An Ego-Based Approach to Planning and Control for Automated Valet Parking Applications*

Massimo Canale†, Francesco Cerrito† and Pandeli Borodani♯

Abstract— **This paper introduces an *ego-based* approach to automated valet parking in low-complexity scenarios. The proposed solution aims at realizing the valet parking application by exploiting a minimum amount of information provided by the infrastructure and implementing all the required driving functions based on proprioceptive sensor data. An encapsulated hierarchical architecture is introduced to accomplish this aim. The higher hierarchical level, i.e. the Global Planner, computes a feasible and robust geometric path from the drop-off area to the parking destination. At the lower level, the Local Planner based on Model Predictive Control and Artificial Potential fields, tracks the path and realizes the final parking maneuver. Decision-making during vehicle maneuvering is implemented by a suitable behavioral logic that, based on sensor-acquired data, manages vehicle interaction in specific situations such as, e.g., precedence in road intersections, and traffic jam handling. Extensives simulation results performed in realistic driving scenarios are introduced to show the effectiveness of the proposed approach.**

I. Introduction

In the near future vehicles will be electric, connected, and autonomous. Many of the leading actors in the automotive industry seek to claim leadership in automated vehicle technologies and all evidence suggests that automated driving [1] will represent a key challenge in the evolution of smart cities and a benchmark for innovation.

Parking is the most time-consuming maneuver during a journey, taking up almost 4 days a year, and during this time drivers experience anxiety and stress [2]. It is also estimated that around 40% of car accidents involving personal injury occur during parking or maneuvering [3]. Another problem is the severe shortage of parking space caused by the increasing number of private vehicles in urban areas [4]. Unfortunately, due to its cost and high land use, the creation of additional parking spaces is not an efficient, all-encompassing solution [5].

Automated Valet Parking (AVP), firstly introduced in [6], has the potential to effectively address such challenges. One of its benefits is the capability to reduce the time required for parking. The inclusion of advanced sensor technologies, effective control systems, and the use of infrastructure data significantly reduces the likelihood of accidents. In addition, AVP's ability to perform precise and coordinated maneuvers allows for optimal use of parking space, even in challenging situations such as tight spaces or traffic congestion.

The operating principle of the AVP application [7] can be summarised in the following points:

- the driver hands over control to the vehicle at the drop-off area and activates through Human Machine Interface (HMI) the AVP functionality;
- the car is automatically driven to the parking location;
- to request the car back, the driver communicates with the vehicle via an HMI, such as a smartphone;
- the vehicle will leave the parking spot and drive autonomously back to the pick-up point.

The automotive industry has shown a keen interest in addressing AVP challenges. For example, Bosch, together with Daimler, implemented an AVP pilot project in a mixed traffic environment, which can be accessed via a smartphone app and does not require a driver [8]. In a parallel development, Audi integrated automated parking with additional services, including the option to recharge and wash the vehicle, creating a more comprehensive and convenient parking experience [9].

Banzhaf et al. [10] performed a thorough review of the AVP literature. Such a study highlights that the intelligence required for automated driving and parking can either be located in the vehicles without changing the infrastructure, or in the infrastructure by simply adding a remote control unit to the car, or to both for mutual support.

The European Union's research project, V-Charge [11], aims to implement AVP using only sensors close to market availability and offers a comprehensive review of the various subsystems demonstrating the feasibility of developing such an application exclusively with close-to-market sensors.

In [12] the infrastructure is actively used in motion planning. In particular, the vehicle trajectory is calculated by the server of the infrastructure system and sent to the vehicle. The vehicle controller uses the homotopic method to track the reference trajectory and account for obstacle boundaries, and Gauss pseudospectral method is implemented to discretize this optimal control problem. Kneissl et al. [13] distribute the control functionality between an infrastructure server and the local autonomous vehicle control units. In particular, via a V2I communication interface, the plant control variables computed by the Model Predictive Control (MPC) are shared with the coordination unit. Exploiting the V2I infrastructure the proposed solution can detect and handle conflict zones, for this purpose, the trajectory generation process is decomposed between the parking area management and the local vehicle controller. This research illustrates the benefits of implementing an intelligent infrastructure. In particular, it enables efficient and effective management of interactions among vehicles, such as managing conflict zones.

A different approach is proposed by [14] where the task of AVP is executed by an external robot, which is responsible for transporting and parking the vehicle. A fleet management system is implemented to coordinate the movements of this robot. This system is crucial to ensure efficient and effective coordination of the robot's operations.

