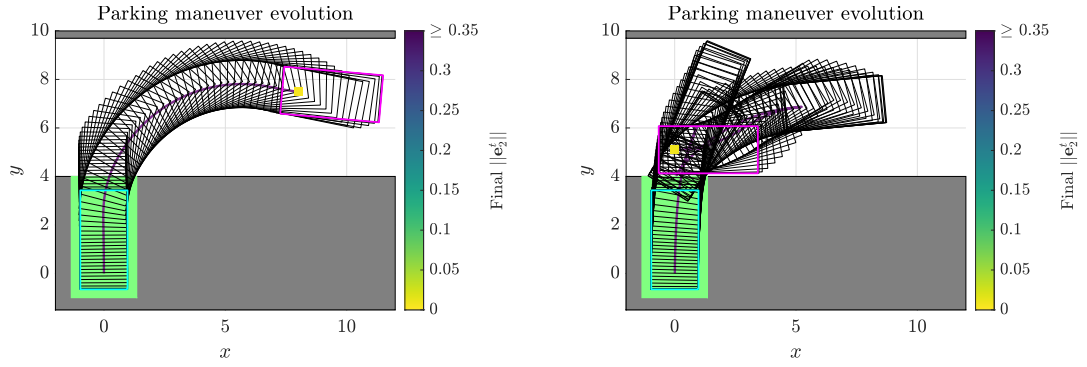


Figure 4.5: Constrained ⊥ backward parking maneuver signals: (a) performed maneuver, (c) control signals, (e) weighting-related signals, (b) main and (d) auxiliary task errors and (d) constrained sensor signals. Initial pose = (8 m, 6 m, 0°)

the final value of $||\mathbf{e}_{\mathrm{m}}^t||$ as the associated colorbar indicates. The initial and final bounding rectangles of the vehicle are colored in magenta and cyan respectively.



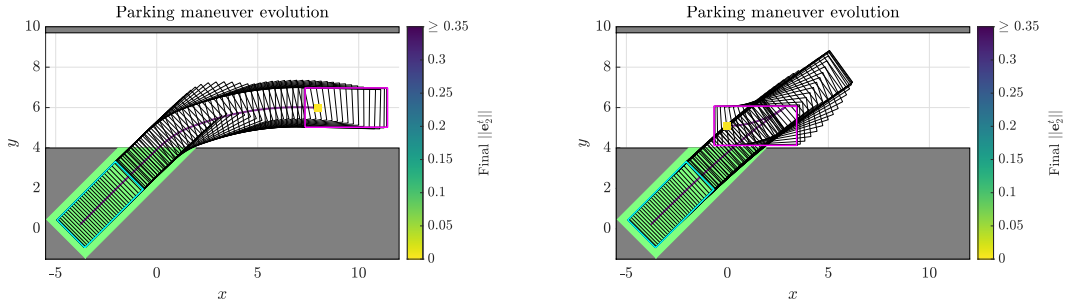
(a) (b)

Figure 4.6: Constrained ⊥ backward parking maneuvers. (a) Initial pose = (8 m, 7.5 m, −5°), (b) Initial pose = (0 m, 5.1 m, 0°)

In addition to the parking maneuver evolution, the evolution of the different control-related signals can be seen in Fig. 4.5. Thanks to the constraints related to the controls signals, they evolve smoothly in general (Fig. 4.5c). The (active) constraints imposed on $\mathbf{s}^c$ are satisfied at each time instant (Fig. 4.5f) ensuring a collision-free maneuver. It can be seen how the main task error $\mathbf{e}_2^t$ (Fig. 4.5b) is minimized as the vehicle moves backwards (i.e. towards the parking spot) while $\mathbf{e}_1^t$ (Fig. 4.5d) is minimized with motions in the opposite direction (i.e. away from the parking spot). Additionally, one can notice that when Q_2 is larger than Q_1 (Fig. 4.5e), the vehicle is moving backwards and when Q_1 is larger the opposite occurs. The divergence of the positioning error of the auxiliary task (e_9^t and e_{12}^t) towards the end of the maneuver (Fig. 4.5d) is to be expected since Q_1 is null towards the end. 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 those related to s_9^t and s_{12}^t pull the vehicle out of the parking spot while those related to s_7^t and s_8^t try to keep the vehicle parallel to main axis of the parking spot.

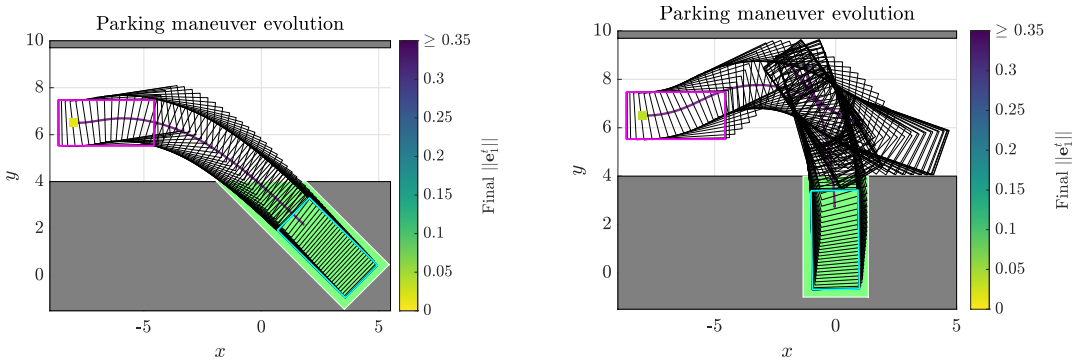
Two more backward perpendicular cases can be seen in Fig. 4.6. Fig. 4.6a shows a case where parking with a single maneuver is possible while Fig. 4.6b presents a rather challenging initial pose from where the vehicle has to perform four maneuvers in order to park. For the three backward perpendicular cases shown, the final value of $||\mathbf{e}_2^t||$ is rather small (< 0.0102).

The initial poses of the cases presented in Figs. 4.7a and 4.7b are, respectively, the



(a) (b)

Figure 4.7: Constrained diagonal backward parking maneuvers. (a) Initial pose = (8 m, 6 m, 0°), (b) Initial pose = (0 m, 5.1 m, 0°)

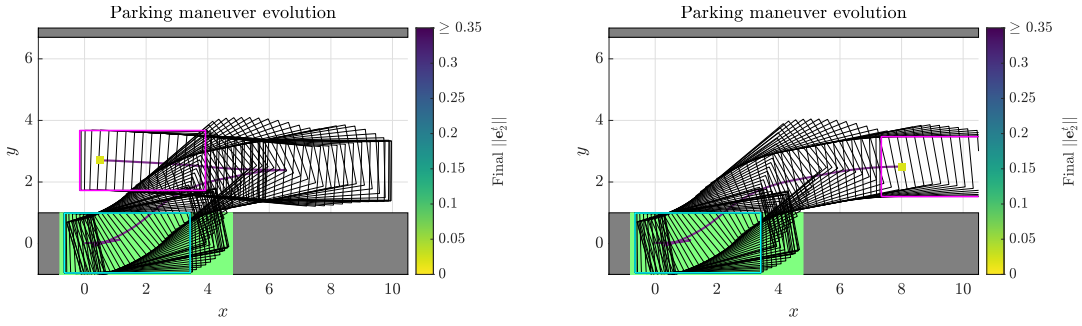


(a) (b)

Figure 4.8: Constrained non-parallel forward parking maneuvers: (a) diagonal and (b) perpendicular. Initial pose = (−8 m, 6.5 m, 0°)

same as those in Figs. 4.5a and 4.6b. It can be seen how, as expected, when considering diagonal parking spots (Fig. 4.7) instead of perpendicular ones the complexity of the maneuvers is reduced. The case shown in Fig. 4.7a 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. 4.7b, the number of maneuvers was reduced to only two compared to four for the perpendicular case.

As most experienced drivers know, parking maneuvers with forward motions are generally 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. 4.8 share the same initial pose. On the one hand, for the diagonal case (Fig. 4.8a) parking with a single maneuver is possible with a final $||\mathbf{e}_1^t|| = 0.0134$.



(a) (b)

Figure 4.9: Constrained ‖ backward parking maneuvers. (a) Initial pose = (0.5 m, 2.7 m, 0°), (b) Initial pose = (8 m, 2.5 m, 0°)

On the other hand, for the perpendicular case shown in Fig. 4.8b the vehicle has to perform five maneuvers (four changes of direction), achieving a final $||\mathbf{e}_1^t|| = 0.0372$. 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. 4.9. In Fig. 4.9a, the vehicle starts the parking maneuver by moving forward in order to preposition itself (similarly to Fig. 4.6b) and then performs three more motions leading to a total of four maneuvers to park with a final $||\mathbf{e}_2^t|| = 0.0322$. For Fig. 4.9b no prepositioning is required and thus the car is able to park with only three maneuvers and a final $||\mathbf{e}_2^t|| = 0.0286$.

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 [40] 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) [91] using its RRT* implementation for the planner. To account for the different planner (BiRRT* in [40] and RRT* in our implementation) the maximum planning time was set to 20 s instead of 6 s as presented in [40]. It was chosen for the HC paths to have zero curvature at the beginning and at the end (denoted as HC^{00}). The path planning results shown below were obtained on an AMD Ryzen 7 2700X at 4.15 GHz.

The initial poses of the cases presented in Figs. 4.10, 4.12 and 4.11 are, respectively,

the same as in Figs. 4.5a, 4.6a and 4.6b. All of them where tested in analogous simulated environments. Two different outcomes for each initial pose of the planned cases are shown: a desirable one (Figs. 4.10a, 4.11a and 4.12a) and an undesirable one (Figs. 4.10b, 4.11b and 4.12b). 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.

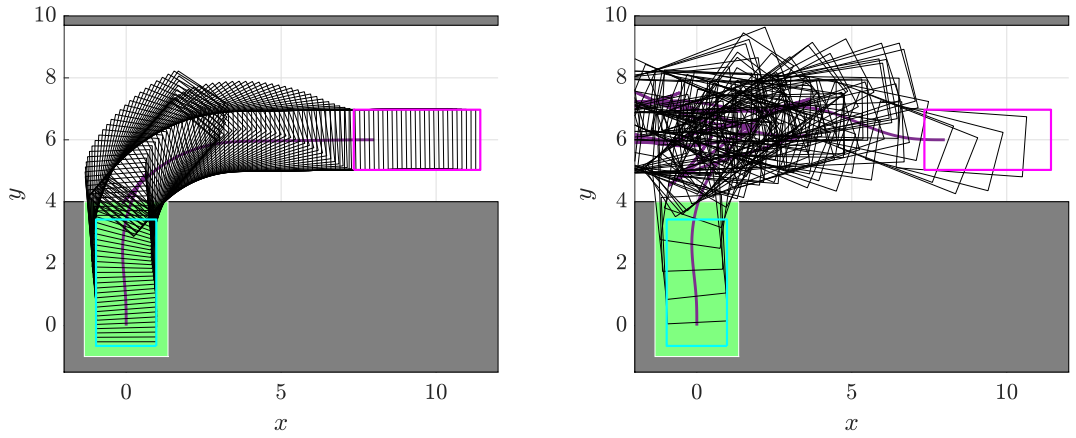


Figure 4.10: RRT* path planning results using HC^{00} steering function. Initial pose = (8m, 6m, 0°)

As shown in Fig. 4.10 the planned path could be rather good (Fig. 4.10a), leading to a visibly shorter path length than our MSBPC approch (Fig. 4.5a) but could also be completely undesirable (Fig. 4.10b). When considering the case where parking in a single maneuver is possible for our approach (Fig. 4.6a), one can notice that the desirable planned path (Fig. 4.11) also performs only one maneuver. In this case 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. 4.6b), the desirable planned path (Fig. 4.12) 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

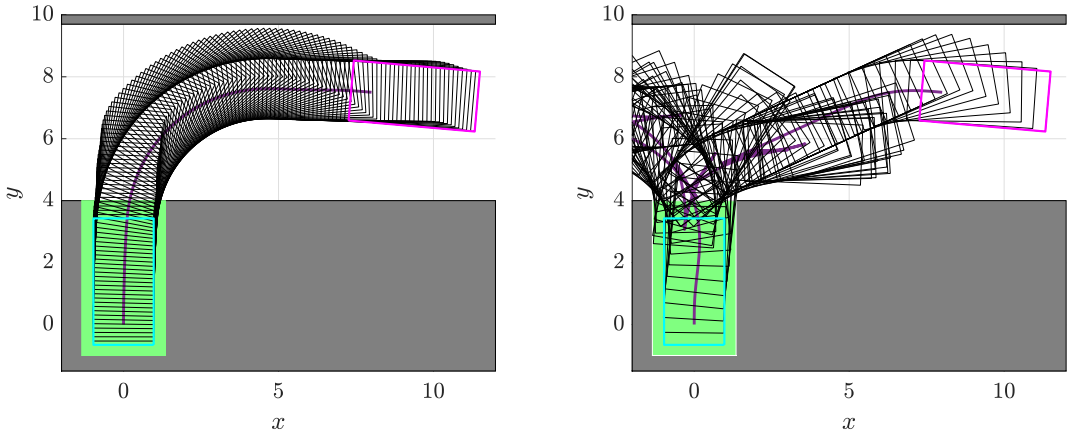


Figure 4.11: RRT* path planning results using HC^{00} steering function. Initial pose = (8m, 7.5m, −5°)

