An Automated Parallel Parking Strategy Using Reach Control Theory [★]

Melkior Ornik [*] **Miad Moarref** [*] **Mireille E. Broucke** [*]

[*] *Department of Electrical & Computer Engineering, University of Toronto, Toronto, ON M5S 3G4 Canada (e-mail: {melkior.ornik@scg.utoronto.ca, miad.moarref@utoronto.ca, broucke@control.utoronto.ca}).*

Abstract: We propose a novel method of parallel parking using reach control theory. Reach control is a hybrid control method to achieve complex control objectives. It relies on triangulating the state space, devising a desired sequence of simplices or polytopes that a trajectory needs to pass through to complete a task, and then constructing a separate closed-loop controller on each polytope that enables the system state to move on to the next member of the sequence. For the parallel parking task, we design a state space consisting of eight polytopes, and we use an automated procedure to construct a continuous piecewise-affine controller for each polytope. Extensive simulations demonstrate the robustness of the approach: a vehicle starting from an acceptable initial position performs the maneuver safely and comes to a stop in the desired parking area.

Keywords: Switching Control, Automated Guided Vehicles, Piecewise Affine Feedback, Parallel Parking, Safety Analysis

1. INTRODUCTION

Automated parallel parking of a vehicle is a widely studied problem in control theory (Lyon, 1992; Murray and Sastry, 1993; Tilbury et al., 1995; Pohl et al., 2009; Demirli and Khoshnejad, 2009; Liang et al., 2012; Chand et al., 2015). The problem seeks to automate the maneuver often used by human drivers to park a car in a limited space between two cars previously parked parallel to the curb. The control strategies predominantly used in previous literature consist of calculating a feasible path that a vehicle should follow during the parallel parking maneuver, starting from a given initial position, and then computing the appropriate steering control to follow this path. This approach is not robust, as a slight deviation from the initial position or a disturbance during the parallel parking maneuver requires a new path to be calculated, if such a path can be calculated at all. We propose a different method using a reach control approach.

The current formulation of the Reach Control Problem (RCP) was given simultaneously by Roszak and Broucke (2006) and Habets et al. (2006). Given an affine system $\dot{\boldsymbol{x}} = A\boldsymbol{x} + Bu + a$ on a polytope $\mathcal{P}$, the RCP seeks to find a closed-loop controller which drives the system states to leave a polytope $\mathcal{P}$ through a predetermined facet $\mathcal{F} \subset \mathcal{P}$. This is a stepping stone to a hybrid system approach to achieve complex control objectives. The methodology of reach control sits within a larger framework where complex control objectives are specified using linear temporal logic (Kloetzer and Belta, 2008). In the case of parallel parking, the system state is required to move from one area of the state space to another. Our approach is to split the state space into polytopes and then devise a sequence of polytopes leading from the initial state to a desired endpoint. On each of these polytopes, a closed-loop controller will be devised. It will drive the system state to leave the polytope through the exit facet that connects it to the next polytope in the sequence.

Robustness and simple controller design make the RCP amenable to a number of applications, including quadcopter motion (Vukosavljev et al., 2016), aircraft control (Belta and Habets, 2006), robotic manipulators (Martino and Broucke, 2014), and aggressive maneuvers of mechanical systems (Vukosavljev and Broucke, 2014). The last work is particularly relevant to the present problem. It deals with driving a gantry crane from one area of the state space to another while avoiding an obstacle. The model used by Vukosavljev and Broucke (2014) is underactuated. This is the case for the vehicle model in our investigation as well. However, these two works have substantial differences: Vukosavljev and Broucke (2014) linearize their nonlinear model around a single point, whereas we seek to ensure the faithfulness of our model by constructing a separate linearization for each region of the partitioned state space. This idea was discussed by Girard and Martin (2012) but was, to our knowledge, never previously fully investigated in an application. Additionally, while the state space of Vukosavljev and Broucke (2014) was four-dimensional, the actual controller design was performed in the 2D output space. In contrast, our design is performed on the full three-dimensional state space.

[★] This research is supported by The Natural Sciences and Engineering Research Council of Canada.

2. MODEL AND PROBLEM STATEMENT

We use the standard front-wheel drive car model based on the unicycle described by, e.g., De Luca et al. (1998). The model uses an external reference frame (x, y, θ), where the x-axis is parallel to the curb, the y-axis is perpendicular to it and pointing into the road, and θ is the orientation of the car with respect to the curb. The car's position (x, y) in this frame of reference is determined by the midpoint of its rear axle. Let v denote the forward-moving speed of the car, and let φ denote the angle in degrees of the front wheels with respect to the orientation of the car, where $\varphi = 0°$ if the front wheels are aligned with the car.