In this study, we present a novel approach to AVP characterized by minimal infrastructure dependency, flexibility, and cost-effectiveness. In particular, the proposed solution

- operates efficiently with minimal infrastructure infor-

*This publication is part of the project Piano Nazionale di Ripresa e Resilienza (PNRR)-Next Generation Europe which has received funding from the Italian Ministry of University and Research – DM 117/2023

† Dipartimento di Automatica e Informatica, Politecnico di Torino, Italy massimo.canale@polito.it, francesco.cerrito@polito.it

♯ Centro Ricerche Fiat, Italy pandeli.borodani@crf.it

mation, providing a comprehensive and effective AVP solution that adeptly manages a wide range of scenarios, from traffic congestion to unregulated road intersections;
- employs an encapsulated architecture control that is based on parameterized algorithms and is organized into three different levels: the Global Planner, the Local Planner, and the Parking Planner. Each level operates hierarchically, performing specific tasks while maintaining its independence. This modular structure enhances the adaptability of the architecture: each module can be replaced to meet different application requirements. For instance, if an intelligent infrastructure is available, the Global Planner can be adapted to work in synergy with it, without any modification to the other levels. In addition, the parameterization of each module allows a tailor-made application for different car models.

Consequently, the proposed *ego-based* approach avoids the need to employ intelligent infrastructure [10], [12], [13] or specialized robots [14]. In addition, the adaptability of the architecture minimizes the need for modifications to the car park to accommodate the AVP service, improving cost efficiency. The decision-making process during vehicle operation is implemented by a Finite-State Machine (FSM) [15] based on the data acquired by the *ego-vehicle* proprioceptive sensors. In addition, Artificial Potential Fields (APFs) are employed to obtain a mathematical and comprehensive description of the surrounding environment [16]. As demonstrated in [17] APFs can be combined to account for the presence of different road actors that affect the vehicle motion. An MPC controller [18] exploits the information provided by the APFs to control the vehicle's lateral and longitudinal maneuvers. The combination of these two techniques has already shown good results in the implementation of automated driving solutions in a highway scenario [19].

II. The ego-based AVP Architecture

According to the SAE J3016 [1] standard, every AD application must be characterized by the operational design domain (ODD) and the dynamic driving tasks (DDTs).
The ODD defined for the AVP service is bounded by the following conditions:

- low-speed maneuvers for reverse and forward motion, range between $-7{}^{km}/_{h}$ and $15{}^{km}/_{h}$;
- structured environment (vehicle reserved area);
- full sensor configuration available;
- V2I communication with the specialized infrastructure available only in the drop-off zone and to notify the infrastructure each time a given target parking bay has been reached by a vehicle.

In the described scenario, the vehicle can perform the following DDTs:

- obstacle classification/identification and decision-making functions;
- precedence system that allows to handle multi-vehicle scenarios at road intersections;
- re-computation of the optimal destination and the optimal route in case of unforeseen events, such as the presence of obstacles, blocking the vehicle's current path;
- implementation of an Adaptive Cruise Control (ACC) and Lane Keeping (LK) mode.

We assume that the *ego-vehicle* is equipped with a complete sensor configuration that includes: front and surround-view cameras, front long and short-range radar, inertial measurement unit, and differential global positioning system.

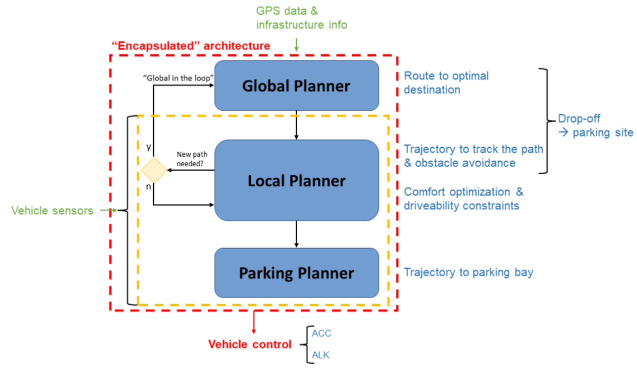

Fig. 1: Motion Planner's encapsulated architecture. Three distinct and interrelated levels are introduced: Global Planner, Local Planner, and Parking Planner.