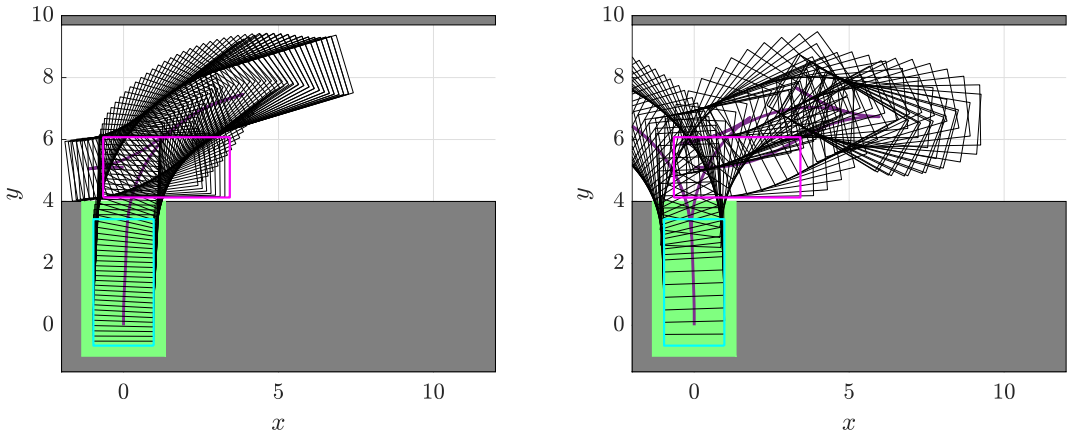
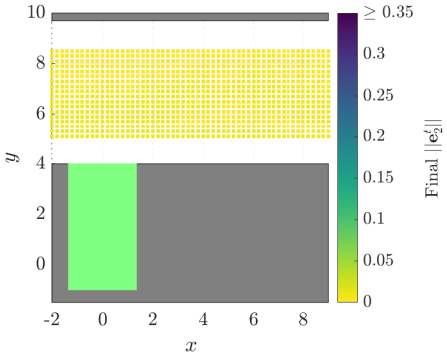


Figure 4.12: RRT* path planning results using HC^{00}-RS steering function. Initial pose = (0m, 5.1m, 0°)

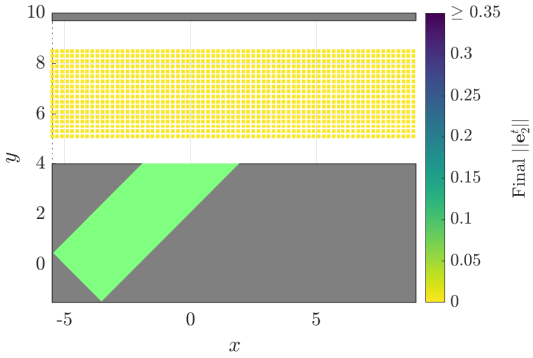
path and instead the controller runs online, there is (virtually) no inherent delay to be able to start a parking maneuver.

Exhaustive simulations

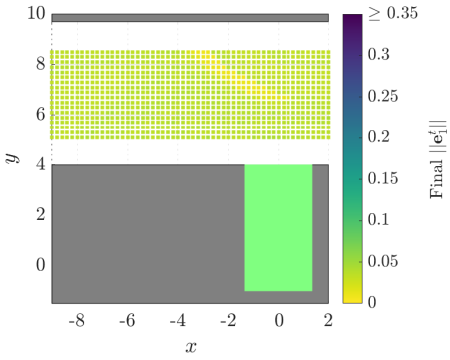
To address the stability and/or convergence concerns that might arise from the various constraints considered and their deactivation conditions as well as the use of finite control and prediction horizons, several convergence analyses for the different parking cases were



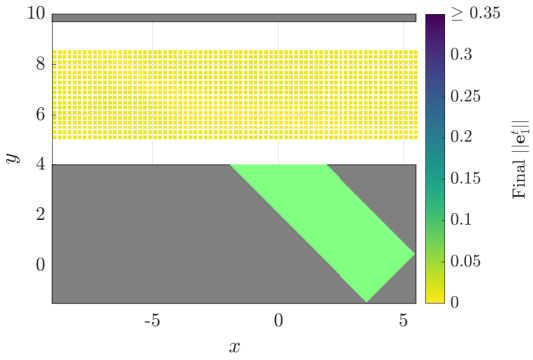
(a) Backward perpendicular case



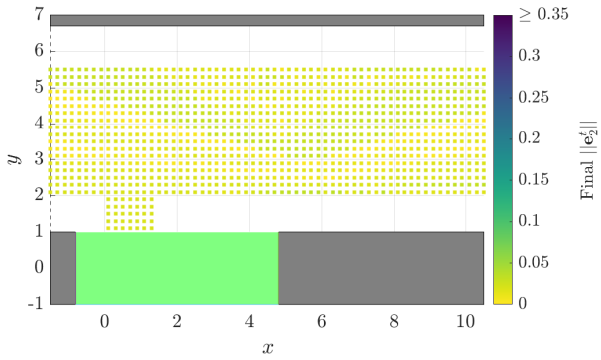
(b) Backward diagonal case



(c) Forward perpendicular case



(d) Forward diagonal case



(e) Backward parallel case

Figure 4.13: Exhaustive simulations for different types of parking maneuvers with a sampling step of 20 cm for the initial position. For non-parallel cases: spot length = 5 m and width = 2.7 m. For (e): spot length = 5.6 m and width = 2 m.

conducted by means of exhaustive simulations. To avoid over-cluttering the dissertation, for all the shown cases (Fig. 4.13), the initial orientation of the vehicle is equal to zero ($\theta_{T=0} = 0°$).

Since the exhaustive simulations are an aggregation of the results obtained from several simulations (like those shown in Figs. 4.5a,4.6a-4.9b), 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. 3.4.1) on the final value of $||\mathbf{e}_{\mathrm{m}}^{t}||$. As denoted by the colorbar, yellow/light green values of each scatter plot are inside the 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 $0.025 < ||\mathbf{e}_{\mathrm{m}}^{t}|| < 0.05$ as opposed to $||\mathbf{e}_{\mathrm{m}}^{t}|| < 0.015$ for the other non-parallel maneuvers. Backward parallel maneuvers are the second most challenging case often yielding larger final $||\mathbf{e}_{\mathrm{m}}^{t}||$ 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^t) error is responsible for the larger final $||\mathbf{e}_{\mathrm{m}}^{t}||$ values.

Considering moving obstacles

For illustrating the behavior of the vehicle when there are moving obstacles in the environment, the case shown in Fig. 4.5 has been considered as a starting point. Unless otherwise noted, the moving obstacle travels straight with a constant speed of $1\,{}^{\mathrm{m}}/_{\mathrm{s}}$ (expressed in the obstacle frame) and it is assumed to be known. It is denoted by a red circle and its orientation by a magenta arrow.

Starting with a situation where the moving obstacle travels from right to left, it can be seen that it doesn't allow the vehicle to steer away the parking spot initially (Figs. 4.14b-4.14d) and later doesn't let it steer into it (Fig. 4.14e). Once the obstacle has passed, the vehicle attempts to steer into the parking spot (Fig. 4.14f) without much success before changing its traveling direction. Due to this, the vehicle is forced to *restart* the parking attempt by driving away towards a better suited unparked pose. Five maneuvers (four changes of direction of motion) are required (Fig. 4.14a).

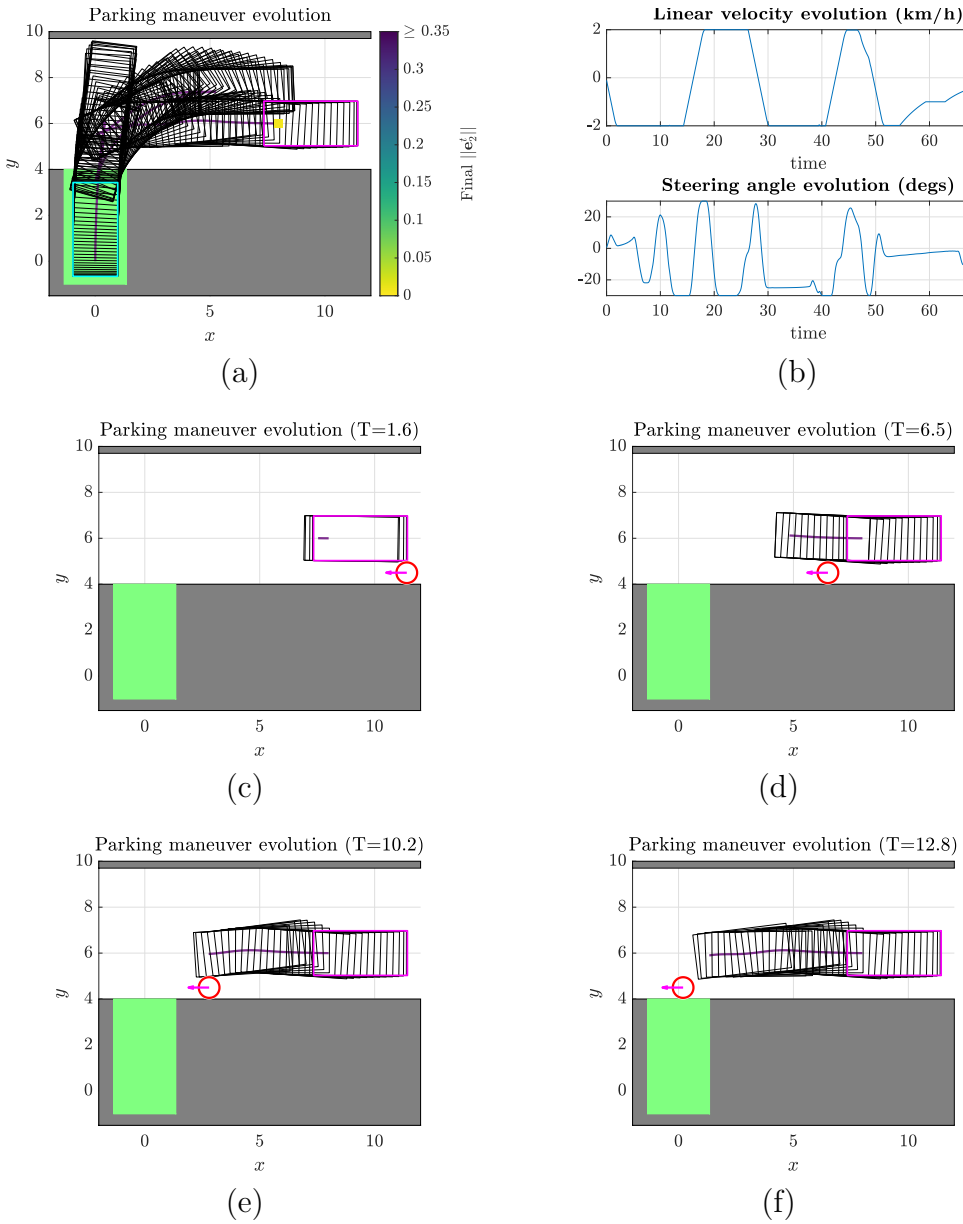


Figure 4.14: Constrained $\perp$ backward parking maneuver with a moving obstacle traveling from right to left

With the moving obstacle traveling from top to bottom in order to cross the road (Fig. 4.15c), the vehicle is forced to reduce its longitudinal velocity (Fig. 4.15b) when the obstacle is passing near its rear side (Figs. 4.15d and 4.15e). Once the obstacle has passed (Fig. 4.15f), the maneuvering continues mostly as in the original case requiring only three maneuvers (two changes of direction of motion) (Fig. 4.15a).