The model equations are given as follows:

$$\begin{aligned} \dot{x} &= v\cos(\varphi)\cos(\theta) \\ \dot{y} &= v\cos(\varphi)\sin(\theta) \\ \dot{\theta} &= \frac{v}{L}\sin(\varphi), \end{aligned} \tag{1}$$

where L is the distance between the front wheels and the back wheels (wheelbase). The exact dimensions of the car are taken from Audi UK (2013), and are given in an online document (Ornik et al., 2017). The maximum steering angle of the front wheels is taken to be $\varphi_M = 33°$. Because of safety and liveness concerns, we take the lowest possible speed of the vehicle $v_S = -5\text{km/h} \approx -1.39\text{m/s}$ and the highest speed $v_F = -10\text{km/h} \approx -2.78\text{m/s}$.

We assume the length of the parking space to be around 1.9 times the length of the car. The two already parked cars are assumed to be equal in size to the car we are parking, and the roadway is assumed to be clear of any obstacles apart from the two cars. An illustration of the available parking space is given in Figure 1. We note that the length of the parking space allows a comfortable parallel parking maneuver. We will show that our approach does not use the entire area of the parking space.

In order to describe the vehicle's initial position, ending position, and obstacles, we define the following four sets:

- $R(\boldsymbol{x}) \subset \mathbb{R}^2$ is the set of all physical positions covered by a car whose position of the midpoint of its rear axle and orientation are given by $\boldsymbol{x}$.
- $\mathcal{I} \subset \mathbb{R}^2$ is the set of allowed initial positions of the vehicle. We assume its dimensions to be $0.5\text{m} \times 0.25\text{m}$, and we assume it to be located so that the back of the car is between 0.5m and 0m behind the back of the first parked car, and the side of the car is located between 0.75m and 1m from the side of the parked cars.
- $\mathcal{E} \subset \mathbb{R}^2$ is the set of all desirable ending positions of the vehicle. It is situated so that no part of the car is colliding with another car or a curb, and that the side of the car is no more than 0.3m from the curb.
- $\mathcal{G} \subset \mathbb{R}^2$ is the set of all obstacles. It consists of the two previously parked vehicles and the sidewalk.

We note that the size of $\mathcal{E}$ imposes a very tight criterion for correct parking. An illustration of $\mathcal{I}$, $\mathcal{E}$, $\mathcal{G}$ is given in Figure 2.

We now formalize the problem we will be solving in the paper.

Problem 1. Let $R(\boldsymbol{x}), \mathcal{I}, \mathcal{E}, \mathcal{G} \subset \mathbb{R}^2$ be as above. Let $U = [v_F, v_S] \times [-\varphi_M, \varphi_M]$. Consider the model (1). Find a connected set $\mathcal{M} \supset \mathcal{I} \times \{0\}$ and a feedback controller $u : \mathcal{M} \to U$ such that for all $\boldsymbol{x}_0 \in \mathcal{I} \times \{0\}$ there exists $T \geq 0$ such that the following holds:

(i) $\phi(T, \boldsymbol{x}_0) \in \mathcal{E} \times \{0\}$,
(ii) $\phi(t, \boldsymbol{x}_0) \in \mathcal{M}$ for all $t \in [0, T]$,
(iii) $R(\phi(t, \boldsymbol{x}_0)) \cap \mathcal{G} = \emptyset$ for all $t \in [0, T]$.

Problem 1 is of the reach-avoid type. Note that the initial condition set is $\mathcal{I} \times \{0\}$ and the final condition set is $\mathcal{E} \times \{0\}$, so the car must be parallel to the curb at the start and the end of the maneuver.

3. METHODOLOGY

In this section, we propose a solution to Problem 1 using reach control. We will construct a sequence of polytopes in the state space such that the first polytope contains the initial states of the car, the feedback control on each polytope pushes the system state into the next one, and the last polytope ends in a acceptable ending state for the parking maneuver.

The solution is motivated by the steps human drivers follow to park the car. Using the notation in Figure 1, the standard way to execute this maneuver consists of four steps:

(1) Bring vehicle C to a stop in parallel to the curb, and next to vehicle A.
(2) Start going in reverse, with the front wheels rotated as far to the right as possible until vehicle C is roughly at a 45° angle to the curb.
(3) Straighten out the wheels and continue going in reverse until vehicle C is situated almost behind vehicle A.
(4) Continue going in reverse, with the front wheels rotated as far to the left as possible until vehicle C is parallel to the curb.

