# Parallel Parking: Optimal Entry and Minimum Slot Dimensions

Jiri Vlasak[1,2][a], Michal Sojka[2][b] and Zdeněk Hanzálek[2][c]

[1] *Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic*

[2] *Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, Czech Republic*

Keywords: Automated Parking, Parallel Parking.

Abstract: The problem of path planning for automated parking is usually presented as finding a collision-free path from initial to goal positions, where three out of four parking slot edges represent obstacles. We rethink the path planning problem for parallel parking by decomposing it into two independent parts. The topic of this paper is finding optimal parking slot entry positions. Path planning from initial to entry position is out of scope here. We show the relation between entry positions, parking slot dimensions, and the number of backward-forward direction changes. This information can be used as an input to optimize other parts of the automated parking process.

## 1 INTRODUCTION

Driver assistance systems experience a great expansion. Proposals for automated driving are topics of many teams of engineers and academics. One of the basic building blocks of automated driving is the (automated) parking assistant.

A common approach to the problem of path planning for parallel parking is to geometrically compute a path from the current (initial) position to the goal position inside the parking slot. When an optimization-based approach is used, the time or path length between initial and goal positions is usually minimized. But what if the given goal position is not chosen optimally?

In this paper, we reformulate the path planning problem of automated parking for a parallel parking slot into the problem of finding optimal entry and goal positions. We solve this problem for the given parking slot and car dimensions using a simulation approach. A feasible path is computed by simulating the car movement between these positions.

We show how we find the entry positions, the dependency between entry positions, parking slot dimensions, and the number of backward-forward direction changes, and how to find the minimum dimensions of a parking slot for the given car. The results of our simulations can be used as parameters in parking assistant algorithms to quickly decide about the possibility of parking and simplify the navigation toward the parking slot.

The rest of the paper is structured as follows. We define the parallel parking problem in Section 2. In Section 3, we describe how we find the entry positions, how entry positions depend on direction changes, and how we compute the minimum dimensions of a parallel parking slot. In Section 4, we perform computational experiments and compare our results to related works. We conclude the paper in Section 5. The source code of our algorithm is publicly available as free and open source software[1].

### 1.1 Related Works

(Blackburn, 2009) explains how to compute the minimum length of a parallel parking slot, to which the car can park in only one maneuver. In this paper, we consider parallel parking slots that could be shorter and therefore we allow multiple backward-forward direction changes.

(Zips et al., 2013) introduce two-phased constraint optimization algorithm to plan the parking maneuver for parallel, perpendicular, and angle parking slot. During the first phase, the algorithm computes a path from the goal position to a phase switching point, from which the car can leave the slot. In the second phase, the algorithm finds a path from the switching point to the initial position. In their follow-up work

[a] https://orcid.org/0000-0002-6618-8152

[b] https://orcid.org/0000-0002-8738-075X

[c] https://orcid.org/0000-0002-8135-1296

[1] https://rtime.ciirc.cvut.cz/gitweb/hubacji1/bcar.git

Vlasak, J., Sojka, M. and Hanzálek, Z.
Parallel Parking: Optimal Entry and Minimum Slot Dimensions.
DOI: 10.5220/0011045600003191
In *Proceedings of the 8th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2022)*, pages 300-307
ISBN: 978-989-758-573-9; ISSN: 2184-495X

(Zips et al., 2016), they present improvements to the algorithm for a path planning outside the parking slot. Our paper differs in the problem definition. We consider the parallel parking slot – not the goal position – as the input. Our approach guarantees the optimality of entry positions as opposed to the phase switching point.

(Vorobieva et al., 2015) present three algorithms for geometric parking: (1) The algorithm for *parking in one maneuver* computes path from the goal position by simulating forward movement with the maximum steering. This algorithm works only for long enough parking slots. (2) Parking in *several parallel trials* algorithm starts by computing the path from initial position to some position partially inside the parking slot with the constraint of the car heading being parallel to the parking slot heading. Then, the algorithm continues by simulating forward-backward moves toward the parking slot with the constraint of the car heading being parallel to the parking slot heading when the car stops for the direction change. (3) Parking algorithm called *several reversed trials* generalize the algorithm for parking in one maneuver. It computes the path in the reverse order to the parking process: the algorithm starts from the goal position and computes the path by simulating forward-backward moves with the maximum steering until the car leaves the parking slot.