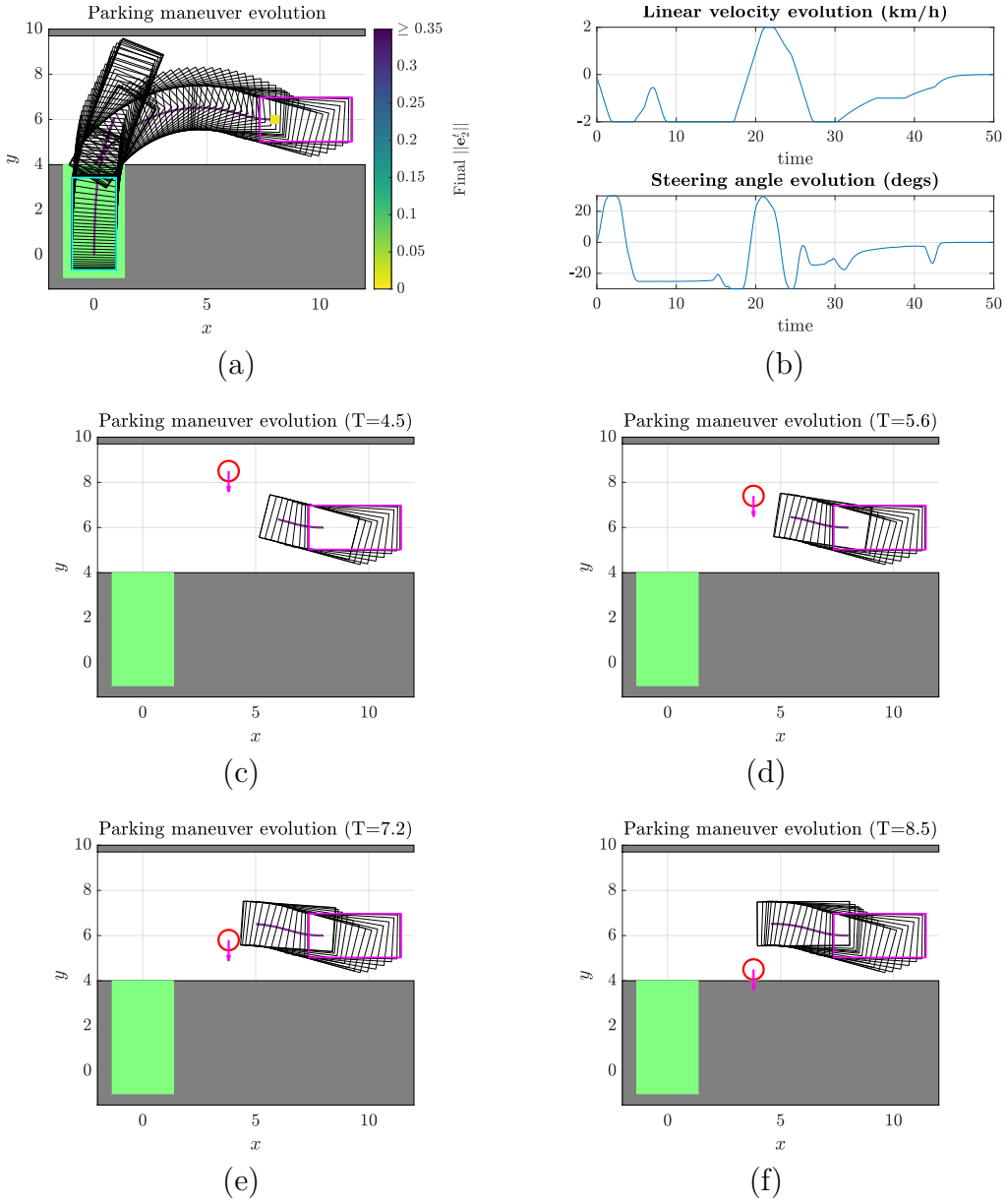


Figure 4.15: Constrained ⊥ backward parking maneuver with a moving obstacle traveling from top to bottom

If the obstacle is now traveling from left to right (Fig. 4.16c), the vehicle cannot steer away from the parking spot initially (Fig. 4.16b). While the moving obstacle is passing on the side of the vehicle (Figs. 4.16d and 4.16e), the motion of the steering angle is rather limited (Fig. 4.16b) but once the obstacle has passed (Fig. 4.16f), the maneuvering continues mostly as in the original case requiring only three maneuvers (two changes of

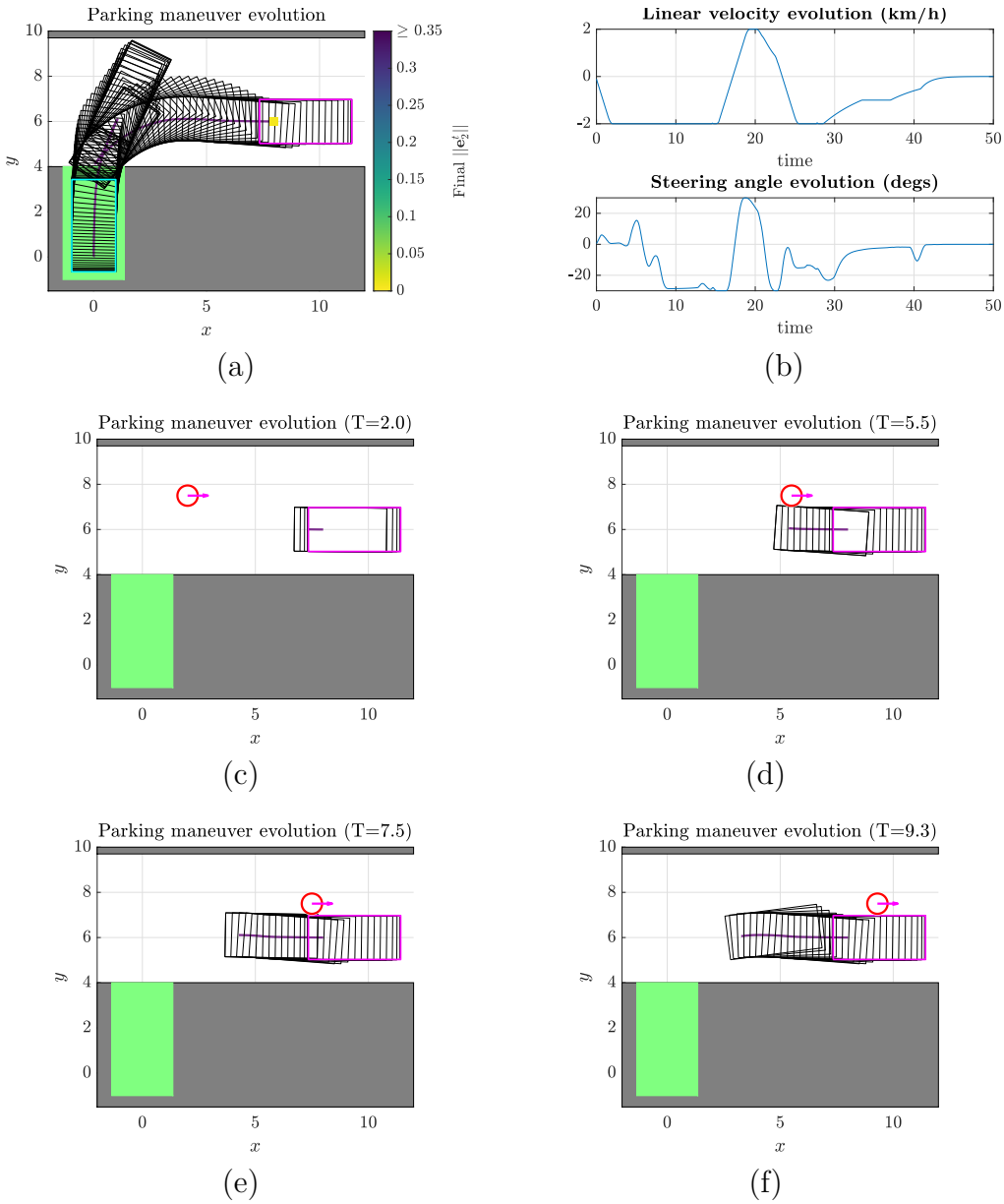


Figure 4.16: Constrained ⊥ backward parking maneuver with a moving obstacle traveling from left to right

direction of motion) (Fig. 4.16a).

For the case presented in Fig. 4.17, the pedestrian is initially static and remains like that until T=14.5 s, moment in which it starts moving from east to west with a constant speed of 1 m/s and stops once it is 1 m away from the left boundary of the parking spot. The lack of a magenta arrow in Figs. 4.17c and 4.17f indicate that the obstacle is not

moving.

The parking maneuver starts as in the original case but as the vehicle approaches the obstacle (Fig. 4.17c) it is forced to reduce its longitudinal velocity (Fig. 4.17b). Once the obstacle starts to move away (Fig. 4.17d and 4.17e) the vehicle is able to increase its longitudinal velocity up to the maximum allowed (Fig. 4.17b). When the obstacle is far

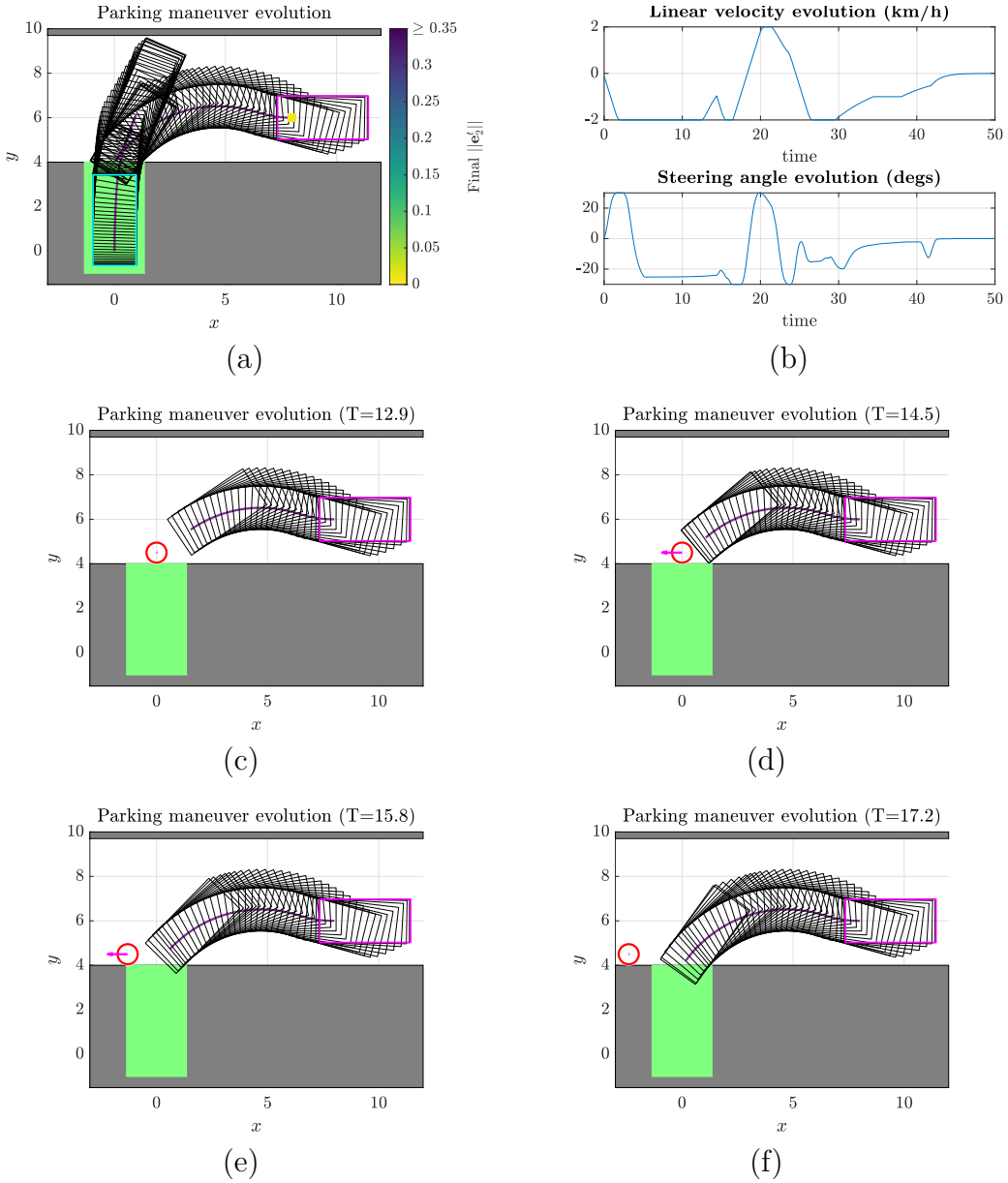


Figure 4.17: Constrained ⊥ backward parking maneuver with an obstacle blocking the entrance of the parking spot at the beginning

enough to the side (Fig. 4.17f) the maneuvering continues mostly as in the original case requiring only three maneuvers (two changes of direction of motion) (Fig. 4.17a).