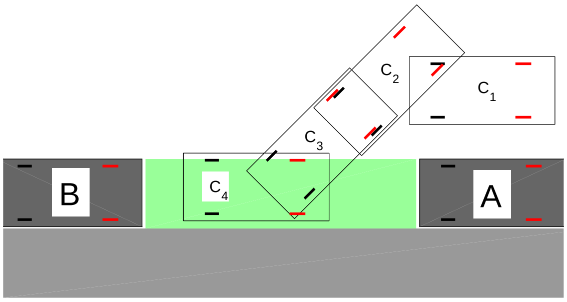

Fig. 1. An illustration of the parallel parking maneuver. Vehicles A and B, marked in dark gray, are previously parked in parallel with the curb (marked in light gray), and at a short distance from it. Vehicle C seeks to park between vehicle A and vehicle B. The front wheels of all vehicles are coloured red, while the rear wheels are coloured black. States C$_i$, $i = 1, \ldots, 4$ denote the position and orientation of vehicle C at the end of Step i of the maneuver described above.

In the remainder of the paper, we assume that a human driver has already completed Step 1 by bringing the car to a standstill in state $\boldsymbol{x}_0 \in \mathcal{I} \times \{0\}$. The steps we want to perform are 2–4. Our approach to designing a control strategy which completes this maneuver consists of the following elements:

- Design a sequence of polytopes $\mathcal{P}_i$ constructed around an ideal but possibly infeasible car trajectory, with the goal of having the car remain close to the ideal trajectory by staying within the polytopes $\mathcal{P}_i$.
- Use the necessary conditions for the solvability of the RCP to design appropriate linearization points at each polytope $\mathcal{P}_i$, in order to turn the nonlinear model (1) into an affine system amenable to an RCP approach.
- Use the sufficient conditions for the solvability of the RCP to determine the exact size of each polytope $\mathcal{P}_i$.
- Use reach control theory to design a piecewise-affine controller on each of the polytopes $\mathcal{P}_i$.

3.1 Polytopes

An intuitive idea of what the position (x, y) and orientation θ of the car "should" look like during steps 2–4 is given in Figure 2. The trajectory in Figure 2 is not necessarily feasible for the system (1) or its linearization. Moreover, it is a trajectory for only one initial condition, whereas our goal is to perform a successful parking maneuver starting from any point in $\mathcal{I} \times \{0\}$. Our approach is to design a control strategy to ensure that actual trajectories of cars starting at any point $\boldsymbol{x}_0 \in \mathcal{I} \times \{0\}$ remain similar to this imaginary trajectory. In order to do that, we will embed the trajectory in Figure 2 into a sequence of polytopes $\mathcal{P}_1, \ldots, \mathcal{P}_m$ and ensure that all trajectories starting from initial states in $\mathcal{I} \times \{0\}$ remain inside those polytopes.

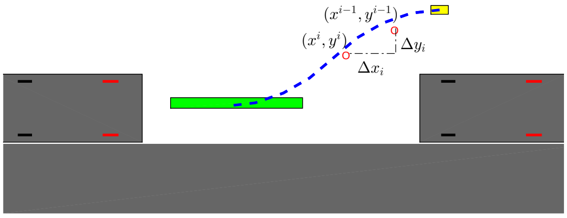

Fig. 2. An illustration of an imaginary ideal path for the parking maneuver. The set of obstacles $\mathcal{G}$ is drawn in gray. The initial box $\mathcal{I}$ for the midpoint of the car's rear axle is drawn in yellow. The acceptable ending box $\mathcal{E}$ is drawn in green.