As shown in Fig. 1 the proposed control architecture of the AVP application is characterized by the interconnection of three hierarchical levels:

- **Global Planner**: that computes an optimal path from a starting point to a destination. Factors such as route length, travel time, lane crossings, and kinematic constraints are accounted for.
- **Local Planner**: which performs the vehicle control to track the path computed by the Global Planner. MPC and APFs are used to obtain a feasible trajectory that accounts for kinematic and dynamic constraints.
- **Parking Planner**: that aims at generating a collision-free, feasible trajectory for the parking maneuver. The design of these maneuvers is beyond the scope of this article, possible solutions can be found in [20] and [21].

As highlighted in the introduction, a design choice of paramount importance for AVP problem consists of defining how to distribute the intelligence between the vehicle and the infrastructure. The main objective of this paper is to develop an *ego-based* control strategy where the intelligence is allocated uniquely to the vehicle to minimize the amount of information about the parking layout and aims at creating an infrastructure-independent application. This approach makes it possible to implement AVP in existing car parking areas while maintaining the layout of the car parks and introducing a simple infrastructure.
In particular, when the user hands out the vehicle at the drop-off area, the AVP functionality takes over and a V2I communication channel is temporarily established to provide the *ego-vehicle* the data required for navigation in the parking area and needed to perform correctly the parking tasks. The adoption of a heavily *ego-based* AVP policy implies that the amount of information given by the infrastructure is as low as possible. The minimum information that still provides the necessary support to enable all the AVP features and functionalities includes:

- lane width and parking bay dimensions;
- position and orientation of all the assigned parking places;
- waypoints map, including location and orientation;
- a list of the waypoints associated with 4-ways intersections;
- a precedence attribute (i.e., a flag) associated to each node, to indicate the points on the map where a mandatory stop must be made.

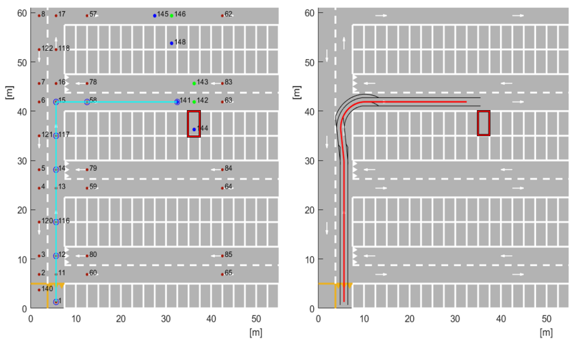

Fig. 2: Global planning path generation process. On the left, the nodes provided by the infrastructure to the ego-vehicle and the optimal path. On the right, the generated geometric path connecting the drop-off area and the assigned parking bay.

III. Global Planner

The Global Planning module exploits the data provided by the infrastructure and involves two main steps (Fig. 2). At first, it finds an optimal path, taking into account environmental constraints and static obstacles. This is done using the Dijkstra algorithm. The procedure iteratively selects nodes with lower costs, designates them as definitive, and updates the costs of neighboring nodes (Fig. 2 left). Second, it connects these nodes using an appropriate geometric path that takes into account the kinematic constraints of the vehicle. Linear segments are used for straight-line nodes, while Dubins' curves [22] are implemented for curved paths (Fig. 2 right).

A. Optimal path computation

To implement the Dijkstra algorithm the data provided by the infrastructure are exploited to create a mathematical representation of the parking area through an adjacency matrix. To this scope, the following criterion is adopted: two nodes, j and k, with position $(x_j,\ y_j)$ and $(x_k,\ y_k)$ respectively, are connected by an arc directed towards k (i.e. k is consecutive to j) if and only if:

1) $x_k > 0$ in the reference frame with origin (x_j, y_j) and i-axis aligned along node's j orientation;
2) there are no other nodes between j and k with the same orientation of j.

The weight assigned to each graph's arc is computed as:

$$\sum_{i=1}^{N} W_i = \sum_{i=1}^{N} (k_1 L_i + k_2 S_i) \tag{1}$$

where N is the number of arcs, L_i is the length of the i-th arc, k_1 and k_2 are gain coefficients that require tuning. S_i is designed to penalize Left-Hand (LH) curves, as they require crossing the opposite lane, except when on the external perimeter.

B. Geometric path generation

It is then necessary to connect such nodes with a suitable geometric path that keeps into account the kinematic constraints of the vehicle, stays within its lane, avoids invading the opposite lane during turns, and does not cross continuous lane markers.

Starting from the sequence of nodes provided by Dijkstra's algorithm, the interconnection path between two subsequent nodes is chosen based on their orientation. If two consecutive nodes have the same orientation, they are connected through a straight line. If this is not the case, a RH or a LH turn is generated by exploiting a combination of line segments and circular arcs.