For the case presented in Fig. 4.18, two pedestrians are considered, one static denoted by an orange circle (a *mother*) and one moving (a *child*) denoted by a red one. The moving obstacle travels in circles with a longitudinal velocity of 1.5 $^{\mathrm{m}}/_{\mathrm{s}}$ and, as the vehicle

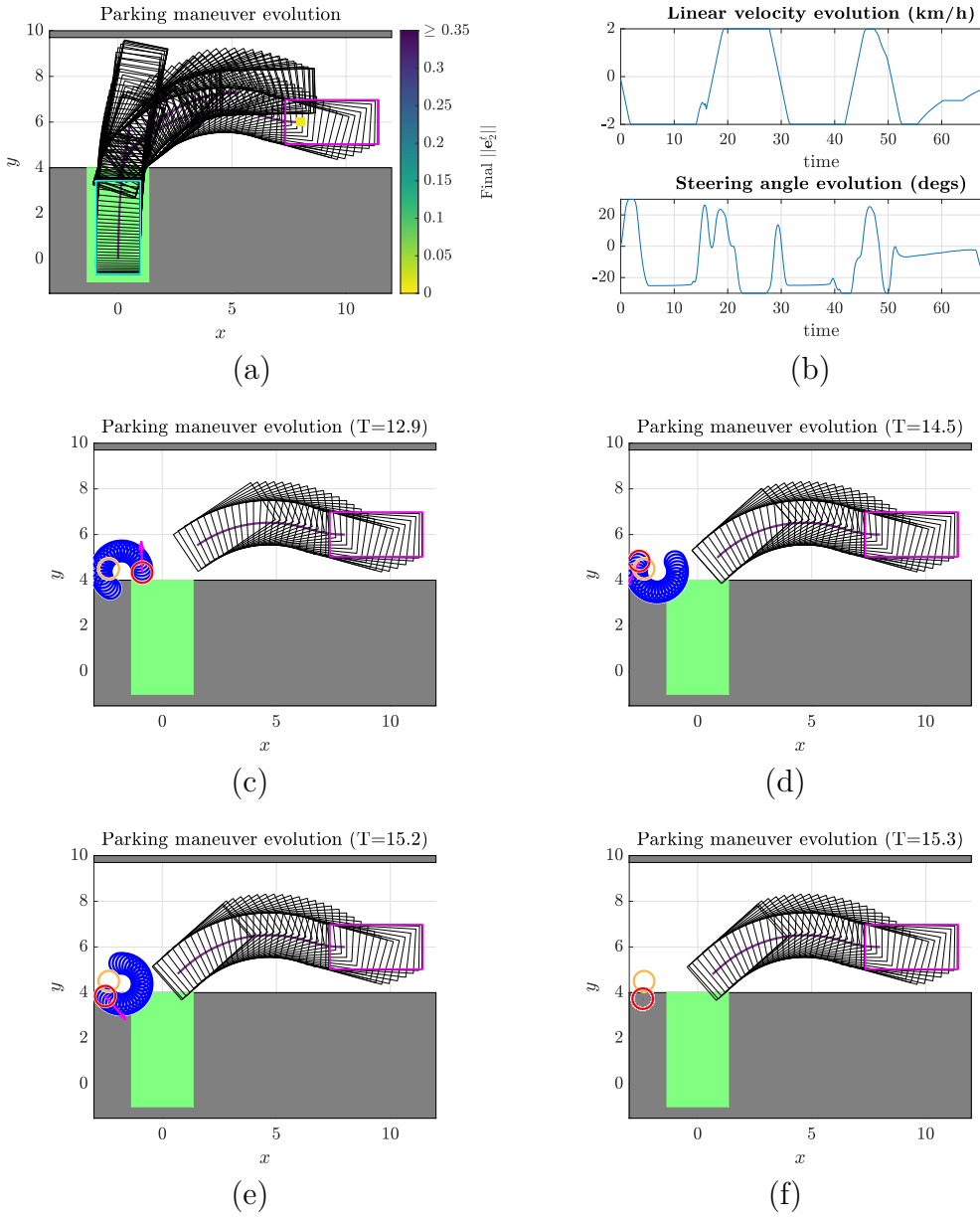


Figure 4.18: Constrained ⊥ backward parking maneuver with a *child* playing around the *mother*

approaches, it stops in a position that does not interfere with the parking maneuver. To better illustrate the *child's* motion, the future N_p positions are displayed with blue circles.

The parking maneuver starts as in the original case and, since the prediction model indicates that the *child* will move away from the vehicle's trajectory (Fig. 4.18c), contrary to the previous case, the vehicle does not reduce its longitudinal velocity as early (Fig. 4.18b) but as the obstacle starts to turn around towards the vehicle (Figs. 4.18d and 4.18e), the car is forced to both reduce its longitudinal velocity and modify its steering angle (Fig. 4.18b) in order to avoid collision. Once the *child* has stopped (Fig. 4.18f), the vehicle attempts to continue to steer into the parking spot for a brief moment before changing its direction of motion and *restart* the parking attempt. Regarding the *mother*, since it remains static the whole time in a position similar to the final one of the obstacle in the previous case, it does not interfere with the maneuvering. Finally, five maneuvers (four changes of direction of motion) are required (Fig. 4.14a) to park successfully.

4.4.2 Unparking simulation results

Individual cases - MATLAB simulations

To illustrate the performance of the proposed approach when unparking, a few examples are shown below. Similarly to the parking results, the final position is marked with a colored square whose color depends on the final value of $\|\mathbf{e}_1^t\|$.

It can be seen in Fig. 4.19a that the vehicle is able to reach the desired unparked pose in three maneuvers with smooth control signals (Fig. 4.19c). The (active) constraints imposed on $\mathbf{s}^c$ are satisfied at each time instant (Fig. 4.19f) ensuring a collision-free maneuver. It can be seen in Fig. 4.5b how the main task error $\mathbf{e}_1^t$ is minimized as the vehicle moves forward while $\mathbf{e}_2^t$ (Fig. 4.5d) is minimized with motions in the opposite direction. As expected, one can notice that when Q_2 is larger than Q_1 (Fig. 4.5e), the vehicle is moving backwards and when Q_1 is larger the opposite occurs. Regarding the main task weights, it can be seen how, similarly to the parking cases, the elements related to the orientation have little 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 those related to s_7^t and s_8^t try to make the vehicle vertical during the first $\approx$15 s and afterwards those related to to $s_1^t 0$ and s_{11}^t try to keep the vehicle horizontal. The vehicle completed the unparking maneuver with a final $\|\mathbf{e}_1^t\| = 0.0151$.

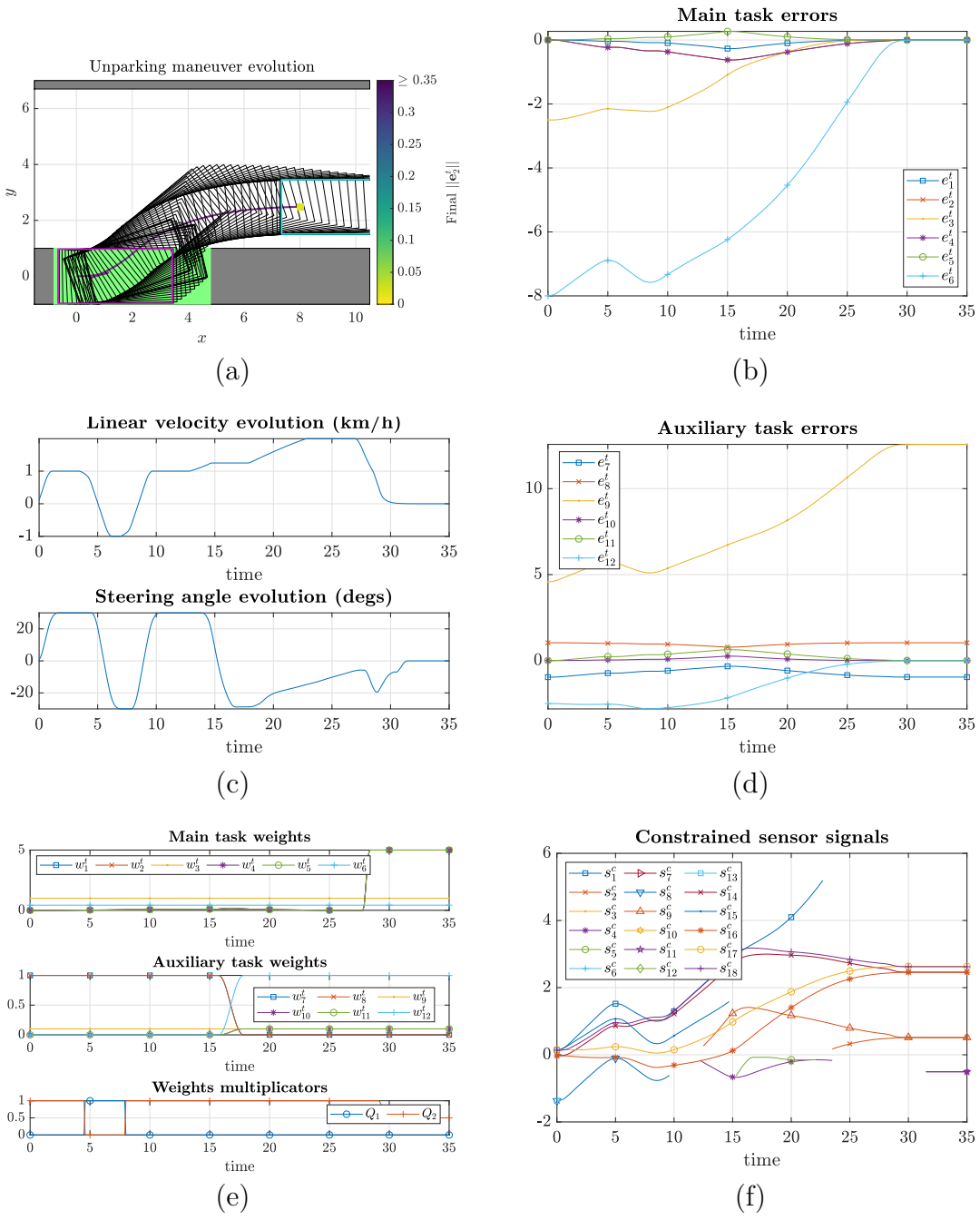
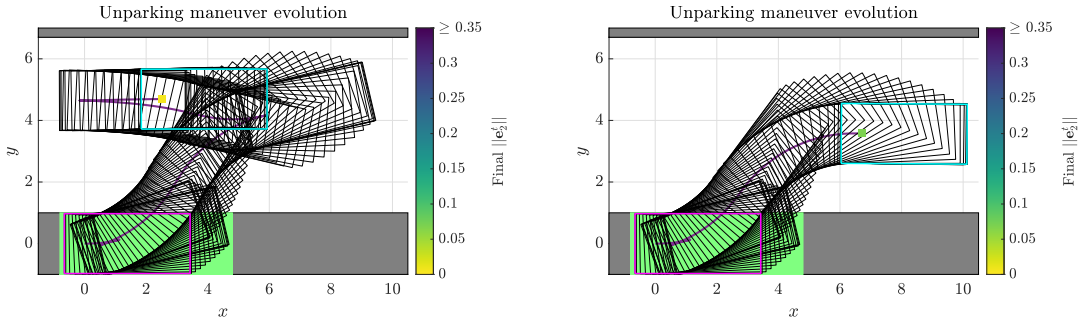


Figure 4.19: Constrained ‖ forward unparking maneuver signals: (a) performed maneuver, (c) control signals, (e) weighting-related signals, (b) main and (d) auxiliary task errors and (d) constrained sensor signals. Desired pose = (8 m, 2.5 m, 0°)

Two additional unparking maneuvers can be seen in Fig. 4.20. For the case presented in Fig. 4.20a, the vehicle has to perform five maneuvers (four changes of direction of



(a) (b)

Figure 4.20: Constrained ‖ forward unparking maneuvers. (a) Desired pose = (2.5 m, 4.7 m, 0°), (b) Initial pose = (6.7 m, 3.5 m, 0°)

motion) in order to reach the desired unparked pose, achieving a final $\|\mathbf{e}_1^t\| = 0.0068$. On the contrary, Fig. 4.20a shows a case where only three maneuvers are required with the caveat that the vehicle completes the maneuver with a final $\|\mathbf{e}_1^t\| = 0.0697$, being the lateral error the culprit. If a smaller final error was to be desired, it would be possible to modify certain parameters (e.g. ϵ_{m}, $\epsilon_{\mathcal{L}_1}$) in order to enforce a smaller final error at the expense of additional maneuvers.

Exhaustive simulations

When looking at the exhaustive simulation results (Fig. 4.21), one can clearly see that the vehicle is able to unpark virtually towards any desired position in the analysis window with no points out of the ROA. The final main task error is $\|\mathbf{e}_1^t\| < 0.05$ for the vast majority of desired poses although, as exemplified in one of the individual cases shown previously, for few desired poses the final error is $0.05 < \|\mathbf{e}_1^t\| < 0.075$. Given that in practice, after an unparking maneuver the vehicle would perform some kind of navigation task and that generally the obstacles would not be as near to the vehicle as when parking, the obtained results can be considered satisfactory.