Let $\mathcal{I} = [x_{11}, x_{12}] \times [y_{11}, y_{12}]$. We now describe the construction of polytopes $\mathcal{P}_1, \ldots, \mathcal{P}_m$. The process is as follows:

- Grid up the interval $[0°, 45°]$ into intervals $[\theta^0, \theta^1]$, $[\theta^1, \theta^2]$, ..., $[\theta^{k-1}, \theta^k]$, where $\theta^0 = 0°$ and $\theta^k = 45°$, so that the (x, y) trajectory of the car in Figure 2 is nearly linear during the time its orientation is between θ^i and θ^{i+1}. We found that $k = 4$ and
$$\theta^i = i \cdot 11.25°, \quad i = 0, 1, \ldots, 4,$$
are suitable.
- Based on a measurement from Figure 2, let (x^i, y^i) be the position of the car when its orientation is $\theta = \theta^i$. We compute
$$\Delta x_i = x^i - x^{i-1}, \quad \Delta y_i = y^i - y^{i-1}.$$
In the interest of space, we omit listing coefficients $\Delta x_i, \Delta y_i$. All are available online (Ornik et al., 2017).
- We now start with $\theta \in [0, \theta^1]$. At $\theta = 0°$, we know that the position of the car is inside $\mathcal{I}$. At $\theta = \theta^1$, the position of the car should be inside $\mathcal{I} + (\Delta x_1, \Delta y_1)$, based on Figure 2. Thus, an ideal polytope $\mathcal{P}_1'$ would be constructed by taking the convex hull of faces $\mathcal{F}_1^{\text{in}} = \mathcal{I} \times \{0\}$ and $\mathcal{F}_1^{\text{out}'} = (\mathcal{I} + (\Delta x_i, \Delta y_i)) \times \{\theta^1\}$. Unfortunately, this will not work because the ideal trajectory in Figure 2 is not necessarily feasible, and the trajectories are not linear.
- Thus, we introduce *widening coefficients* $w_{1x}, w_{1y} \geq 0$, and we construct a modified polytope $\mathcal{P}_1$ to be the convex hull of $\mathcal{F}_1^{\text{in}} = \mathcal{I} \times \{0\}$ and $\mathcal{F}_1^{\text{out}} = [x_{11} + \Delta x_1 - w_{1x}, x_{12} + \Delta x_1 + w_{1x}] \times [y_{11} + \Delta y_1 - w_{1y}, y_{12} + \Delta x_1 + w_{1y}] \times \{\theta^1\}$. In particular, $\mathcal{F}_i^{\text{out}}$ is now widened.
- We generate $\mathcal{P}_2, \ldots, \mathcal{P}_k$ inductively, where $\mathcal{P}_i$ is the convex hull of the facets given by
$$\mathcal{F}_i^{\text{in}} = \mathcal{F}_{i-1}^{\text{out}},$$
$$\mathcal{F}_i^{\text{out}} = [x_{i+1,1}, x_{i+1,2}] \times [y_{i+1,1}, y_{i+1,2}] \times \{\theta^i\},$$
where
$$\begin{aligned} x_{i1} &= x_{i-1,1} + \Delta x_{i-1} - w_{i-1,x}, \\ x_{i2} &= x_{i-1,2} + \Delta x_{i-1} + w_{i-1,x}, \\ y_{i1} &= y_{i-1,1} + \Delta y_{i-1} - w_{i-1,y}, \\ y_{i2} &= y_{i-1,2} + \Delta y_{i-1} + w_{i-1,y}. \end{aligned}$$

Notice that the "in facet" of $\mathcal{P}_i$ is the "out facet" of $\mathcal{P}_{i-1}$, and $\mathcal{F}_i^{\text{out}}$ is again widened as above. See Figure 3 for an illustration of the above procedure.

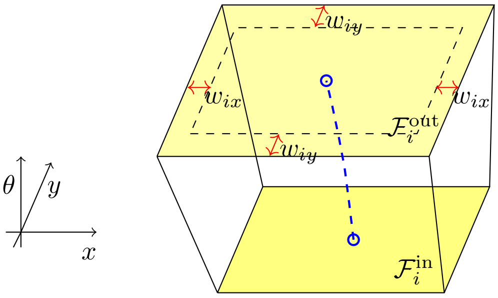

Fig. 3. An illustration of polytope $\mathcal{P}_i$. The polytope resembles a parallelepiped with $\mathcal{F}_i^{\text{in}}$ and $\mathcal{F}_i^{\text{out}}$ as its bases. The imaginary ideal trajectory of the car while going through polytope $\mathcal{P}_i$ is given by a blue dashed line. Widening coefficients w_{ix}, w_{iy} are represented by red arrows.