To generate the optimal combination of arcs and segments, the Dubins and Reeds-Shepp curves [23] are used. This solution allows to take directly into account vehicle kinematic constraints as a function of the car wheelbase and its steering angle [24], which is bounded between $-24.5°$ and $24.5°$.

C. Global-in-the-loop

The global-in-the-loop is the architecture's module that seeks alternative paths. When, e.g., a traffic jam is detected, the *ego-vehicle* starts braking until it stops and evaluates two key elements:

- the position of the jam, as $(x_{jam},\ y_{jam})$;
- the latest route node, based on the knowledge of its own position at the current instant.

The adopted approach updates the adjacency matrix, finds the closest node to the jam, and updates the connecting edge. It uses Dijkstra's algorithm to find a new route and designs connection maneuvers. The global-in-the-loop improves vehicle intelligence, providing flexibility and mimicking human decision-making.

IV. Local planner

The Local Planner is the module that is in charge of tracking the geometric path generated by the Global Planner. The Global Planner generates a geometric path that is collision-free and feasible in terms of maximum speed and curvature, but it does not take into account any constraints on acceleration, jerk and derivative of steering angle, nor is the path curvature continuous when changing from a straight line to an arc of radius. For this reason, an MPC that uses the APF function in its cost function is designed.

APFs are implemented to perform all the required driving tasks through an organic solution. The Potential Fields (PFs) presented in [19] are used to influence the vehicle behaviour through repulsive and attractive forces. In particular, repulsive PFs are used for LK and obstacle avoidance, while attractive PFs are used to implement ACC functionalities.

Since the MPC must solve an optimization problem to compute the desired control action in real-time, the choice of the prediction time T_p and the control time must be consistent with the computational time needed to perform these operations. These values in combination with the sample time T_s determine two pivotal parameters of the MPC: the prediction horizon H_p and the control horizon H_c. As reported in [25], the reaction time for a human driver typically is in the range of $0.5s$ to $1.2s$. Hence, the following values have been selected: $T_s = 0.5s$, $H_p = 6$, $H_c = 2$.

To account for the driving task execution, the APFs are included in the MPC optimization problem as additional terms in the cost function, along with the standard state tracking error and control input rate effort. In the resulting cost function, the APFs act as a virtual force exerted on the vehicle at each step. The MPC minimizes at each time instant k the cost function J acting on the plant inputs a and w_δ:

$$\begin{aligned} J(U(k)) = \sum_{i=k}^{k+H_p} \Big[& W_L P_L^2(i) + W_o P_o^2(i) + \\ & W_{ACC}((P_o^{ACC})^2(i) + P_{sink}^2(i)) + \\ & Q_1(x(i) - x_{tar})^2 + Q_2(y(i) - y_{tar})^2 + \\ & Q_3(\theta(i) - \theta_{tar})^2 + Q_4(v(i) - v_{des})^2 + \\ & Q_5\delta(i)^2 + R_1\Delta v(i)^2 + R_2\Delta\delta(i)^2 \Big] \end{aligned} \tag{2}$$

$$
\begin{aligned}
\text{subject to} \quad & x(i+1) = f(x(i), u(i)) \\
& v_{min} \le v(i) \le v_{max} \\
& a_{min}T_s \le \Delta v(i) \le a_{max}T_s \\
& \delta_{min} \le \delta(i) \le \delta_{max} \\
& w_{\delta_{min}} \le \Delta\delta(i) \le w_{\delta_{max}} \\
& j_{min}T_s \le \Delta a(i) \le j_{max}T_s
\end{aligned} \tag{3}
$$

The terms of the optimal control problem (2) (3) take into account the following aspects:

- $f(x(i), u(i))$ is the single-track kinematic model that describes the vehicle behavior, see e.g. [26]. In the kinematic model variables x, y, and θ uniquely define the position and the orientation of the vehicle in the inertial frame. v, and δ are the vehicle speed and the wheel steering angle respectively, while a and w_δ are the longitudinal acceleration and the wheel steering speed.
- P_L and P_o are the path-keeping and obstacle avoidance APFs respectively. P_o^{ACC} and P_{sink} also include the contribution of the ACC APFs, which becomes active whenever the controller operates in the corresponding mode.
- $(x - x_{tar})$, $(y - y_{tar})$ and $(\theta - \theta_{tar})$ define the target pose error. These terms are used to ensure that the vehicle stops and parks at the desired point with the correct orientation.
- $(v - v_{des})$ represents the error between the target and actual vehicle speed.
- δ introduces a cost that describes the steering angle effort.
- W_L, W_O,W_{ACC}, Q_1, Q_2, Q_3, Q_4, Q_5, R_1 and R_2 , are the weighting factors. These values strongly affect the behavior of the controller, therefore their values are a function of the behavioral logic state.