(Li and Tseng, 2016) propose complete system for automated parking. Their planning algorithm computes a path for a parallel parking slot from three circle segments. The initial position is considered a part of the input to the algorithm and only one (backward) move is allowed.

The works above describe a geometric planner to find a path between initial and goal position. In this paper, we also use the geometric approach. However, we consider entry and goal positions as the result, which allows us to guarantee that the car can park from entry positions into the parking slot with the minimum number of backward-forward direction changes. Path planning from initial to entry position is out of scope this paper.

(Li et al., 2016) present dynamic optimization framework for computing a time-optimal maneuvers for parallel parking. They compute trajectory from initial position to any parked position, i.e., a position of a car when the car is completely inside the parking slot. Compared to our approach, we do not consider the time. However, we compute a set of entry positions and return it as the result.

To solve the path planning problem of parallel parking for automated vehicle, (Jing et al., 2018) introduce nonlinear programming optimization that minimizes multiple objectives such as path length, distance from the car front to the parking slot front, and the distance from the car center to the parking slot center. The initial position of the car is fixed and only one (backward) move is allowed. In this paper, we allow multiple backward-forward direction changes, but we minimize their count.

# 2 PARALLEL PARKING PROBLEM

We define the used terminology and the parallel parking problem in this section.

## 2.1 Definitions

*Car position* is a tuple $C = (x, y, \theta, s, \phi)$, where $x$ and $y$ are cartesian coordinates of the rear axle center, $\theta$ is car heading, $s \in \{-1, +1\}$ is direction of the movement (backward and forward respectively), and $\phi \in [-\phi_{\max}, +\phi_{\max}]$ is car steering angle. Car positions are subject to a discrete kinematic model (Kuwata et al., 2009) $C_{k+1} = f(C_k)$, $k \in \mathbb{N}$ where function $f$ is given by Eq. (1):

$$\begin{aligned} x_{k+1} &= x_k + s_k \cdot \Delta \cdot cos(\theta_k) \\ y_{k+1} &= y_k + s_k \cdot \Delta \cdot sin(\theta_k) \\ \theta_{k+1} &= \theta_k + \frac{s_k \cdot \Delta}{b} \cdot tan(\phi_k), \end{aligned} \tag{1}$$

where $\Delta \in \mathbb{R}^+$ is a positive constant we call the *step distance*.

The *direction change* is a change of the car movement direction $s$. We denote $\Gamma$ the number of direction changes.

*Car dimensions* is a tuple $D = (w, d_{\mathrm{f}}, d_{\mathrm{r}}, b, \phi_{\max})$, where $w$ is the width of the car, $d_{\mathrm{f}}$ and $d_{\mathrm{r}}$ is the distance from the rear axle center to the front, respective back of the car, $b$ is the wheelbase (distance between the front and rear axle), and $\phi_{\max}$ is the maximum steering angle.

*Car frame* $\mathcal{F}(C)$ is a rectangle given by car position $C$ and car dimensions $D$.

*Minimum turning radius* is the radius $r$ of the circle traced by the rear axle center when the car moves with the maximum steering. It holds that $r = \frac{b}{\tan \phi_{\max}}$.

*Curb-to-curb distance* is the diameter $d$ of the circle traced by the outer front wheel when the car moves with the maximum steering. It holds that $d = 2 \cdot \sqrt{\left(r + \frac{w}{2}\right)^2 + b^2}$.

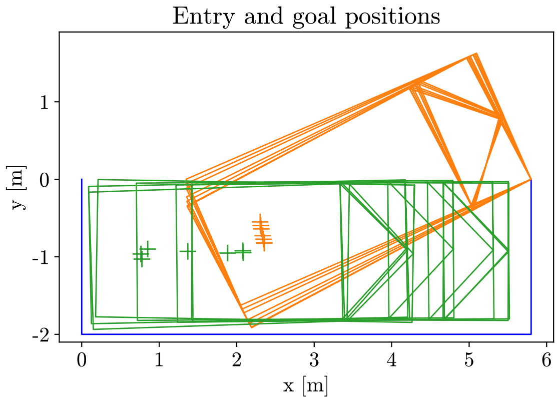

Figure 1: Orange rectangles represent an example subset of computed entry positions, green rectangles are goal positions, and blue frame is parallel parking slot. The cross represents the middle of the rear axle.