The above process corresponds to Step 2 of the maneuver. Step 3, in which the orientation θ of the vehicle does not change significantly, is then interpreted by repeating the same process as above, just with $\theta^i = \theta^{i-1} = 45°$, until some polynomial $\mathcal{P}_l$. After that, Step 4 is again performed in the same way as Step 2, just with $0° < \theta^i < \theta^{i-1}$, until $\theta^m = 0°$ for some $m \geq l$. In our case, the entire construction is performed with just eight polytopes: $k = 4$, $l = k = 4$, and $m = 8$. This means that Step 3 is short enough to not make a significant impact in the parking maneuver. Orientations $\theta^5, \ldots, \theta^8$ are chosen by
$$\theta^i = 90° - i \cdot 11.25°, \quad i = 5, 6, 7, 8.$$

We note that we still did not select widening coefficients w_{ix}, w_{iy}. The widening coefficients need to be selected so that there exists a controller driving all trajectories from $\mathcal{P}_i$ to leave $\mathcal{P}_i$ through the facet $\mathcal{F}_i^{\text{out}}$. Since we will be using reach control theory to construct such a controller, the choice of widening coefficients will follow from the solvability conditions of RCP. After we elucidate these conditions in the following section, we will provide the appropriate widening coefficients in Section 3.4.

3.2 Reach Control Problem

We revisit the theory of solvability of the RCP. Let $\mathcal{P} \subset \mathbb{R}^n$ be a convex polytope, and let $\mathcal{F}_0 \subset \mathcal{P}$ be its designated exit facet. Let us consider system $\dot{\boldsymbol{x}} = A\boldsymbol{x} + Bu + a$ on $\mathcal{P}$. In order to drive the system state not to leave through any facets other than $\mathcal{F}$, all the velocity vectors $A\boldsymbol{x}+Bu+a$ on the boundary of $\mathcal{P}$ need to point inside $\mathcal{P}$, except perhaps on $\mathcal{F}$, where they are allowed to point through $\mathcal{F}$. This is formally encoded in the following way: let h_i, $i \in \{1, \ldots, r\}$ be the unit vector normal to $\mathcal{F}_i$ and pointing outside of polytope $\mathcal{P}$. For each $\boldsymbol{x} \in \mathcal{P}$ we define the cone

$$\mathcal{C}(\boldsymbol{x}) = \{y \in \mathbb{R}^n \mid h_j \cdot y \leq 0 \text{ if } \boldsymbol{x} \in \mathcal{F}_j, 1 \leq j \leq r\}, \quad (2)$$

where $h_j \cdot y$ denotes the inner product of the two vectors. In particular, $\mathcal{C}(\boldsymbol{x}) = \mathbb{R}^n$ for all $\boldsymbol{x}$ in the interior of $\mathcal{P}$.

Definition 2. If for all $\boldsymbol{x} \in \mathcal{P}$ there exists u such that $A\boldsymbol{x}+Bu+a \in \mathcal{C}(\boldsymbol{x})$, we say that *the invariance conditions are solvable.*

It was shown by Habets and van Schuppen (2004) that solvability of the invariance conditions is a necessary condition for solvability of the RCP by continuous state feedback. Additionally, Habets and van Schuppen (2004) show that solvability of invariance conditions on the entire polytope $\mathcal{P}$ is equivalent to finding u_i such that $Av_i + Bu_i + a \in \mathcal{C}(v_i)$ for all vertices $v_1, \ldots, v_p \in \mathcal{P}$. As $\mathcal{C}(v_i)$ is a closed, convex cone, this is a linear feasibility problem.

Another necessary condition for a system to leave $\mathcal{P}$ through a designated facet $\mathcal{F}$ is the lack of equilibria in $\mathcal{P}$. Clearly, if $\dot{\boldsymbol{x}} = A\boldsymbol{x} + Bu(\boldsymbol{x}) + a = 0$ for some $\boldsymbol{x} \in \mathcal{P}$, the trajectory starting in $\boldsymbol{x}$ will not leave $\mathcal{P}$ at all. $A\boldsymbol{x} + Bu(\boldsymbol{x}) + a = 0$ implies $A\boldsymbol{x} + a = B(-u(\boldsymbol{x}))$, so we define

$$\mathcal{O}_{\mathcal{P}} = \{\boldsymbol{x} \in \mathcal{P} \mid A\boldsymbol{x} + a \in \text{Im}(B)\}.$$

$\mathcal{O}_{\mathcal{P}}$ is the set of all potential equilibria of the control system $\dot{\boldsymbol{x}} = A\boldsymbol{x} + Bu + a$ on $\mathcal{P}$. It was shown by Helwa and Broucke (2013) that, if $\mathcal{O}_{\mathcal{P}} = \emptyset$, solvability of the invariance conditions is in fact a sufficient condition for solvability of the RCP by continuous piecewise affine feedback. In the following section we will linearize system (1) in such a way that $\mathcal{O}_{\mathcal{P}_i} = \emptyset$ for all $i \in \{1, \ldots, m\}$.