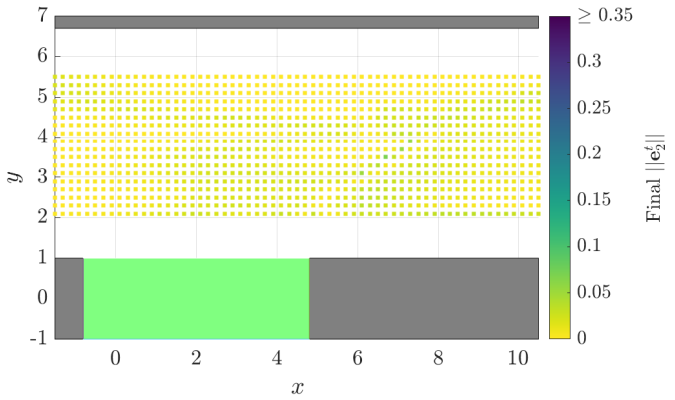


Figure 4.21: Exhaustive simulations for forward parallel unparking maneuvers with a sampling step of 20 cm for the initial position. Spot length = 5.6 m and width = 2 m

4.4.3 Real experimentation

Real experimentation was conducted for the following cases: backward perpendicular parking (Fig. 4.22), backward parallel parking (Fig. 4.24) and forward parallel unparking (Fig. 4.26). The controller is implemented in C++ using the solver NLopt [85] with a SLSQP algorithm [86] and runs at 10 Hz. The whole software architecture (and not only the parking/unparking controller) runs on a dual core Intel Core i5-3610ME.

To address the stability 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. 4.22). The estimated minimum and maximum values of roll and pitch (in degrees) were, respectively, $[-1.73, 1.87]$ and $[-1.94, 3.52]$.

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. 4.22, 4.23b) while satisfying the constraints during the whole maneuver (Fig. 4.23d) thus validating the results obtained from the corresponding convergence analysis. The final $||\mathbf{e}_2^t||$ achieved was 0.0408 which if reconstructed translates to errors of approximately 2.41 cm laterally, 3.25 cm longitudinally and $-0.22°$ in orientation.

The linear velocity setpoint and response can be seen in Fig. 4.23a. Because of the low-level controller, the setpoint given by the parking controller is not perfectly followed, especially when the vehicle passes over the speed bump. Nevertheless, the error generally remains small enough so that the car manages to park successfully.

Figure 4.22: Experimental car parking in a perpendicular spot

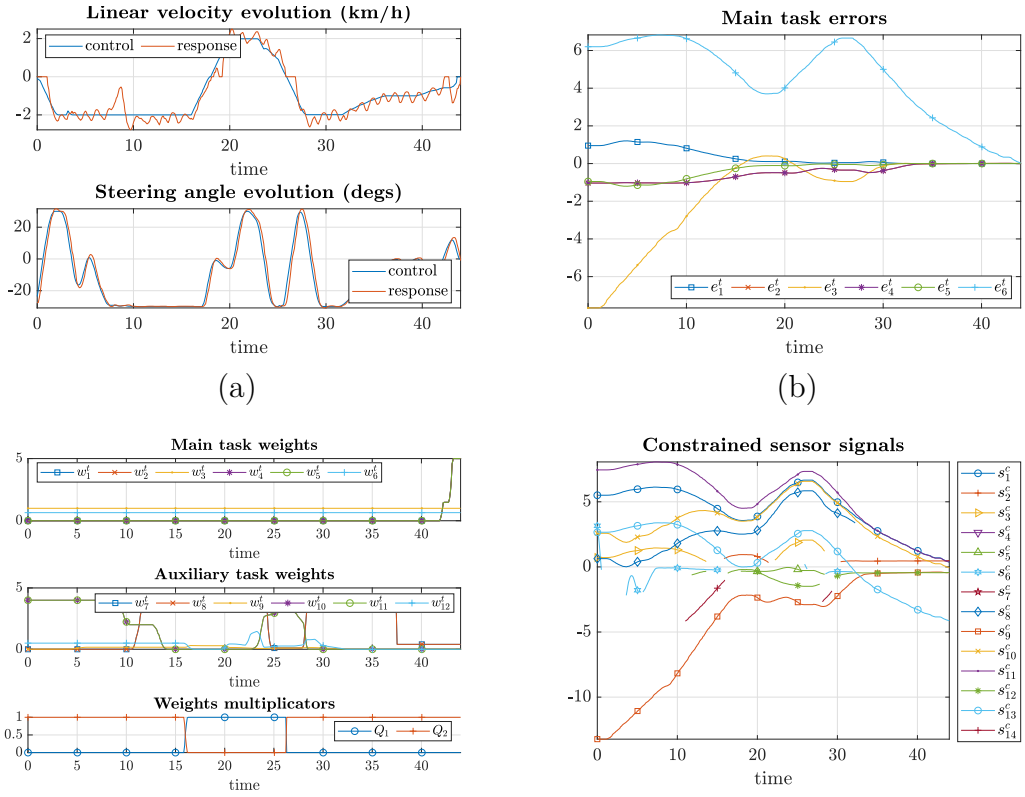
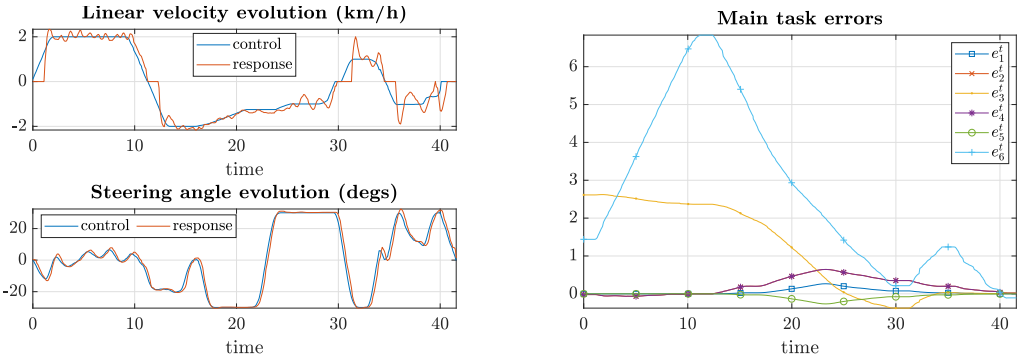


Figure 4.23: Constrained real backward ⊥ parking maneuver signals

Regarding the backward parallel parking maneuver (Fig. 4.24), it was chosen for the

vehicle to start the maneuver by moving forward in order to preposition itself, similarly to the simulated case shown in Fig. 4.9a, requiring three changes of direction (Fig. 4.25a) in order to complete the task successfully (Figs. 4.24, 4.25b). The final $||\mathbf{e}_2^t||$ achieved was 0.1121 which if reconstructed translates to errors of approximately 2.3 cm laterally, −10.67 cm longitudinally and 1.03° in orientation. The relatively large final longitudinal error can be easily explained by the spike on the longitudinal velocity response near the end of the maneuver (Fig. 4.25a).

Figure 4.24: Experimental car parking in a parallel spot



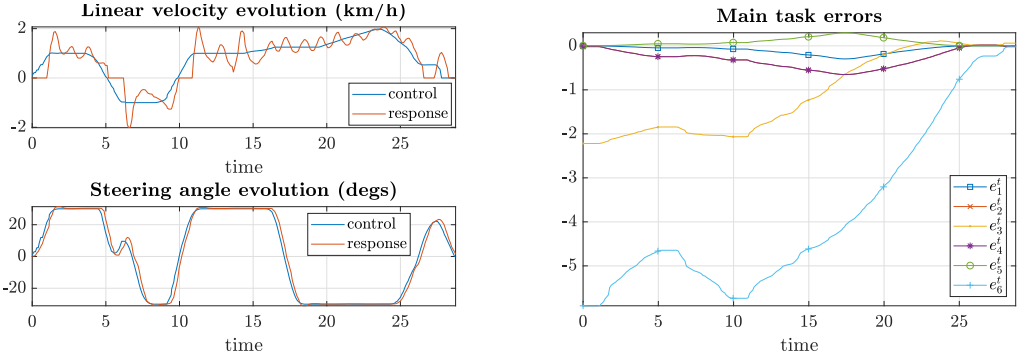
(a) (b)

Figure 4.25: Constrained real backward ‖ parking maneuver signals

The forward parallel unparking maneuver (Fig. 4.26) was started from the parked pose that the vehicle reached by itself. It can be seen that in this case only two changes of

direction were required (Fig. 4.27a) in order to complete the unparking task successfully (Figs. 4.26, 4.27b). The final $||\mathbf{e}_1^t||$ achieved was 0.0685 which if reconstructed translates to errors of approximately -2.47 cm laterally, 6.14 cm longitudinally and $-0.72°$ in orientation. Similarly to the parallel parking case, the relatively large final longitudinal error can be easily explained by the spike on the longitudinal velocity response near the end of the maneuver (Fig. 4.27a).

Figure 4.26: Experimental car unparking from a parallel spot



(a) (b)

Figure 4.27: Constrained real forward || unparking maneuver signals

It can be clearly seen that the less than ideal response of the longitudinal velocity affected negatively the final longitudinal error for both parking and unparking parallel maneuvers although, in spite of that, the final lateral and orientation errors are satisfactory.

4.5 Conclusions

As an extension of the MSBC by means of a predictive control strategy, here we have formalized under a single common framework five different parking cases: parking perpendicular and diagonal parking spots with both forward and backward motions and into parallel ones with backward motions. Additionally, it has been shown that with some minor modifications in the task definition, it is possible to perform unparking tasks using the same framework.

Most of the positive properties of the MSBC apply for the MSBPC one:

- 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.

- The differences on the exact definition of the constrained sensor features to use and on deactivation conditions are larger than for the task but the main idea remains the same across every case.

- By putting all the effort on 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.

Moreover, thanks to the convergence analyses performed, it has been proved that the presented MSBPC approach is stable and is able to converge from virtually any initial pose.

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 into a perpendicular spot with a satisfactory final $||\mathbf{e}_2^t|| = 0.0408$ while satisfying the constraints during the whole maneuver. Moreover, even if the final $||\mathbf{e}_\mathrm{p}^t||$ of the parallel parking and unparking maneuvers was not as satisfactory as for the perpendicular case, when reconstructing the error into its lateral, longitudinal and orientation components, one can clearly see that the culprit is the longitudinal error which is directly linked to the less than ideal response

of the longitudinal velocity - managed by an independent low-level velocity controller - while the values of the other components are satisfactory. On the one hand, the achieved results showcase the robustness of the approach against an unmodeled (erratic) behavior and on the other hand, an improved the low-level velocity controller should improve the final results. Furthermore, real experimentation shows that, in spite of the inherent computational complexity of predictive strategies, the presented approach can be executed online.

Additionally, simulation results with moving obstacles show that the presented approach is able to deal with reduction of free space provoked by the *pedestrians* in a variety of situations. This represents a key advantage when compared to path planning techniques, which typically would have to constantly replan the path in order to avoid collision - not to mention the associated computational cost, since our approach is able to react virtually instantaneously and take explicitly into account the pedestrian's motion in order to ensure collision-free maneuvers.

Conclusion

In spite of the interesting properties that sensor-based control techniques bring to the table and the fact that it has been proved effective in many different robotic applications, few work has been done on car-like robot applications, especially on maneuvering tasks. Given their properties, this type of techniques present an interesting alternative to the (seemingly omnipresent) path planning approaches typically used for Intelligent Vehicles applications. With the objective of developing a more advanced Intelligent Parking system than what is currently available, in this dissertation we have revisited different maneuvers typically performed in parking situations using Multi-Sensor-Based Control techniques.

Contributions

The first Multi-Sensor-Based Control (MSBC) framework presented in this dissertation allows to formalize under a single common framework the six different parking cases: perpendicular, diagonal and parallel for both forward and backward motions. It has been shown as well that the same framework allows to formalize unparking operations.

For both parking or unparking maneuvers, the definition of the task is rather similar, having just to consider slight adaptations on the interaction model for the different cases. Indeed, if one was interested in unconstrained parking/unparking maneuvers (e.g. because there are no obstacles in risk of collision), the differences among the different cases would be minimal.

As for the constraints imposed on certain sensor features in order to avoid collision, due to the nature of each of the maneuvers, the differences on the exact definition of the sensor features to use and on deactivation conditions are larger but the main idea remains the same across every case: avoid collision by imposing certain constraints on a set of sensor features (lines defining the boundaries of the parking spot, points at the corners of said spot, etc.).

Using a robotized Renault ZOE, it was shown that the MSBC approach is rather robust against sensor noise and external disturbances (e.g. bumpers). Additionally, we have proved that this approach does not inherently require any localization - i.e. the

approach is able to work even if no localization information is available, as long as the perception system is able to provide the required sensor features.

Nevertheless, since the MSBC framework is based on a classical multi-sensor-based approach, it is only able to perform one trial for parking/unparking - i.e it is not able to change the direction of (longitudinal) motion thus, depending on the initial (desired) pose of the vehicle with respect to the parking spot, the parking (unparking) maneuver might not be successful.

To overcome this critical limitation and to better account for the dynamics of the system, the MSBC approach was extended using a predictive control strategy. From this, a new Multi-Sensor-Based Predictive Control (MSBPC) framework was developed. For developing this second framework, most of what had been learned while developing the first one, particularly on the interaction model, task definition and constrained sensor features, was useful and/or transferable to some extent.