When generating the trajectory, along with the mentioned constraints on the discretized plant model $f(x(i), u(i))$, the velocity v and steering angle δ, the comfort constraints on acceleration a, jerk j, and the derivative of the steering angle w_δ are introduced as constraints (3).

V. Behavioral Logic

Based on the sensor data, the Behavioural Logic (BL) performs obstacle detection, evaluates the most urgent driving task, and performs a decision-making process to select the most appropriate maneuver. The core element of the BL is a FSM, that selects the appropriate controller mode depending on the driving situation detected by the sensors. The FSM may change its state in response to some inputs; the change from one state to another is called a transition (Table I).
Fig. 3 shows a schematic of the designed FSM with the different states and transitions. Each state of the FSM is characterized by a specific control mode, defined as a set of cost function (2) weights, defined in Table II. Hereafter, we introduce a description of the considered driving modes.

A. Nominal mode

The vehicle follows its trajectory, and no obstacle that requires immediate action is detected. Different weights (Table II) of the cost function (2) and velocity references are set during the path, depending on the *ego-vehicle*'s distance from the proximity position to the parking bay. When the *ego-vehicle* approaches the destination, the reference velocity is set to zero to obtain a comfortable, and accurate stop phase. Furthermore, the weights associated with the position error with respect to the final destination are activated. Another situation that triggers a modification of the velocity reference occurs when a standing obstacle is detected on the path at a long range.

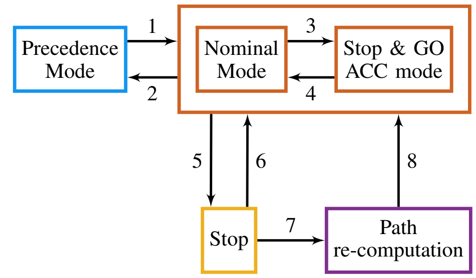

Fig. 3: Finite state machine representation

TABLE I: FSM transitions

Transition	Conditions
1	No further detection in the crossing area
2	1) A LH turn must be performed 2) The next route node is a precedence node
3	Leading vehicle detection
4	Absence of leading vehicle detection
5	1) Jam detection 2) Parking leading vehicle 3) Close-range obstacle on the path
6	No further on path detection
7	Jam detection confirmed AND Vehicle has stopped
8	Connection maneuver have been tracked, if present

B. Stop&Go ACC mode

Stop&Go ACC mode is activated when a LV is detected and the *ego-vehicle* must track a suitable distance from the preceding vehicle. In addition, this driving mode is applied when the *ego-vehicle* is near the target parking bay: the reference velocity is set to zero and the weights associated with the position error with respect to the final destination are activated.

C. Stop mode

This state refers to stop maneuvers that must be activated in the presence of the scenarios described below:

- a moving or still car has been suddenly detected on the *ego-vehicle*'s path at close range. Prescribing a stop is a precautionary measure to avoid collision while fulfilling comfort requirements;
- a jam has been identified and the global-in-the-loop is invoked to find an alternative path;
- in Stop&Go ACC mode, if the LV stops suddenly and gets too close, the *ego-vehicle* will stop.

D. Precedence mode

In the presence of a road junction two different precedence situations can occur and two different states must be designed accordingly:

- **Type 1**: the *ego-vehicle* is approaching an intersection with a compulsory stop. Entering this state:
  1) the *ego-vehicle* evaluates the position and orientation that maximizes visibility at the intersection as a function of the *ego-vehicle* characteristics (e.g. dimensions, sensors) and stops exactly as described above;

TABLE II: Cost function weights

Weight Coefficient	Nominal Mode	ACC Mode	Stop Mode	Precedence Mode	Tarffic Jam Retro
W_L	1	5	12	10	2
W_{ACC}	0	1.3	0	0	0
$Q_{1,2,3}$	[0,0,0]	[0,0,0]	[15,15,20]	[15,15,20]	[0,0,0]
$Q_{4,5}$	[0,16]	[0,3]	[1,14.8]	[1,15]	[0,16]
$R_{1,2}$	[0.1,10]	[0.1,10]	[1,5]	[1,5]	[0.1,10]