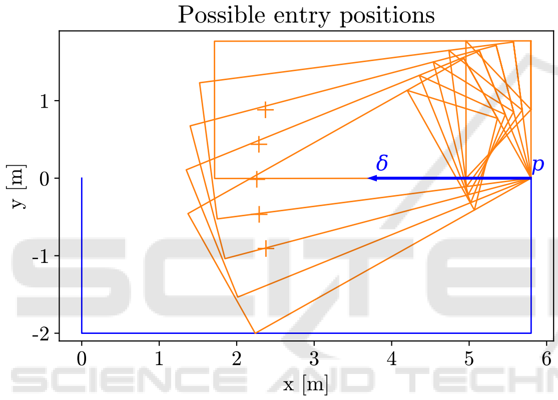

Figure 2: Orange rectangles represent an example subset of possible entry positions, blue frame is parallel parking slot. The cross represents the middle of the rear axle.

*Parallel parking slot* is a rectangle whose one side (that we call *entry side*) is adjacent to the road. *Parallel parking slot* is defined as a tuple $P = (p, \delta, W, L)$, where point $p \in \mathbb{R}^2$ is one corner of the rectangle on the entry side, $\delta$ is the direction vector of the entry side relative to $p$, $W > w$ is parking slot width greater than the width of the car, and the length of the parking slot, equal to the length of the entry side, is $L \geq d_f + d_r$.

*Goal position* is a car position $C_G$ for which $\mathcal{F}(C_G)$ is completely inside the parallel parking slot $P$. We can see an example of different green goal positions $C_G$ in Fig. 1.

*Possible entry position* for parallel parking slot positioned on the right side of a road (for left-sided parking slot the definition would be symmetric) is a car position $C_E^P = (x, y, \theta, -1, +\phi_{max})$ with the following properties: (i) right front corner of the car frame $\mathcal{F}(C_E^P)$ corresponds to the parking slot corner $p$, (ii) the car heading is in between the angle parallel to the entry side up to the angle perpendicular to the entry side, i.e., $\delta + \pi \leq \theta \leq \delta + 3/2\pi$, and (iii) car frame $\mathcal{F}(C_E^P)$ does not intersect with non-entry sides of the parking slot. We denote $\mathbf{C}_E^P$ a set of possible entry positions and we can see an example subset of $\mathbf{C}_E^P$ in Fig. 2.

## 2.2 Problem Statement

The main problem we solve is:

1. Given a parallel parking slot and car dimensions, find a set of entry positions $\mathbf{C}_E \subset \mathbf{C}_E^P$ from which the car can park into the slot with the minimum number of direction changes.

There are two additional related problems we solve along with the main one:

2. Given a parallel parking slot, car dimensions, and the maximum number of direction changes, what is a set of all entry positions $\mathbf{C}_E^A \subset \mathbf{C}_E^P$ from which the car can park into the slot?
3. Given the car dimensions and the maximum number of direction changes, what are the minimum dimensions (i.e., width and length) of a parallel parking slot the car can park into?

# 3 PARKING SIMULATION ALGORITHMS

In this section, we describe our parallel parking simulation algorithm, and we show how we use it to solve the problems from Section 2.2. In Section 3.1, we describe our algorithm for finding entry positions from which it is possible to park into the slot with the minimum number of direction changes, i.e., the main problem. In Section 3.2, we relax the condition on the minimum number of direction changes (problem 2), and in Section 3.3 we describe how we use the algorithm to find the minimum dimensions of a parallel parking slot (problem 3).

## 3.1 Finding Entry Positions

Our simulation algorithm first computes the set of possible entry positions $\mathbf{C}_E^P$. Then, the algorithm simulates backward-forward moves from each possible entry position $C_E^P \in \mathbf{C}_E^P$ until the car frame is completely inside the parking slot which mean the goal position is found. We can see the pseudocode in Algorithm 1. $\Delta_\theta$ is the increment to the car position heading for the consequent possible entry positions.

Algorithm 1: Finding entry positions.