3.3 Linearization

In order to make (1) amenable to an RCP approach, we linearize it around $(x_0, y_0, \theta_0, \varphi_0, v_0)$, and assume $\varphi_0 = 0$. Note that we do not require $(x_0, y_0, \theta_0, \varphi_0, v_0)$ to be an equilibrium for system (1). This results in the linearized system having affine form

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + Bu + a \quad (3)$$

which fits within the setting of reach control. In order to make the system (3) as faithful to (1) as possible, we will choose a different linearization point for each $\mathcal{P}_i$, $i \in \{1, \ldots, m\}$. First, we want to choose $(x_0, y_0, \theta_0, v_0, \varphi_0)$ in such a way that $\mathcal{O}_{\mathcal{P}_i} = \emptyset$. The choice of x_0, y_0 is irrelevant as they do not appear on the right side of (1). Additionally, as $v_0 \in [v_S, v_F]$ and $\varphi_0 \in [-\varphi_M, \varphi_M]$, it is natural to take $v_0 = (v_S + v_F)/2$, $\varphi_0 = 0$. Going back to the affine model (3), we can now easily calculate A, B, and a, and establish that $\mathcal{O}_{\mathcal{P}_i} = \{(x, y, \theta) \in \mathcal{P}_i \mid \theta = \theta_0\}$.

Thus, if we choose θ_0 so that $\mathcal{P}_i$ does not contain any points (x, y, θ) with $\theta = \theta_0$, it is guaranteed that there are no equilibria in $\mathcal{P}$. By construction in Section 3.1, the orientations θ in $\mathcal{P}_i$ are between θ^{i-1} and θ^i, with the convention that $\theta^0 = 0$. As we want the linearization (3) of the system to be as faithful to (1) as possible, we want θ_0 close to $[\theta^{i-1}, \theta^i]$. Thus, we will choose θ_0 for each polytope $\mathcal{P}_i$ so that $\theta_0 < \min(\theta^{i-1}, \theta^i)$, with $\min(\theta^{i-1}, \theta^i) - \theta_0 = \varepsilon > 0$ as small as possible. In our simulations, we chose $\varepsilon = 10^{-10}$.

3.4 Choice of Widening Coefficients

Since $\mathcal{O}_{\mathcal{P}_i} = \emptyset$, we know from Helwa and Broucke (2013) that if the invariance conditions $Ax + Bu(x) + a \in \mathcal{C}(x)$ are solvable on the vertices $v \in \mathcal{P}_i$, the RCP is solvable using continuous piecewise feedback. The vertices of $\mathcal{P}_i$ were defined in Section 3.1, but we did not yet give the exact values for w_{ix}, w_{iy}. It is intuitive to expect that the larger w_{ix}, w_{iy} we choose, the invariance conditions will be more likely to be solvable, as we allow the car to deviate more from our ideal trajectory. The trade-off is that choosing larger w_{ix}, w_{iy} results in larger polytopes $\mathcal{P}_i$. Unfortunately, there is no developed theory for the choices of w_{ix}, w_{iy}. The parameters that worked in our case are listed in an online document (Ornik et al., 2017).

We have now fully defined our sequence $\mathcal{P}_i$, $i = 1, \ldots, m$. It can be verified that the invariance conditions are solvable on each polytope $\mathcal{P}_i$. Thus, by Theorem 4.2 in Helwa and Broucke (2013), the RCP is solvable on each $\mathcal{P}_i$. What remains is to find a piecewise affine controller on each $\mathcal{P}_i$. This is done using Algorithm 4.11 of Habets and van Schuppen (2004).