Using the MSBPC framework, we have formalized five different parking cases: parking perpendicular and diagonal parking spots with both forward and backward motions and into parallel ones with backward motions, in addition to unparking from parallel spots with forward motions. Furthermore, by means of exhaustive simulations, we have proved that the MSBPC is stable and is able to converge from (towards, when unparking) virtually any initial pose. Moreover we have shown that this approach is able to deal with reduction of free space provoked by moving obstacles in a variety of situations.

Furthermore, the presented approach has been validated by means of real experimentation, showing good robustness against external disturbances such as driving over a bump, or against model-plant mismatches such as the less than ideal longitudinal velocity response. Moreover, real experimentation shows that, in spite of the inherent computational complexity of predictive strategies, the presented approach can be executed online even in a (by today's standards) relatively weak computer.

Perspectives

Even if numerous positive outcomes have been obtained from the PhD thesis, there are still several tasks to do in order to have a complete level ≥ 4 Intelligent Parking system.

Formalization of different tasks Several parking-related operations are yet to be formalized using the MSBPC framework, most notably unparking maneuvers. Even if

non-parallel unparking maneuvers have been formalized under the MSBC framework, the added versatility and robustness that the more advanced MSBPC framework provides should be exploited. Moreover, a different type of maneuver but still related to parking operations: half turns, has yet to be formalized. Additionally, local navigation (to/from the parking spot/desired pose) using a MSBPC approach has yet to be studied.

Validation in dynamic environments Even if we have shown in simulation that the presented MSBPC approach is able to deal with moving obstacles, it is still necessary to validate that our experimental system does in fact reproduce what we have found in simulation. To this end, we are currently limited to some extent by the perception capabilities of our experimental vehicles.

Investigation of different modalities for avoiding moving obstacles Additionally to experimental validation of the proposed approach for dealing with moving obstacles, from a scientific point of view there is still work to be done. The behavior of the system when different types of sensor features are considered (alongside the associated constraint deactivation conditions) to model the moving obstacles should be investigated. Moreover, the possibility of considering such type of obstacles as part of the cost function (e.g. using potential fields) instead of as explicit constraints should be assessed. Furthermore, hybrid modalities could be explored as well.

Perception Following the previous idea, the perception capabilities of our vehicles, both in terms of physical sensors and algorithms, have to be considerably improved in order to have a more flexible and robust Intelligent Parking system. Perception setups that allow the vehicle to have information from all around, not only about obstacles but as well from signs, markings on the ground, etc., as well as the associated algorithms would be of great benefit.

Link to AI algorithms As evoked in the first chapter, promising works mixing MPC and AI algorithms are available in the literature, thus the possibility of linking AI algorithms to the framework presented in this dissertation should be investigated. On the one hand, one could consider using AI for learning certain parameters/functions (e.g. the dynamics of the vehicle, the weighting of the different sensor features, constrained sensor features and the associated deactivation conditions) to later be used in the presented approach, On the other hand, the possibility of using the presented approach to train a

model-based Artificial Intelligence (AI) algorithm that could then potentially be refined by using a hybrid model-based-model-free strategy, could be explored.

Appendices

Appendix A

Conditions for constraints deactivation (MSBC)

The constraints deactivation conditions used to obtain the results presented in this work are now detailed (Tables A.1-A.4). To simplify the content of the tables, the following notation is considered: $v_{\max}^{\text{abs}}$ 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 $-\phi_{\max}$), ${}^{i}p_a^{\text{Cart}}$ describes the point ${}^{i}p_a$ in Cartesian coordinates, the superscript $^{\text{c(angle)}}$ denotes a multiplication of the base by cos(angle) with angle expressed in degrees, act(*constraint*) is a boolean variable that is equal to true when the *constraint* is active and false otherwise and, ϵ_{long} and ϵ_{lat} are small positive values considered for constraints that are mostly related to, respectively, the longitudinal or lateral motions ($\epsilon_{\text{long}} = 0.05$ and $\epsilon_{\text{lat}} = 0.1$).

A.1 Parking

Table A.1: Constraints deactivation - backward non-parallel case

Constraint	*Deactivate if*
${}^3h_4$	${}^4h_4 \leq 0$
${}^3X_2$	${}^3Y_2 < -\epsilon_{\text{long}}$ or ${}^3X_2 < -2v_{\text{max}}^{\text{abs}}$
${}^3Y_2$	$({}^3h_4 > 0$ and ${}^4h_4 > 0)$ or $\text{act}({}^3d_{lat_2})$
${}^3d_{lat_2}$	$\phi \geq 0$ or $\text{act}({}^6d_3)$ or $!({}^3Y_2 > -\rho^{\text{c45}}_{m_{\text{min}^+}}$ and ${}^3X_2 < l_{ro}$ and $!(\lvert {}^3h_4 - {}^4h_4\rvert < l_{ve}^{\text{c30}}$ and ${}^3X_2 < -\rho^{\text{c45}}_{m_{\text{min}^+}}))$
${}^6h_3$	$\text{act}({}^6d_3)$ or ${}^3Y_2 < -\epsilon_{\text{lat}}$ or ${}^6X_3 < -2v_{\text{max}}^{\text{abs}}$ or $({}^6X_3 < 0$ and ${}^6Y_3 > 0)$
${}^6X_3$	$\text{act}({}^6d_3)$ or ${}^3Y_3 < -\epsilon_{\text{long}}$ or ${}^6Y_3 > \epsilon_{\text{long}}$ or ${}^6X_3 < -2v_{\text{max}}^{\text{abs}}$
${}^6Y_3$	${}^6Y_3 < 0$ or $(\lVert {}^6p_3^{\text{Cart}}\rVert > \rho^{\text{c75}}_{m_{\text{min}^+}}$ and ${}^6X_3 < 0)$
${}^6d_3$	$\phi \geq 0$ or $!({}^4h_4 > (l_{ro} - l_{ve})^{\text{c60}}$ and ${}^6Y_3 < \epsilon_{\text{lat}}$ and ${}^6X_3 < 0)$

Table A.2: Constraints deactivation - forward non-parallel case

Constraint	*Deactivate if*
${}^4h_3$	${}^4X_3 > 2v_{\text{max}}^{\text{abs}}$ or $(\text{sign}({}^4Y_2)\text{sign}({}^4Y_3)) > 0$
${}^4X_3$	${}^4Y_3 < -\epsilon_{\text{long}}$ or ${}^5Y_3 > \epsilon_{\text{long}}$ or ${}^4X_3 > 2v_{\text{max}}^{\text{abs}})$
${}^4Y_3$	${}^4X_3 > \rho^{\text{c80}}_{m_{\text{min}^+}}$ or ${}^4Y_3 < -\rho^{\text{c60}}_{m_{\text{min}^+}}$
${}^5h_4$	${}^5X_2 > \epsilon_{\text{lat}}$
${}^5Y_2$	${}^5X_2 > \rho^{\text{c80}}_{m_{\text{min}^+}}$ or ${}^5Y_2 > \rho^{\text{c60}}_{m_{\text{min}^+}}$ or $({}^5X_2 > \epsilon_{\text{lat}}$ and ${}^5Y_2 < \epsilon_{\text{lat}})$
${}^5d_2$	$!(\phi < 0$ and ${}^5d_{2_{\text{min}^-}} > 2v_{\text{max}}^{\text{abs}}$ and $\lVert {}^5p_2^{\text{Cart}}\rVert < \rho^{\text{c30}}_{m_{\text{min}^+}})$

Table A.3: Constraints deactivation - backward parallel case

Constraint	*Deactivate if*
${}^3X_2$	${}^3Y_2 < -\epsilon_{\text{long}}$ or ${}^3X_2 < -2v_{\text{max}}^{\text{abs}}$ or ${}^4X_2 > 0$
${}^3Y_2$	${}^3X_2 > l_{ro}$
${}^3d_{lat_2}$	$!(\phi < 0$ and ${}^3X_2 < l_{ro}$ and ${}^3Y_2 < 0)$ or $\text{act}({}^4d_2)$
${}^4Y_2$	${}^4X_2 > 0$
${}^4d_2$	$\text{act}({}^3d_{lat_2})$ or $!(\phi > 0$ and ${}^4X_2 < 0$ and ${}^3X_2 > 0)$
${}^4Y_4$	${}^4X_2 > 0$ or ${}^3h_4 > l_{ro}$ or ${}^3Y_2 < -\rho^{\text{c80}}_{m_{\text{min}^+}}$

Table A.4: Constraints deactivation - forward parallel case

Constraint	*Deactivate if*
${}^3d_{lat_3}$	$\phi \geq 0$ or ${}^3X_3 < 0$
${}^4X_3$	${}^4Y_3 < -\epsilon_{\text{long}}$ or ${}^4X_3 > 2v_{\text{max}}^{\text{abs}}$ or ${}^3X_3 < 0$
${}^4Y_3$	$\text{act}({}^3d_{lat_3})$

A.2 Unparking

Table A.5: Constraints deactivation conditions - forward non-parallel case

Constraint	*Deactivate if*
${}^{3}d_{lat_2}$	$!(\phi < 0$ and ${}^{3}X_2 > -x_i)$
${}^{3}Y_2$	act$({}^{3}d_{lat_2})$ or ${}^{3}h_5 > 0$
${}^{6}h_3$	${}^{6}X_3 < 0$ or ${}^{6}h_5 > 0$
${}^{6}Y_3$	act$({}^{6}d_{lat_3})$ or ${}^{6}X_3 < 0$ or ${}^{6}h_5 > 0$
${}^{6}d_{lat_3}$	$!(\phi > 0$ and ${}^{6}X_3 > -x_i)$

Table A.6: Constraints deactivation conditions - backward non-parallel case

Constraint	*Deactivate if*
${}^{4}h_3$	${}^{4}h_5 > 0$
${}^{4}Y_3$	${}^{4}h_5 > 0$
${}^{5}h_4$	${}^{5}X_2 > 0$ or ${}^{5}h_5 > 0$
${}^{5}Y_2$	!act$({}^{5}h_4)$ or ${}^{6}h_4 > {}^{5}h_4$

Appendix B

Conditions for constraints deactivation (MSBPC)

The constraints deactivation conditions used to obtain the results presented in this work are now detailed (Tables B.1-B.4). 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(angle) with angle expressed in degrees and 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.

Table B.1: Constraints deactivation - backward non-parallel case

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

Table B.2: Constraints deactivation - forward non-parallel case

Constraint	*Deactivate if*
${}^{3}h_{5}^{c^-}$	${}^{3}h_4 > 0$ and ${}^{3}h_3 < 0$ and ${}^{3}h_5 > {}^{4}h_5$
${}^{4}h_{2}^{c^-}$	–
${}^{4}h_{3}^{c^+}$	${}^{4}X_3 > 2v_{\max}^{\text{abs}}$ or $(\text{sign}({}^{4}Y_2)\text{sign}({}^{4}Y_3)) > 0$ or ${}^{4}h_5 > 0$
${}^{4}h_{5}^{c^-}$	${}^{4}h_4 > \epsilon_1$ and ${}^{4}h_3 < -\epsilon_1$
${}^{4}X_{3}^{c^-}$	${}^{4}Y_3 < 0$ or ${}^{5}Y_3 > \epsilon_1$ or ${}^{4}X_3 > 2v_{\max}^{\text{abs}}$ or ${}^{3}X_3 < -\epsilon_1$
${}^{4}Y_{3}^{c^+}$	${}^{4}X_3 > 2v_{\max}^{\text{abs}}$ or ${}^{3}h_4 > \epsilon_1$ or ${}^{3}h_5 < {}^{4}h_5$
${}^{5}h_{2}^{c^-}$	–
${}^{5}h_{4}^{c^-}$	act $\left({}^{5}d_{2}^{c^-}\right)$ or ${}^{5}h_5 > 0$ or ${}^{5}X_2 > 0$
${}^{5}h_{5}^{c^+}$	–
${}^{5}X_{2}^{c^-}$	${}^{5}Y_2 > \epsilon_1$ or ${}^{4}Y_2 < -\epsilon_1$ or ${}^{5}X_2 > 2v_{\max}^{\text{abs}}$ or ${}^{6}X_2 < -\epsilon_1$ or $({}^{3}h_3 < 0$ and ${}^{3}h_5 < \epsilon_1$ and ${}^{3}h_5 < {}^{4}h_5)$
${}^{5}Y_{2}^{c^+}$	${}^{4}h_4 > 0$ or ${}^{5}X_2 > 2v_{\max}^{\text{abs}}$ or ${}^{6}X_2 < -\epsilon_1$
${}^{5}d_{2}^{c^-}$	$v_{x_m} \leq 0$ or $\phi \geq 0$ or ${}^{4}Y_3 \geq -\epsilon_3$ or ${}^{5}d_{2_{\min^-}} \leq \epsilon_3$ or ${}^{5}h_5 \leq 0$ or ${}^{5}h_1 \geq 0$ or ${}^{5}h_5 \geq {}^{5}\rho_{5_{\min^-}}^{\text{c75}} - 0.75v_{\max}^{\text{abs}}$ or ${}^{m}h_1 \geq 0$
${}^{6}h_{5}^{c^+}$	–