- **Input:** car dimensions $D$, parking slot $P$
- **Output:** entry positions $\mathbf{C}_{\mathrm{E}}$

```
1:  $\mathbf{C}_0 \leftarrow \{C_{\mathrm{E}}^{\mathrm{P}}(D,P,\theta) | \theta = \delta + \pi + m \cdot \Delta_\theta, m \in \mathbb{N}\}$
2:  $\mathbf{H} \leftarrow \{(C_0, C_0, s, \phi, \Gamma) | C_0 \in \mathbf{C}_0,\ s = -1,\ \phi = \phi_{max}, \Gamma = 0\}$
3:  $\Gamma_{\max} \leftarrow \infty$
4:  $\mathbf{I} \leftarrow \emptyset$                    ▷ Set of goal and entry positions
5:  while $\mathbf{I} = \emptyset$ do
6:      for $(C, C_0, s, \phi, \Gamma) \in \mathbf{H}$ do
7:          if $\Gamma > \Gamma_{\max}$ then
8:              continue
9:          end if
10:         while $F(C)$ not intersects with $P$ do
11:             $C \leftarrow f(C, s, \phi)$          ▷ Simulate move
12:         end while
13:         if $F(C)$ inside $P$ then
14:             $\mathbf{I} \leftarrow \mathbf{I} \cup (C, C_0)$
15:             $\Gamma_{\max} \leftarrow \Gamma$
16:         else
17:             $\mathbf{H} \leftarrow \mathbf{H} \cup (C, C_0, -s, -\phi, \Gamma + 1)$
18:         end if
19:     end for
20: end while
21: return $\mathbf{C}_{\mathrm{E}} = \{C_0 | (C, C_0) \in \mathbf{I}\}$
```

### 3.2 Limit on the Direction Changes

Algorithm 1 finds the set of entry positions from which it is possible to park into the parking slot with the minimum number of direction changes. To find all entry positions $\mathbf{C}_{\mathrm{E}}^{\mathrm{A}} \subset \mathbf{C}_{\mathrm{E}}^{\mathrm{P}}$ for the given maximum number of direction changes, we provide $\Gamma_{\max}$ as the input to the Algorithm 1 and remove $\Gamma_{\max}$ update in Line 15.

By removing $\Gamma_{\max}$ update, the computed set of entry positions does not need to be continuous, i.e., the difference in headings of neighboring entry positions could be greater than $\Delta_\theta$. We can see such a situation in Fig. 1, where orange rectangles represent an example subset of computed entry positions where the continuity is broken approximately in the middle.

### 3.3 Minimum Slot Dimensions

To find the minimum dimensions (i.e., width and length) of a parallel parking slot for the given car dimensions and maximum number of direction changes $\Gamma_{\max}$, we run Algorithm 1 for finding entry positions repeatedly, looping over different parking slot widths and lengths. First, we initialize the parking slot width $W = w$ and length $L = d_{\mathrm{f}} + d_{\mathrm{r}}$ to match the car dimensions. Then, we gradually increase the parking slot width and length by the width step $\Delta_{\mathrm{W}}$ and the length step $\Delta_{\mathrm{L}}$ respectively.

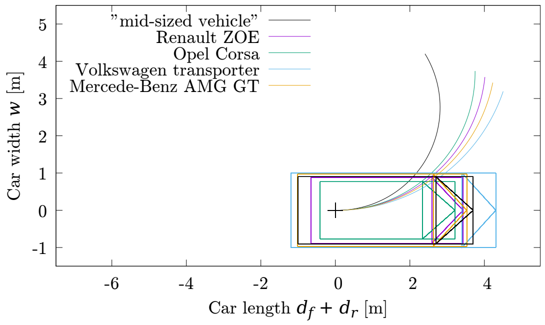

Figure 3: Car frame and the simulated path given by the forward movement with the maximum steering angle for "mid-sized vehicle", Renault ZOE, Opel Corsa, Volkswagen transporter, and Mercedes-Benz AMG GT.

## 4 RESULTS

In this section, we present the results of the computational experiments. In Section 4.1, we compare our approach to the related works. In Section 4.2, we show the range of all entry positions heading for the given maximum number of direction changes. Finally, we show the minimum dimensions of a parallel parking slot in Section 4.3.

We provide the results for five different cars. Their dimensions are shown in Table 1. In Fig. 3, we can see the car frames and the simulated paths given by the forward movement with the maximum steering angle. The width of the car is always without left and right rear view mirrors.

### 4.1 Comparison to Related Works

(Zips et al., 2013) use the dimensions of the "mid-sized vehicle" and the parallel parking slot with the width $W = 2.2\,\mathrm{m}$ and the length $L = 5.1\,\mathrm{m}$ for their simulations. There is no change to the first phase of their algorithm in their follow-up work.