3.5 Control Design

Algorithm 4.11 of Habets and van Schuppen (2004) requires that we first triangulate each polytope $\mathcal{P}_i$ into simplices. This can be performed in multiple ways. We used Delaunay triangulation with the addition of a centroid node (see, e.g., the work by Cheng et al. (2013) for more on Delaunay triangulation). After splitting polytope $\mathcal{P}_i$ into simplices $\mathcal{S}_1, \ldots, \mathcal{S}_s$, a control vector $u_v \in U$ such that $Av + Bu_v + a \in \mathcal{C}(v)$ is chosen at every point v in the union of vertex sets of $\mathcal{S}_1, \ldots, \mathcal{S}_s$. An affine feedback control $u_j : \mathcal{S}_j \to U$ is then defined by $u_j(v) = u_v$ for every vertex $v \subset \mathcal{S}_j$, and this is affinely extended to the entire $\mathcal{S}_j$. This defines a continuous piecewise affine control $u : \mathcal{P}_i \to U$ by defining the restriction of u on $\mathcal{S}_j$ as $u|_{\mathcal{S}_j} \equiv u_j$. Finally, we can use controls u defined on $\mathcal{P}_1, \ldots, \mathcal{P}_m$ to define $u : \cup_{i=1}^m \mathcal{P}_m \to U$. Note there is no guarantee that u is continuous on the boundary between $\mathcal{P}_i$ and $\mathcal{P}_{i+1}$, $i \in \{1, \ldots, m-1\}$. However, in our assignment of control values, u only has a discontinuity between $\mathcal{P}_4$ and $\mathcal{P}_5$.

As listing all 96 simplices and 48 assigned control values would take up a considerable amount of space, we omit stating the control values and affine controllers at each simplex. All relevant values are available in a supplementary document (Ornik et al., 2017). This completes our design of a controller for the parallel parking maneuver.

4. SIMULATION RESULTS

The procedure in Section 3 resulted in a hybrid affine system

$$\dot{x} = A_i x + B_i u(x) + a_i, \quad x \in \mathcal{P}_i, \qquad (4)$$

defined on $\mathcal{M}' = \mathcal{P}_1 \cup \mathcal{P}_2 \ldots \cup \mathcal{P}_8$. The system data (A_i, B_i, a_i), $i \in \{1, \ldots, 8\}$ is obtained from the linearization (3) of the system (1) around points

$$\boldsymbol{x}_i^L = (x_0, y_0, \min\{\theta^i, \theta^{i-1}\} - \varepsilon, (v_F + v_S)/2, 0).$$

This choice of linearization was justified in Section 3.3. Control $u : \mathcal{M} \to U$ is given by defining $u|_{\mathcal{P}_i}$ on each polytope as in Section 3.5.

Because of our choice of widening parameters w_{iy}, the state space $\mathcal{M}' = \mathcal{P}_1 \cup \mathcal{P}_2 \ldots \cup \mathcal{P}_m$ becomes very large in y-coordinate by the time the trajectories reach $\mathcal{P}_m$. (See Figure 4 for a drawing.) The polytopes are placed far from the car parked in front of the parking space to ensure that there is no collision, but there is currently no theory available to guarantee that our vehicle will not hit the curb or end up too far from it. The RCP theory merely guarantees that, starting from *any* state in $\mathcal{M}'$, the system state will exit through $\mathcal{F}_m^{\text{out}}$. However, $\mathcal{F}_m^{\text{out}} \not\subset \mathcal{E} \times \{0\}$. Thus, there is no theoretical proof that Problem 1 is solved by our algorithm. On the other hand, we note that the specifications of our problem do not use reach control theory to the full extent, as the initial position and orientation of the car are actually guaranteed to be inside $\mathcal{I} \times \{0\} = \mathcal{F}_1^{\text{in}}$, which is just one facet of one polytope in $\mathcal{M}'$. Thus, the set of possible system states of the car at the point of leaving $\mathcal{P}_1$ is a proper subset of $\mathcal{F}_1^{\text{out}} = \mathcal{F}_2^{\text{in}}$. Simulations show that it is in fact much smaller than $\mathcal{F}_1^{\text{out}}$. As we then proceed further through subsequent polytopes, this brings us to the conclusion that the final states of the car make up only a small subset of $\mathcal{F}_m^{\text{out}}$. Hence, we achieve much better behaviour than guaranteed by reach control theory.

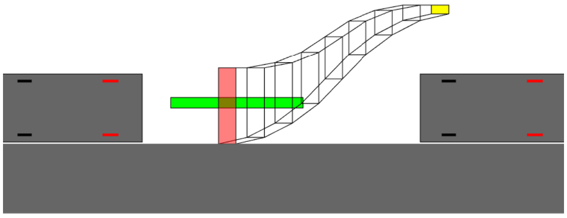

Fig. 4. A projection of state space $\mathcal{M}'$ designed from parameters in Section 3 on the x, y-space. Polytopes $\mathcal{P}_i \subset \mathcal{M}'$ are shown in black. The initial box $\mathcal{I}$ is marked in yellow, the ending box $\mathcal{E}$ in green, while the guaranteed final facet $\mathcal{F}_8^{\text{out}}$ is marked with red.