Table B.3: Constraints deactivation - backward parallel parking / forward parallel unparking

Constraint	*Deactivate if*
${}^{3}h_{2}^{c^-}$	${}^{3}h_5 > \epsilon_2$
${}^{3}h_{5}^{c^-}$	${}^{3}h_2 > \epsilon_1$ and ${}^{3}h_4 < -\epsilon_1$
${}^{3}X_{3}^{c^+}$	${}^{3}Y_3 < -\epsilon_1$ or ${}^{6}Y_3 > \epsilon_1$ or ${}^{3}X_3 < -2v_{\max}^{\text{abs}}$ or ${}^{4}X_3 > 0$ or ${}^{4}h_4 < 0$ or act $\left({}^{3}h_{5}^{c^-}\right)$
${}^{3}Y_{3}^{c^+}$	${}^{3}Y_3 > 0$ or $({}^{3}X_3 > \lvert x_3\rvert$ and $v_{x_m} < 0)$ or ${}^{4}X_3 > 0$ or act $\left({}^{4}X_{3}^{c^-}\right)$ or $(\lvert\phi\rvert > 0$ and $({}^{3}h_5 > 0$ and ${}^{3}h_4 < 0))$
${}^{3}d_{lat_3}^{c^+}$	$\phi \geq 0$ or ${}^{3}Y_3 > -\epsilon_2$ or $(v_{x_m} \leq 0$ and ${}^{3}X_3 > \lvert x_3\rvert)$ or $(v_{x_m} \geq 0$ and $({}^{3}X_3 < \lvert x_3\rvert$ or ${}^{3}h_5 > \epsilon_2))$
${}^{3}X_{4}^{c^+}$	$v_{x_m} \geq 0$ or ${}^{3}Y_4 < -\epsilon_1$ or ${}^{6}Y_4 > \epsilon_1$
${}^{3}Y_{4}^{c^+}$	${}^{4}h_5 > \epsilon_1$ or ${}^{3}h_5 < {}^{4}h_5$ or $({}^{3}h_5 < \epsilon_2$ and ${}^{4}h_5 < \epsilon_2)$
${}^{4}h_{4}^{c^+}$	${}^{4}h_5 > \epsilon_2$
${}^{4}h_{5}^{c^-}$	${}^{4}h_4 < 0$ and ${}^{4}h_2 > 0$ and $(({}^{3}h_5 < {}^{4}h_5$ and ${}^{4}h_5 > 0)$ or ${}^{4}h_5 < 0)$
${}^{4}X_{3}^{c^-}$	act $\left({}^{3}X_{3}^{c^+}\right)$ or ${}^{4}Y_3 < -\epsilon_1$ or ${}^{5}Y_3 > \epsilon_1$ or ${}^{4}X_3 > 2v_{\max}^{\text{abs}}$ or ${}^{6}h_5 < 0$ or ${}^{3}h_4 > 0$ or ${}^{3}X_3 < 0$
${}^{4}Y_{3}^{c^+}$	${}^{4}Y_3 > 0$ or ${}^{4}h_4 < 0$ or $\lvert\phi\rvert > 0$
${}^{4}d_{3}^{c^+}$	$\phi \leq 0$ or $v_{x_m} > 0$ or ${}^{4}X_3 > \epsilon_3$ or ${}^{3}h_5 > \epsilon_2$ or ${}^{3}h_4 > -\lvert x_3\rvert$ or ${}^{4}h_5 < 0$
${}^{5}h_{4}^{c^+}$	$v_{x_m} < 0$ or ${}^{5}h_5 > 0$
${}^{5}h_{5}^{c^+}$	–
${}^{6}h_{2}^{c^-}$	${}^{3}h_5 > \epsilon_2$ or ${}^{6}h_5 > \epsilon_2$
${}^{6}h_{5}^{c^+}$	–
${}^{7}h_{3}^{c^-}$	–
${}^{8}h_{3}^{c^-}$	–

Table B.4: Constraints deactivation - moving obstacle

Constraint	*Deactivate if*
${}^{3}X_{\text{mo}}^{c^+}$	${}^{3}X_{\text{mo}} > 0$ or ${}^{3}Y_{\text{mo}} < -\epsilon_4$ or ${}^{5}Y_{\text{mo}} > \epsilon_4$
${}^{3}Y_{\text{mo}}^{c^+}$	${}^{3}Y_{\text{mo}} > 0$ or ${}^{3}X_{\text{mo}} < -\rho_{m_{\text{min}^+}} - 1.5v_{\text{max}}^{\text{abs}}$ or ${}^{4}X_{\text{mo}} > \rho_{m_{\text{min}^+}} - 1.5v_{\text{max}}^{\text{abs}}$ or $({}^{3}X_{\text{mo}} < -\epsilon_4$ and $({}^{m}v_{x_{\text{mo}}} \leq 0$ or $\lvert {}^{m}v_{y_{\text{mo}}}\rvert > \lvert {}^{m}v_{x_{\text{mo}}}\rvert))$ or $({}^{4}X_{\text{mo}} > \epsilon_4$ and $({}^{m}v_{x_{\text{mo}}} \geq 0$ or $\lvert {}^{m}v_{y_{\text{mo}}}\rvert > \lvert {}^{m}v_{x_{\text{mo}}}\rvert))$ or $({}^{3}X_{\text{mo}} > -\epsilon_5$ and ${}^{4}X_{\text{mo}} < \epsilon_5$ and ${}^{3}Y_{\text{mo}} > -\epsilon_5$ and ${}^{6}Y_{\text{mo}} < \epsilon_5)$
${}^{4}X_{\text{mo}}^{c^-}$	${}^{4}X_{\text{mo}} < 0$ or ${}^{3}Y_{\text{mo}} < -\epsilon_4$ or ${}^{5}Y_{\text{mo}} > \epsilon_4$
${}^{4}Y_{\text{mo}}^{c^+}$	!act $\left({}^{3}Y_{\text{mo}}^{c^+}\right)$
${}^{5}X_{\text{mo}}^{c^-}$	!act $\left({}^{4}X_{\text{mo}}^{c^-}\right)$
${}^{5}Y_{\text{mo}}^{c^-}$	${}^{5}Y_{\text{mo}} < 0$ or ${}^{3}X_{\text{mo}} < -\rho_{m_{\text{min}^+}} - 1.5v_{\text{max}}^{\text{abs}}$ or ${}^{4}X_{\text{mo}} > \rho_{m_{\text{min}^+}} - 1.5v_{\text{max}}^{\text{abs}}$ or $({}^{3}X_{\text{mo}} < -\epsilon_4$ and $({}^{m}v_{x_{\text{mo}}} \leq 0$ or $\lvert {}^{m}v_{y_{\text{mo}}}\rvert > \lvert {}^{m}v_{x_{\text{mo}}}\rvert))$ or $({}^{4}X_{\text{mo}} > \epsilon_4$ and $({}^{m}v_{x_{\text{mo}}} \geq 0$ or $\lvert {}^{m}v_{y_{\text{mo}}}\rvert > \lvert {}^{m}v_{x_{\text{mo}}}\rvert))$ or $({}^{3}X_{\text{mo}} > -\epsilon_5$ and ${}^{4}X_{\text{mo}} < \epsilon_5$ and ${}^{3}Y_{\text{mo}} > -\epsilon_5$ and ${}^{6}Y_{\text{mo}} < \epsilon_5)$
${}^{6}X_{\text{mo}}^{c^+}$	!act $\left({}^{3}X_{\text{mo}}^{c^+}\right)$
${}^{6}Y_{\text{mo}}^{c^-}$	!act $\left({}^{5}Y_{\text{mo}}^{c^-}\right)$

Bibliography

[1] A. Khalid, *Ford CEO says the company 'overestimated' self-driving cars*, 2019. [Online]. Available: https://www.engadget.com/2019/04/10/ford-ceo-says-the-company-overestimated-self-driving-cars/ (visited on 09/05/2019).

[2] L. Eliot, *Tesla Autopilot's Enhanced Summon Is A Nice Concept That Opens A Can Of Worms*, 2019. [Online]. Available: https://www.forbes.com/sites/lanceeliot/2019/06/13/tesla-autopilots-enhanced-summon-is-a-nice-concept-that-opens-a-can-of-worms/ (visited on 09/05/2019).

[3] International Parking Insitute, "2012 Emerging Trends in Parking", Tech. Rep., 2012. [Online]. Available: http://www.parking.org/media/129563/2012%20emergingtrends%20in%20parking%20final.pdf.

[4] Transportation Alternatives, "No Vacancy: Park Slope's Parking Problem and how to fix it", 2007.

[5] E. Gantelet and A. Lefauconnier, "The time looking for a parking space: Strategies, associated nuisances and stakes of parking management in France", *in European Transport Conference*, Association for European Transport, 2006, pp. 1–7.

[6] W. Wang, Y. Song, J. Zhang, and H. Deng, "Automatic parking of vehicles: A review of literatures", *International Journal of Automotive Technology*, vol. 15, *6*, pp. 967–978, 2014.

[7] M. Seiter, H.-J. Mathony, and P. Knoll, "Parking Assist", *in Handbook of Intelligent Vehicles*, 1, A. Eskandarian, Ed., London: Springer London, 2012, pp. 829–864.

[8] Volvo Car Group, *Autonomous Parking - Volvo Cars Innovations*, 2013. [Online]. Available: https://www.youtube.com/watch?v=v0EbNYZhJEo (visited on 12/20/2015).

[9] C. Samson, B. Espiau, and M. Le Borgne, *Robot Control: The Task Function Approach*. Oxford University Press, 1991.

[10] SAE, "SAE J3016™: Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems", SAE International, Tech. Rep., 2018.

[11] OICA, *Automated Driving Definition for Levels of Automation*, 2014. [Online]. Available: http://www.oica.net/wp-content/uploads//WP-29-162-20-OICA-automated-driving.pdf.

[12] CNN.com, *Toyota unveils car that park itself*, 2003. [Online]. Available: http://edition.cnn.com/2003/TECH/ptech/09/01/toyota.prius.reut/index.html (visited on 01/13/2016).

[13] PSA Peugeot Citroën, *Fully automatic parking, a reality from 2017*. [Online]. Available: http://www.psa-peugeot-citroen.com/en/featured-content/connected-car/city-park-full-automatic (visited on 01/13/2016).

[14] S. Kane, *2012 Family Cars With Self-Parking Technology*, 2011. [Online]. Available: http://www.thecarconnection.com/news/1067819%7B%5C_%7D2012-family-cars-with-self-parking-technology (visited on 01/13/2016).

[15] J. Healey, *Which cars park themselves best? Challenge results*, 2012. [Online]. Available: http://www.usatoday.com/story/money/cars/2012/12/06/self-parking-cars-challenge/1743199/ (visited on 01/13/2016).

[16] BMW, *STRESS-FREE PARKING*. [Online]. Available: http://www.bmw.com/com/en/insights/technology/connecteddrive/2013/driver%7B%5C_%7Dassistance/intelligent%7B%5C_%7Dparking.html (visited on 01/13/2016).

[17] B. Lee, *Tesla Autopark Demo*, 2015. [Online]. Available: https://www.youtube.com/watch?v=6LBPYT-P14Y (visited on 01/13/2016).

[18] M. Prigg, *The incredible self-parking car that can even come and meet you when you're done shopping*. [Online]. Available: http://www.dailymail.co.uk/sciencetech/article-2259558/Audi-A7-The-incredible-self-parking-car-come-meet-youre-shopping.html (visited on 01/13/2016).

[19] A. English, *The end of parking prangs*. [Online]. Available: http://www.telegraph.co.uk/cars/news/the-end-of-parking-prangs/ (visited on 01/13/2016).

[20] The Tesla Motors Team, *Summon Your Tesla from Your Phone*. [Online]. Available: https://www.teslamotors.com/blog/summon-your-tesla-your-phone (visited on 01/13/2016).

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

[22] M. Marouf, E. Pollard, and F. Nashashibi, "Automatic parallel parking and platooning to redistribute electric vehicles in a car-sharing application", *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 486–491, 2014.

[23] C. Laugier and I. E. Paromtchik, "Autonomous Parallel Parking of a non Holomonic Vehicle", *in Intelligent Vehicles Symposium*, 1996, pp. 13–18.

[24] Jaeyoung Moon, I. Bae, Jae-gwang Cha, and Shiho Kim, "A trajectory planning method based on forward path generation and backward tracking algorithm for Automatic Parking Systems", *in 17th International IEEE Conference on Intelligent Transportation Systems*, 2014, pp. 719–724.

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

[26] H. Vorobieva, S. Glaser, N. Minoiu-Enache, and S. Mammar, "Automatic parallel parking in tiny spots: Path planning and control", *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, *1*, pp. 396–410, 2015.