Along with the parallel parking slot and car dimensions, (Zips et al., 2013) are given the initial and goal positions. There is no further information or exact values provided in their simulation studies. They need at least 12 direction changes to park in the slot.

We compute that it's possible to park into the slot they use with 10 direction changes for the given car dimensions, so we conclude that the used goal position is not optimal.

(Vorobieva et al., 2015) use the dimensions of the Renault ZOE for their experiments. They use *several*

Table 1: Car dimensions used in the computations.

| Car | $w$ [m] | $d_f$ [m] | $d_r$ [m] | $b$ [m] | $\phi_{max}$ [°] |
|---|---|---|---|---|---|
| "mid-sized vehicle" | 1.8 | 3.7 | 1.0 | 2.7 | 45 |
| Renault ZOE | 1.771 | 3.427 | 0.657 | 2.588 | 33 |
| Opel Corsa | 1.532 | 3.212 | 0.410 | 2.343 | 32 |
| Volkswagen transporter | 1.994 | 4.308 | 1.192 | 3.400 | 36 |
| Mercedes-Benz AMG GT | 1.939 | 3.528 | 1.016 | 2.630 | 32 |

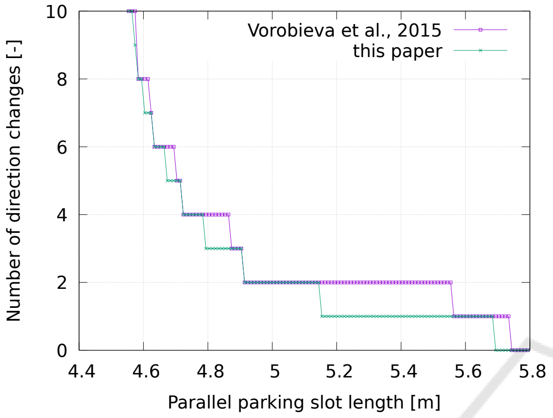

Figure 4: Comparison of direction changes for Renault ZOE when *several reversed trials* method is used with goal positions from (Vorobieva et al., 2015) or with goal positions from this paper. Goal positions from (Vorobieva et al., 2015) has the car frame aligned with the entry and rear sides of the parallel parking slot. The parallel parking slot has the width $W = 2.0\,\mathrm{m}$ and variable length of up to $L = 5.8\,\mathrm{m}$ with the length step of $\Delta_L = 0.01\,\mathrm{m}$.

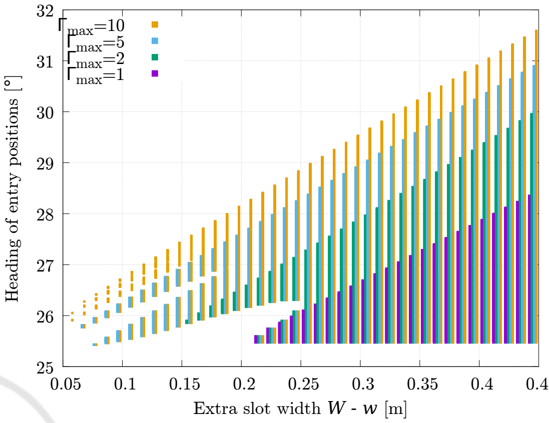

Figure 5: Entry positions heading for Renault ZOE, the parallel parking slot with different extra slot width $W - w$ with the width step of $\Delta_W = 0.01\,\mathrm{m}$, constant extra slot length $L - d_f - d_r = 1.1\,\mathrm{m}$, and different maximum number of direction changes $\Gamma_{max}$.

*reversed trials* method to plan a path into the parallel parking slot.

We compare our implementation of *several reversed trials* for goal positions used in (Vorobieva et al., 2015) with goal positions computed by our approach.

The goal position used in (Vorobieva et al., 2015) is not explicitly stated. They assume the goal position is known. For our computational experiments, we deduce the goal position from the context of the paper, i.e. the goal position for parallel parking slot positioned on the right side of a road has the following properties: (i) left side of the car frame given by the goal position corresponds to parking slot entry side, and (ii) rear side of the car frame given by the goal position corresponds to parking slot rear side.