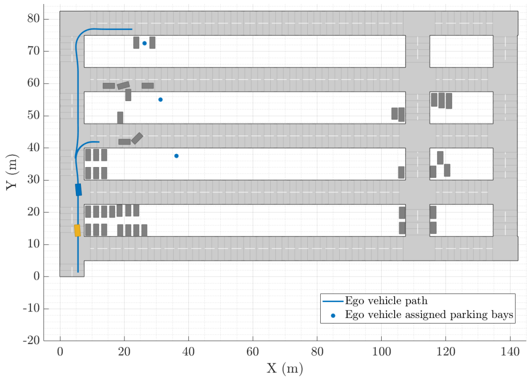

Fig. 4: Simulation environment overview

2) the *ego-vehicle* stands still until the intersection is free from other vehicles;
3) when the road is free, the vehicle comes back in Nominal Mode to resume its motion along the path.

- **Type 2**: the *ego-vehicle* is approaching a LH turn and precedence must be given if necessary. Entering this state:
1) the *ego-vehicle* slows down to be ready for a possible stopping maneuver;
2) the crossing area and the opposite lane are scanned while keeping low speed. If no other vehicles are detected in this range, the *ego-vehicle* speeds up and goes on its path, otherwise, it stops at a designated point that does not invade the intersection;
3) when the road is free the vehicle can come back in Nominal Mode and resume its motion along the path.

E. Path re-computation

This state is activated when a jam on the path is detected, requiring a new path computation. First, the *ego-vehicle* must stop, then the global-in-the-loop algorithm is invoked. If no connection maneuver must be tracked, the vehicle immediately returns to Nominal Mode. When tracking a backward connection maneuver, the reference velocity is set to $-1 m/s$ and to zero when the vehicle is close to the target.

VI. Simulation

A simplified simulation scenario is introduced in the following to highlight the principal features of the proposed approach. The *ego-vehicles* enter the parking area one after the other and drive autonomously to the parking bays assigned by the infrastructure. During the path to the parking proximity point, the vehicle operates maneuvers that highlight how functions like global-in-the-loop and stop mode are effectively handled.
Fig. 4 provides a complete overview of each *ego-vehicle's* path and assigned bays. To provide a comprehensive simulation environment, the test scenario is implemented in the MATLAB Automated Driving Toolbox [27]. Furthermore, a realistic 3D scenario has been created to conduct a comprehensive evaluation of the performance of our proposed

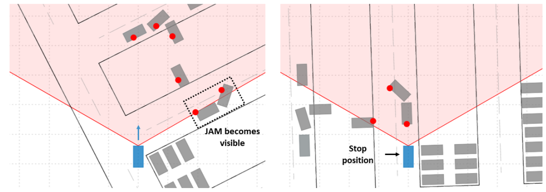

Fig. 5: The *ego-vehicle*, after a RH turn, detects a traffic jam caused by two stationary vehicles

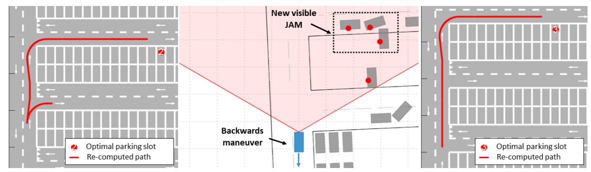

Fig. 6: In the presence of a traffic jam, the *ego-vehicle* recompute the optimal parking slot and updates its path

solution.
The global-in-the-loop operates when it is invoked, and in general, to showcase the vehicle's obstacle recognition capabilities. As shown in Fig. 4 three different parking slots are assigned to the *Ego-vehicle 1* represented in blue. While in the drop-off zone, the Global Planner is invoked and computes the optimal parking bay and route. In this phase, the target bay is the one that minimizes route length.
As shown in Fig. 5, left, the vehicle encountered a traffic jam after the RH turn performed to reach the parking bay 1. This situation cannot be detected in advance by the sensors because of the presence of other parked vehicles. In particular, the *ego-vehicle* detects two distinct obstacles on the road, occupying both lanes, understands that they are not moving, and decides to stop and identify them as a jam (Fig. 5 right). This information is passed to the FSM section of the behavioral logic, which invokes the global-in-the-loop to evaluate possible alternative routes to another assigned parking location. The grid of nodes is exploited for the computation of the new path and a smooth connection maneuver is generated to link the current vehicle's position to the new path. In this scenario, since the jam occupies both lanes of the road, a backward-turn connection is performed (Fig. 6, left image). While on its way during the backward connection maneuver, the *ego-vehicle* can detect a group of standing vehicles on the newly calculated path, that prevents access to the parking bay 2. This time, thanks to the clean line of sight, the *ego-vehicle* correctly detects and identifies the obstacles as a jam (Fig. 6, center image), invokes once again the global-in-the-loop, re-evaluates the optimal destination and path, and decides to change parking place, selecting the third and uppermost one as a destination (Fig. 6, right image).
Fig. 7 reports the *Ego vehicle 1* speed and steering angle during the journey: it is possible to see that all the maneuvers are performed obtaining a smooth behavior of the plots and fulfilling the constraints identified during the design phase.