[27] F. Cuesta, F. Gomez-Bravo, and A. Ollero, "A Combined Planned/Reactive System for Motion Control of Vehicles Manoeuvres", *IFAC Proceedings Volumes*, vol. 31, *27*, pp. 283–288, Sep. 1998.

[28] K. Demirli and M. Khoshnejad, "Autonomous parallel parking of a car-like mobile robot by a neuro-fuzzy sensor-based controller", *Fuzzy Sets and Systems*, vol. 160, *19*, pp. 2876–2891, Oct. 2009.

[29] H. Grimmett, M. Buerki, L. Paz, P. Pinies, P. Furgale, I. Posner, and P. Newman, "Integrating Metric and Semantic Maps for Vision-Only Automated Parking", *International Conference on Robotics and Automation*, pp. 2159–2166, 2015.

[30] J. Pérez, F. Nashashibi, B. Lefaudeux, P. Resende, and E. Pollard, "Autonomous Docking Based on Infrared System for Electric Vehicle Charging in Urban Areas", *Sensors*, vol. 13, *2*, pp. 2645–2663, 2013.

[31] F. Lamiraux and O. Lefebvre, "Sensor-based Trajectory Deformation: Application to Reactive Navigation of Nonholonomic Robots", *in Visual Servoing via Advanced Numerical Methods*, G. Chesi and K. Hashimoto, Eds., Springer, 2010, pp. 315–334.

[32] F. Lamiraux, D. Bonnafous, and O. Lefebvre, "Reactive Path Deformation for Nonholonomic Mobile Robots", *IEEE Transactions on Robotics*, vol. 20, *6*, pp. 967–977, Dec. 2004.

[33] D. Gonzalez, J. Perez, V. Milanes, and F. Nashashibi, "A Review of Motion Planning Techniques for Automated Vehicles", *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, *4*, pp. 1135–1145, 2016.

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

[35] K. Min and J. Choi, "A control system for autonomous vehicle valet parking", *in 13th International Conference on Control, Automation and Systems*, Gwangju, South Korea, 2013, pp. 1714–1717.

[36] Y. Yi, Z. Lu, Q. Xin, L. Jinzhou, L. Yijin, and W. Jianhang, "Smooth path planning for autonomous parking system", *in IEEE Intelligent Vehicles Symposium*, 2017, pp. 167–173.

[37] M. Chirca, G. Martin, R. Chapuis, C. Debain, and R. Lenain, "Autonomous Valet Parking System Architecture", *in IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 2619–2624.

[38] H. Vorobieva, S. Glaser, N. Minoiu-Enache, and S. Mammar, "Automatic parallel parking with geometric continuous-curvature path planning", *in IEEE Intelligent Vehicles Symposium*, 2014, pp. 465–471.

[39] 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 Symposium*, Seoul, Korea, 2015, pp. 1148–1153.

[40] 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 20th International Conference on Intelligent Transportation Systems*, Yokohama, 2017, pp. 1–8.

[41] 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 Symposium*, Changshu, China, 2018, pp. 326–333.

[42] H. Banzhaf, P. Sanzenbacher, U. Baumann, and J. M. Zollner, "Learning to Predict Ego-Vehicle Poses for Sampling-Based Nonholonomic Motion Planning", *IEEE Robotics and Automation Letters*, vol. 4, *2*, pp. 1053–1060, 2019.

[43] F. Chaumette, P. Rives, and B. Espiau, "The task function approach applied to vision-based control", *in Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*, IEEE, 1991, 1392–1397 vol.2.

[44] S. Hutchinson, G. Hager, and P. I. Corke, "A tutorial on visual servo control", *IEEE International Conference on Robotics and Automation*, vol. 12, *5*, pp. 651–670, 1996.

[45] A. Cherubini, F. Chaumette, and G. Oriolo, "Visual servoing for path reaching with nonholonomic robots", *Robotica*, vol. 29, *07*, pp. 1037–1048, 2011.

[46] A. Cherubini, F. Spindler, and F. Chaumette, "Autonomous visual navigation and laser-based moving obstacle avoidance", *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, *5*, pp. 2101–2110, 2014.

[47] 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 International Conference on Intelligent Transportation Systems*, Las Palmas, Spain, 2015, pp. 1791–1796.

[48] 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 Symposium*, Seoul, South Korea, 2015, pp. 77–82.

[49] D. Pérez-Morales, "Autonomous Parking", Master on Advanced Robotics, IRCCyN, École Centrale de Nantes, 2016.

[50] E. F. Camacho and C. Bordons, "Nonlinear Model Predictive Control: An Introductory Review", *in Assessment and Future Directions of Nonlinear Model Predictive Control*, 1, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 1–16.

[51] J. Richalet, A. Rault, J. Testud, and J. Papon, "Model predictive heuristic control", *Automatica*, vol. 14, *5*, pp. 413–428, Sep. 1978.

[52] S. V. Raković and W. S. Levine, Eds., *Handbook of Model Predictive Control*, ser. Control Engineering. Cham: Springer International Publishing, 2019.

[53] R. Findeisen and F. Allgower, "An Introduction to Nonlinear Model Predictive Control", *in 21st Benelux Meeting on Systems and Control, 2002, Veldhoven*, 2002, pp. 119–141.

[54] H. van Essen and H. Nijmeijer, "Non-linear model predictive control for constrained mobile robots", *in European Control Conference*, IEEE, 2001, pp. 1157–1162.

[55] F. Künhe, J. Gomes, and W. Fetter, "Mobile Robot Trajectory Tracking Using Model Predictive Control", *in II IEEE Latin-American Robotics Symposium*, São Luís, 2005, pp. 1–7.

[56] M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur, and P. Souères, "A Reactive Walking Pattern Generator Based on Nonlinear Model Predictive Control", *IEEE Robotics and Automation Letters*, vol. 2, *1*, pp. 10–17, 2017.

[57] R. Lenain, "Contribution à la modèlisation et à la commande de robots mobiles en présence de glissement : application au suivi de trajectoire pour les engins agricoles", PhD thesis, Université Blaise Pascal - Clermont-Ferrand II, 2005.

[58] R. Lenain, B. Thuilot, C. Cariou, and P. Martinet, "Adaptive and Predictive Path Tracking Control for Off-road Mobile Robots", *European Journal of Control*, vol. 13, *4*, pp. 419–439, Jan. 2007.

[59] A. Guillet, R. Lenain, B. Thuilot, and P. Martinet, "Adaptable Robot Formation Control: Adaptive and Predictive Formation Control of Autonomous Vehicles", *IEEE Robotics & Automation Magazine*, vol. 21, *1*, pp. 28–39, Mar. 2014.

[60] B. Amos, I. D. Jimenez Rodriguez, J. Sacks, B. Boots, and J. Z. Kolter, "Differentiable MPC for End-to-end Planning and Control", *in Advances in Neural Information Processing Systems*, 2018, pp. 8289–8300.

[61] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural Network Dynamics for Model-Based Deep Reinforcement Learning with Model-Free Fine-Tuning", *in IEEE International Conference on Robotics and Automation*, 2018, pp. 7559–7566.

[62] M. Fruchard, G. Allibert, and E. Courtial, "Choice of the control horizon in an NMPC strategy for the full-state control of nonholonomic systems", *2*, pp. 4149–4154, 2012.

[63] G. Allibert, E. Courtial, and F. Chaumette, "Predictive Control for Constrained Image-Based Visual Servoing", *IEEE Transactions on Robotics*, vol. 26, *5*, pp. 933–939, 2010.

[64] A. Amditis, P. Lytrivis, and E. Portouli, "Sensing and Actuation in Intelligent Vehicles", *in Handbook of Intelligent Vehicles*, London: Springer London, 2012, pp. 31–60.

[65] F. Rosique, P. J. Navarro, C. Fernández, and A. Padilla, "A Systematic Review of Perception System and Simulators for Autonomous Vehicles Research", *Sensors*, vol. 19, *3*, p. 648, Feb. 2019.

[66] C. Loper, C. Brunken, G. Thomaidis, S. Lapoehn, P. P. Fouopi, H. Mosebach, and F. Koster, "Automated valet parking as part of an integrated travel assistance", *in IEEE Conference on Intelligent Transportation Systems*, 2013, pp. 2341–2348.

[67] S.-H. Kim, J.-S. Kim, and W.-Y. Kim, "A method of detecting parking slot in hough space and pose estimation using rear view image for autonomous parking system", *in IEEE International Conference on Network Infrastructure and Digital Content*, 2016, pp. 452–454.

[68] J. K. Suhr and H. G. Jung, "Automatic Parking Space Detection and Tracking for Underground and Indoor Environments", *IEEE Transactions on Industrial Electronics*, vol. 63, *9*, pp. 5687–5698, 2016.

[69] S.-J. Han and J. Choi, "Parking Space Recognition for Autonomous Valet Parking Using Height and Salient-Line Probability Maps", *ETRI Journal*, vol. 37, *6*, pp. 1220–1230, 2015.

[70] Ho Gi Jung, Young Ha Cho, Pal Joo Yoon, and Jaihie Kim, "Scanning Laser Radar-Based Target Position Designation for Parking Aid System", *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, *3*, pp. 406–424, 2008.

[71] T. Westfechtel, K. Ohno, N. Mizuno, R. Hamada, S. Kojima, and S. Tadokoro, "Parking Spot Estimation and Mapping Method for Mobile Robots", *IEEE Robotics and Automation Letters*, vol. 3, *4*, pp. 3371–3378, Oct. 2018.

[72] 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.

[73] O. Kermorgant and F. Chaumette, "Dealing with constraints in sensor-based robot control", *IEEE Transactions on Robotics*, vol. 30, *1*, pp. 244–257, 2014.

[74] F. Chaumette and S. Hutchinson, "Visual servo control, part I : Basic Approaches", *IEEE Robotics Automation Magazine*, vol. 13, *December*, pp. 82–90, 2006.

[75] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)", *in IEEE International Conference on Robotics and Automation*, 2011, pp. 1–4.

[76] N. Andreff, B. Espiau, and R. Horaud, "Visual Servoing from Lines", *International Journal of Robotics Research*, vol. 21, *8*, pp. 679–699, 2002.

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

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

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

[80] D. Perez-Morales, O. Kermorgant, S. Dominguez-Quijada, and P. Martinet, "Laser-Based Control Law for Autonomous Parallel and Perpendicular Parking", *in Second IEEE International Conference on Robotic Computing*, Laguna Hills, USA, 2018, pp. 64–71.

[81] O. Kanoun, F. Lamiraux, and P.-B. Wieber, "Kinematic control of redundant manipulators: generalizing the task priority framework to inequality tasks", *IEEE Transactions on Robotics*, vol. 27, *4*, pp. 785–792, 2011.

[82] G. Antonelli, "Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems", *IEEE Transactions on Robotics*, vol. 25, *5*, pp. 985–994, 2009.

[83] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation", *The International Journal of Robotics Research*, vol. 33, *7*, pp. 1006–1028, Jun. 2014.

[84] R. Brockett, "Asymptotic stability and feedback stabilization", *in Differential Geometric Control Theory*, R. Brockett, R. Millman, and H. Sussmann, Eds., New York: Birkhauser, 1983, pp. 181–191.

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

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

[87] N. Shahriari, S. Fantasia, F. Flacco, and G. Oriolo, "Robotic visual servoing of moving targets", *in IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 77–82.

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

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

[90] G. Allibert, E. Courtial, and F. Chaumette, "Visual Servoing via Nonlinear Predictive Control", *in Visual Servoing via Advanced Numerical Methods*, 2010, pp. 375–393.

[91] I. A. Sucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library", *IEEE Robotics & Automation Magazine*, vol. 19, *4*, pp. 72–82, 2012.

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

CENTRALE NANTES

Titre: Commande référencée multi-capteurs pour des applications de Parking Intelligent

Mot clés : Véhicules intelligents, commande référencée capteur, commande prédictive, système de parking intelligent

Resumé : Dans l'objectif de développer des systèmes de parking plus avancés que ce que l'on trouve actuellement, différentes manœuvres typiquement réalisées dans des situations de parking ont été revisitées en utilisant une approche originale : la commande référencée multi-capteurs. En outre, pour surmonter les limitations bien connues des techniques référencée capteur classiques, les manœuvres de parking ont également été formalisées dans un cadre commun de commande prédictive référencée multi-capteur. Les stratégies développées ont été testées avec une Renault ZOE robotisée validant l'approche.

Title: Multi-sensor-based control for Intelligent Parking applications

Keywords : Intelligent vehicles, sensor-based control, model predictive control, intelligent parking system.

Abstract: With the aim of developing a more advanced parking system than what is currently available, different maneuvers typically performed in parking situations have been revisited using an original Multi-Sensor-Based Control (MSBC) approach. Furthermore, in order to overcome the well-known limitations of classical sensor-based techniques, the parking maneuvers have been formalized as well under a common Multi-Sensor-Based Predictive Control (MSBPC) framework. The developed strategies have been tested extensively using a robotized Renault ZOE with positive outcomes.