The parallel parking slot has the width $W = 2.0\,\mathrm{m}$, and we test the length of up to $L = 5.8\,\mathrm{m}$ with the length step of $\Delta_L = 0.01\,\mathrm{m}$.

In Fig. 4, we can see that when goal position is given by our approach, it leads to less direction changes.

(Li et al., 2016) use the dimensions of the Renault ZOE for their simulations. The parallel parking slot has the width $W = 2.0\,\mathrm{m}$ and the length $L = 4.5\,\mathrm{m}$. (Li et al., 2016) need 19 direction changes for the given parallel parking slot and car dimensions. Our approach results in 19 direction changes for the given values, too.

## 4.2 Limit on the Direction Changes

To compute the set of all entry positions for the given maximum number of direction changes, we use the car dimensions of the Renault ZOE.

In Fig. 5, we can see the entry positions heading for the parallel parking slot with constant extra slot length $L - d_f - d_r = 1.1\,\mathrm{m}$ and different maximum number of direction changes $\Gamma_{max} = 1$, 2, 5, and 10 respectively.

We can see that the range of entry positions heading is wider for larger extra slot width $W - w$ and for increasing $\Gamma_{max}$. Also, we can see that the entry positions are found for narrower slots for higher $\Gamma_{max}$. Finally, the set of entry positions does not need to be continuous, i.e., the difference between two neighboring entry positions could be greater than $\Delta_\theta$. The range of entry positions heading then consists of the subranges.

In Fig. 6, we can see the number of entry positions

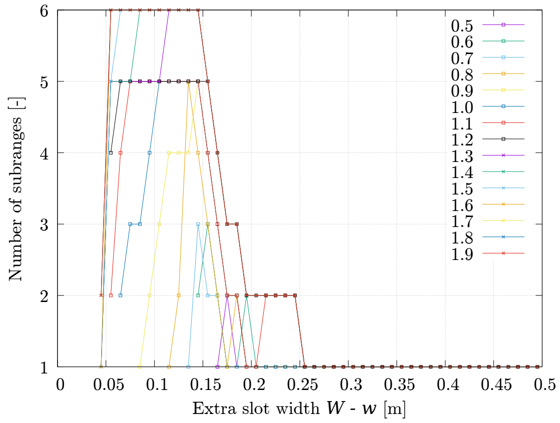

Figure 6: The number of entry positions heading subranges for Renault ZOE, the parallel parking slot with different extra slot width $W - w$ with the width step of $\Delta_W = 0.01\,\mathrm{m}$, extra slot length $L - d_f - d_r$ from 0.5 m up to 1.9 m with the length step of $\Delta_L = 0.1\,\mathrm{m}$, and constant $\Gamma_{\max} = 10$.

heading subranges for the parallel parking slot with different extra slot lengths $L - d_f - d_r$ from 0.5 m up to 1.9 m and constant $\Gamma_{\max} = 10$.

We can see that there is only one range of entry positions heading for wide enough slots. The number of subranges increases for narrower slots but decreases again for the slots that are too narrow.

## 4.3 Minimum Slot Dimensions

To compute the minimum dimensions of a parallel parking slot, we use the possible entry position heading step $\Delta_\theta = 10^{-4}\,\mathrm{rad}$, parking slot width step $\Delta_W = 0.01\,\mathrm{m}$, and the length step $\Delta_L = 0.01\,\mathrm{m}$.

In Fig. 7, we can see the computed minimum dimensions of a parallel parking slot for Renault ZOE and different maximum number of direction changes $\Gamma_{\max}$.

Renault ZOE needs to perform at least $\Gamma = 30$ direction changes when parking into the parallel parking slot with extra slot length $L - d_f - d_r = 0.4\,\mathrm{m}$. Also, there is a negligible improvement of 0.02 m when increasing the number of direction changes to $\Gamma = 100$.

In Fig. 8, we can see the minimum dimensions of a parallel parking slot for Opel Corsa, Volkswagen transporter, and Mercedes-Benz AMG GT.

Opel Corsa can park into the parallel parking slot with extra slot length of 0.4 m with $\Gamma = 10$ direction changes. Volkswagen transporter needs the same number of direction changes $\Gamma = 30$ as the Renault ZOE, although it requires more extra slot width. It is not possible to park Mercedes-Benz AMG GT into the parallel parking slot with extra slot length of 0.4 m within the reasonable number of backward-forward direction changes.