VII. Conclusion

This paper presented an *ego-based* AVP system that avoids the need for intelligent infrastructure. In this approach, the infrastructure only provides essential data without any decision-making responsibility. Simulations provide a comprehensive understanding of the controller's ability to manage different scenarios. The introduction of APFs and FSMs enables the *ego-vehicle* to safely perform tasks in

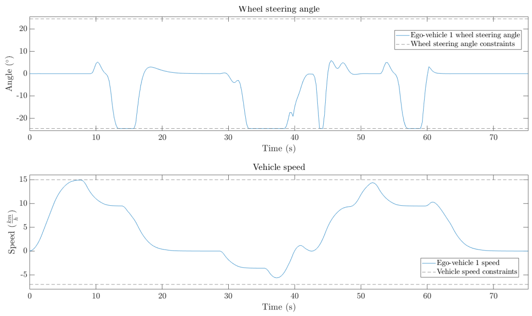

Fig. 7: *Ego-vehcile 1* speed and wheel steering angle

all outlined contexts, effectively emulating human driver behavior. This architecture is a promising solution for the future, mainly due to two factors. Firstly, it eliminates the need for expensive modifications to parking facilities or the design and implementation of intelligent infrastructure. The *ego-based* AVP can be seamlessly integrated into traditional parking structures with minimal adjustments. Secondly, the use of the *ego-based* approach is critical to maintaining the reliability of the AVP application in the event of failures in the intelligent infrastructure. While it is undeniable that intelligent infrastructure can significantly improve AVP performance, it's equally important to ensure uninterrupted operation in the event of such infrastructure failures. This goal can be achieved thanks to the flexibility of the proposed *ego-based* approach, which is able to handle both situations efficiently within a unified framework.

Acknowledgment

This research was developed under the partnership project between Politecnico di Torino and Centro Ricerche Fiat/Stellantis: *AVP control design via advanced MPC methodologies*. The authors would also like to thank A. Lattarulo, F. Menon, G. Morello, and E. Piantelli former students at *Politecnico di Torino* whose MSc thesis inspired this work.

References

[1] "J3016_202104: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles - SAE International." [Online]. Available: https://www.sae.org/standards/content/j3016_202104/

[2] "Motorists spend nearly 4 days a year looking for a parking space," Jan. 2017. [Online]. Available: https://www.britishparking-media.co.uk/news/motorists-spend-nearly-4-days-a-year-looking-for-a-parking-space

[3] "Allianz | A sudden bang when parking." [Online]. Available: https://www.allianz.com/en/press/news/commitment/community/150505-a-sudden-bang-when-parking.html

[4] "Report shows extreme shortage of parking spaces in China's megacities." [Online]. Available: https://news.cgtn.com/news/3d67544e7a49444e/share_p.html

[5] "Parking Dominates Our Cities. But Do We Really *See* It?" Nov. 2019. [Online]. Available: https://www.strongtowns.org/journal/2019/11/27/parking-dominates-our-cities-but-do-we-really-see-it

[6] M. Wada, K. S. Yoon, and H. Hashimoto, "Development of advanced parking assistance system," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 1, pp. 4–17, Feb 2003.

[7] R. Dariani, T. Lobig, J. Lauermann, J. Rieck, and J. Schindler, "Trajectory planning for cooperative autonomous valet parking," 10 2019.

[8] "Automated Valet Parking – fast, safe, driverless." [Online]. Available: https://www.bosch.com/stories/automated-valet-parking/

[9] "Automated Valet Parking: That time my Audi parked itself." [Online]. Available: https://www.audi.com/en/innovation/autonomous-driving/automated-valet-parking.html

[10] H. Banzhaf, D. Nienhüser, S. Knoop, and J. M. Zöllner, "The future of parking: A survey on automated valet parking with an outlook on high density parking," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2017, pp. 1827–1834.