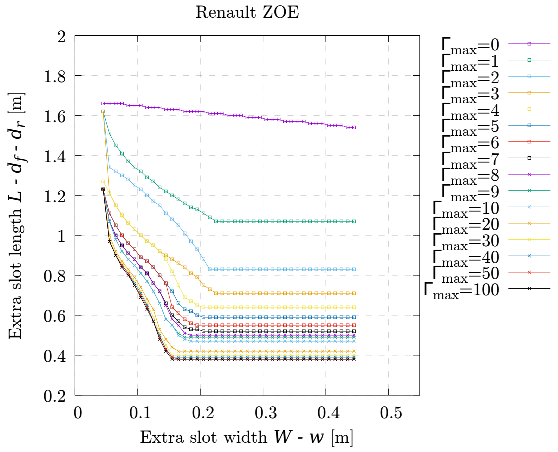

Figure 7: Computed minimum dimensions of a parallel parking slot for Renault ZOE and different maximum number of direction changes. The first line at the top of the graph shows the minimum dimensions when no direction change is allowed ($\Gamma_{\max} = 0$), followed by the maximum number ($\Gamma_{\max}$) of 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, and 100 direction changes. The width step $\Delta_W = 0.01\,\mathrm{m}$ and the length step $\Delta_L = 0.01\,\mathrm{m}$.

The improvement to extra slot length for $\Gamma > 20$ is negligible, i.e., in the order of $10^{-2}\,\mathrm{m}$. The shape of the slot's minimum dimensions curve depends on whether the number of backward-forward direction changes is even or odd.

Fig. 9 shows the minimum dimensions of a parallel parking slot for all the cars. The maximum number of direction changes shown in the figure is $\Gamma_{\max} = 1$, 2, 5, and 10 respectively.

# 5 CONCLUSION

In this paper, we re-formulate the path planning problem for parallel parking: Given a parallel parking slot and car dimensions, we compute a set of entry positions to the parking slot from which it is possible to park into the slot with the minimum number of backward-forward direction changes. We also use our approach to compute the minimum dimensions (i.e., width and length) of a parallel parking slot for the given car dimensions.

Our algorithm is designed for offline use. By precomputing entry positions for different parallel parking slots and car dimensions and using them in online planning, we can speed up the path planning as it does not need to plan the path inside the parking slot. Parking assistants using our results can ensure that when the automated vehicle reaches one of the entry posi-

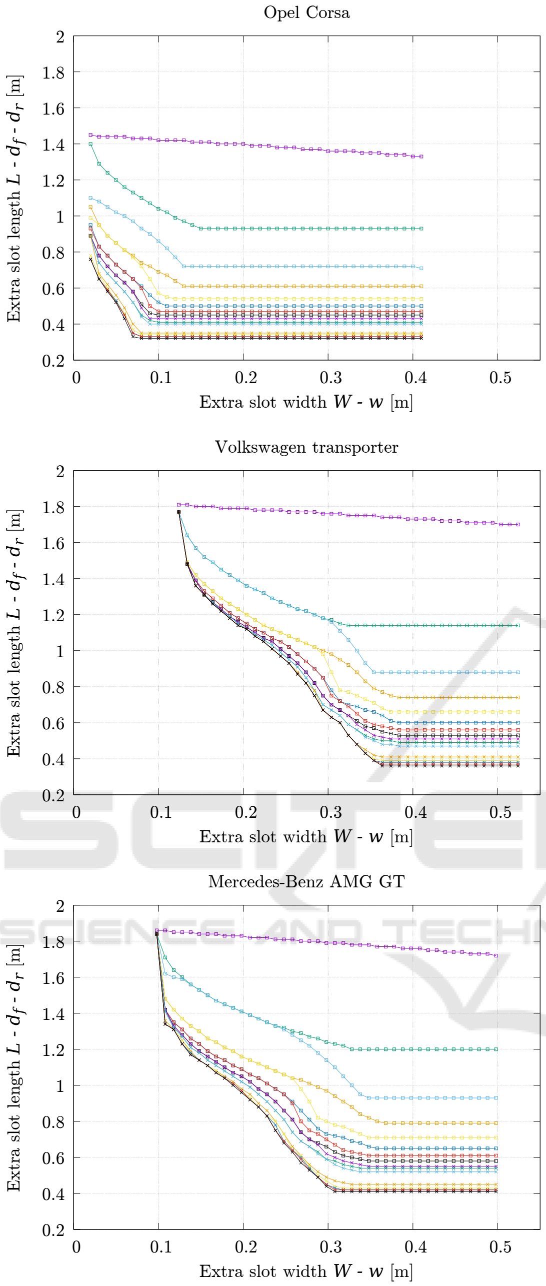

Figure 8: Computed minimum dimensions of a parallel parking slot for Opel Corsa, Volkswagen transporter, Mercedes-Benz AMG GT, and different maximum number of direction changes.

tions, it can optimally park into the parallel parking slot.

The range of entry positions from which the given car can park optimally depends on the slot dimensions. The range is wider for larger slots and a higher number of backward-forward direction changes as shown in Section 4.

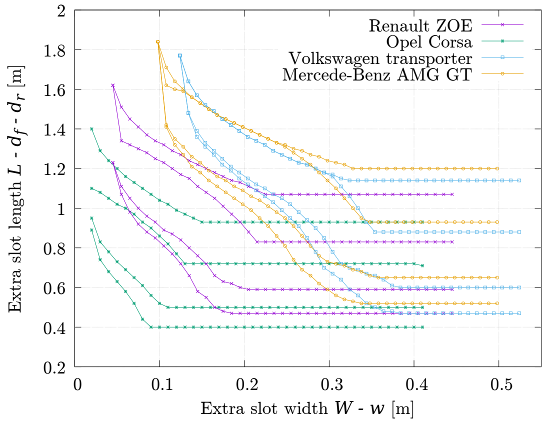

Figure 9: Computed minimum dimensions of a parallel parking slot for Renault ZOE, Opel Corsa, Volkswagen transporter, and Mercedes-Benz AMG GT. The width step $\Delta_W = 0.01\,\mathrm{m}$ and the length step $\Delta_L = 0.01\,\mathrm{m}$. The maximum number of direction changes is $\Gamma_{\max} = 1, 2, 5$, and 10 respectively.

By precomputing minimum dimensions of different parallel parking slots, we can simplify the decision of whether it is possible to park into the slot.

Using more than 20 direction changes does not lead to significant reduction of the parking slot size. The shape of the slot's minimum dimensions curve depends on whether the number of backward-forward direction changes is even or odd.

## ACKNOWLEDGEMENTS

Research leading to these results has received funding from the EU ECSEL Joint Undertaking and the Ministry of Education of the Czech Republic under grant agreement 826452, 8A19011, and MSMT-24623/2019-4/11 (project Arrowhead Tools).

## REFERENCES

Blackburn, S. R. (2009). The Geometry of Perfect Parking | Department Of Mathematics, Royal Holloway, University of London.

Jing, W., Feng, D., Zhang, P., Zhang, S., Lin, S., and Tang, B. (2018). A Multi-Objective Optimization-based Path Planning Method for Parallel Parking of Autonomous Vehicle via Nonlinear Programming. In *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1665–1670.

Kuwata, Y., Teo, J., Fiore, G., Karaman, S., Frazzoli, E., and How, J. P. (2009). Real-time motion planning with applications to autonomous urban driving. *IEEE Trans-*

*actions on Control Systems Technology*, 17(5):1105–1118.

Li, B., Wang, K., and Shao, Z. (2016). Time-Optimal Maneuver Planning in Automatic Parallel Parking Using a Simultaneous Dynamic Optimization Approach. *IEEE Transactions on Intelligent Transportation Systems*, 17(11):3263–3274.

Li, M. H. and Tseng, P. K. (2016). Implementation of an autonomous driving system for parallel and perpendicular parking. In *Proc. IEEE/SICE Int. Symp. System Integration (SII)*, pages 198–203.

Vorobieva, H., Glaser, S., Minoiu-Enache, N., and Mammar, S. (2015). Automatic Parallel Parking in Tiny Spots: Path Planning and Control. *IEEE*, 16(1):396–410.

Zips, P., Böck, M., and Kugi, A. (2016). Optimisation based path planning for car parking in narrow environments. *Robotics and Autonomous Systems*, 79:1–11.

Zips, P., Bock, M., and Kugi, A. (2013). A fast motion planning algorithm for car parking based on static optimization. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2392–2397. ISSN: 2153-0866.