[11] U. Schwesinger, M. Bürki, J. Timpner, S. Rottmann, L. Wolf, L. M. Paz, H. Grimmett, I. Posner, P. Newman, C. Häne, L. Heng, G. H. Lee, T. Sattler, M. Pollefeys, M. Allodi, F. Valenti, K. Mimura, B. Goebelsmann, W. Derendarz, P. Mühlfellner, S. Wonneberger, R. Waldmann, S. Grysczyk, C. Last, S. Brüning, S. Horstmann, M. Bartholomäus, C. Brummer, M. Stellmacher, F. Pucks, M. Nicklas, and R. Siegwart, "Automated valet parking and charging for e-mobility," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2016, pp. 157–164.

[12] C. Chen, B. Wu, L. Xuan, J. Chen, T. Wang, and L. Qian, "A Trajectory Planning Method for Autonomous Valet Parking via Solving an Optimal Control Problem," *Sensors*, vol. 20, no. 22, p. 6435, Jan. 2020, number: 22 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/20/22/6435

[13] M. Kneissl, A. K. Madhusudhanan, A. Molin, H. Esen, and S. Hirche, "A multi-vehicle control framework with application to automated valet parking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5697–5707, 2021.

[14] B. B. Carlos, M. Williams, and B. Peloudeau, "Real-time nmpc for an automated valet parking with load-based safety constraints and a path-parametric model," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 10 006–10 013.

[15] J. Wang, *Formal Methods in Computer Science*. New York: Chapman and Hall/CRC, Jul. 2019.

[16] J. Zhao, Z. Zhang, Z. Xue, and L. Li, "A Hierarchical Vehicle Motion Planning Method For Cruise In Parking Area," in *2021 5th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, Oct. 2021, pp. 1–6.

[17] Z. Huang, Q. Wu, J. Ma, and S. Fan, "An APF and MPC combined collaborative driving controller using vehicular communication technologies," *Chaos, Solitons & Fractals*, vol. 89, pp. 232–242, Aug. 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0960077915003689

[18] Y. Shi, P. Wang, and X. Wang, "An Autonomous Valet Parking Algorithm for Path Planning and Tracking," in *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*, Sep. 2022, pp. 1–7, iSSN: 2577-2465.

[19] M. Canale and V. Razza, "Automated driving control in highway scenarios through a two-level hierarchical architecture," *IEEE Access*, vol. 12, pp. 86 470–86 486, 2024.

[20] D. J. Kim and C. C. Chung, "Automated perpendicular parking system with approximated clothoid-based local path planning," *IEEE Control Systems Letters*, vol. 5, no. 6, pp. 1940–1945, 2021.

[21] S. Sedighi, D.-V. Nguyen, and K.-D. Kuhnert, "A New Method of Clothoid-Based Path Planning Algorithm for Narrow Perpendicular Parking Spaces," in *Proceedings of the 5th International Conference on Mechatronics and Robotics Engineering*. Rome Italy: ACM, Feb. 2019, pp. 50–55. [Online]. Available: https://dl.acm.org/doi/10.1145/3314493.3314512

[22] L. E. Dubins, "On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents," *American Journal of Mathematics*, vol. 79, no. 3, p. 497, Jul. 1957. [Online]. Available: https://www.jstor.org/stable/2372560?origin=crossref

[23] J. Reeds and L. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367–393, Oct. 1990. [Online]. Available: http://msp.org/pjm/1990/145-2/p06.xhtml

[24] R. Oliveira, P. F. Lima, M. Cirillo, J. Mårtensson, and B. Wahlberg, "Trajectory Generation using Sharpness Continuous Dubins-like Paths with Applications in Control of Heavy Duty Vehicles," May 2018, arXiv:1801.08995 [cs]. [Online]. Available: http://arxiv.org/abs/1801.08995

[25] P. Droździel, S. Tarkowski, I. Rybicka, and R. Wrona, "Drivers 'reaction time research in the conditions in the real traffic," *Open Engineering*, vol. 10, no. 1, pp. 35–47, Jan. 2020, publisher: De Gruyter Open Access. [Online]. Available: https://www.degruyter.com/document/doi/10.1515/eng-2020-0004/html

[26] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2015, pp. 1094–1099, iSSN: 1931-0587.

[27] T. M. Inc., "Automated driving toolbox (R2022b)," Natick, Massachusetts, United States, 2022. [Online]. Available: https://www.mathworks.com