We now present the results of our simulations. We simulated the behaviour of 10 cars the nonlinear model (1). The initial states of the cars were chosen at random in $\mathcal{I} \times \{0\}$. However, to ensure that the system does not leave $\mathcal{M}'$ at the very beginning of the maneuver, we restricted the initial positions to be inside a 0.3m × 0.15m box in the middle of $\mathcal{I}$. The controller $u : \mathcal{M}' \to U$ remains the same as in the hybrid affine case. Figure 5 shows the results.

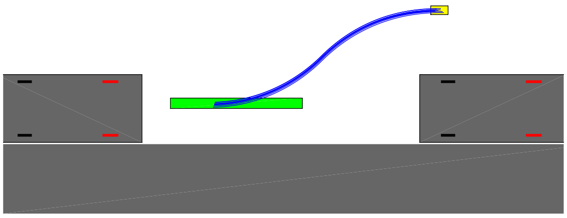

Fig. 5. The positions of cars during the maneuver, under the nonlinear model (1). The desired ending box $\mathcal{E}$ is given in green.

As reach control theory only pertains to the hybrid system (4), in the nonlinear case we had no guarantees at all for the behaviour of system state trajectories. However, Figure 5 shows that the linearization performed in Section 3 worked surprisingly well. The differences between the two models could be made smaller by working with a larger number of polytopes m, at the expense of computational power when calculating the feedback controller.

We additionally note that in the affine case the only guarantee provided by our control strategy was that all the cars will stop inside $\mathcal{F}_8^{\text{out}}$. However, all the cars, both using (4) and (1), in fact stopped in close proximity to each other. In particular, all the x-coordinates of the stopped cars at the end of the maneuvers are almost exactly the same. This was not guaranteed by our construction of the final exit facet $\mathcal{F}_8^{\text{out}}$.

To illustrate the similarity between the trajectories obtained by the two models, in Figure 6 we compare the trajectory of a car starting from the middle of the box $\mathcal{I} \times \{0\}$ and using the nonlinear model 1 with the trajectory of a car starting from the same initial condition, but obtained using the hybrid model (4). We additionally compare this to the intuitive, but not necessarily feasible, trajectory from Figure 2, which served as a motivation for our choice of polytopes $\mathcal{P}_i$. The trajectories are clearly similar.

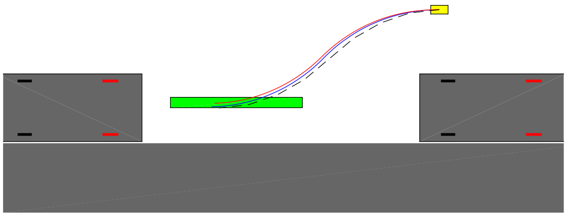

Fig. 6. A comparison between two trajectories of a car parking in the desired spot using the hybrid model (4), with the midpoint of the rear axle drawn in blue, and the nonlinear model (1), with the midpoint of the rear axle drawn in red. The trajectory given by the dashed black line is the imaginary trajectory from Figure 2.

5. CONCLUSION

The contribution of this paper is to give a robust control strategy for parallel parking which does not require the car to follow a specific previously calculated path. Our solution is based on a novel application of reach control theory. Extensive simulations show that the constructed feedback control results in the vehicle parking correctly in a desired position. We now list all assumptions that were made to obtain the presented solution.

- Nonlinear model (1) was linearized around non-equilibrium points to obtain a hybrid model (4). Additionally, the linearization used on each polytope $\mathcal{P}_i$ was obtained by linearizing around a point that is outside of $\mathcal{P}_i$.
- State space $\mathcal{M}'$ was chosen heuristically, based on an imaginary ideal trajectory, and without a guarantee of a correct parking maneuver.
- Controllers $u : \mathcal{P}_i \to U$ were not calculated to ensure the correctness of a parking maneuver. They were not even chosen by hand to match our expectations. The only criterion in place was that u solved the invariance conditions on $\mathcal{P}_i$. During simulations, we also required u to be continuous on $\mathcal{M}'$ except between $\mathcal{P}_4$ and $\mathcal{P}_5$. A controller automatically chosen by MATLAB that satisfied those criteria was used.

The fact that our model went through this many crude assumptions and still produced a correct parking maneuver is surprising and indicative of a deeper theory behind this problem. This paper is the first step towards exploring this theory. In particular, the fact that we let MATLAB choose any controller that satisfied the invariance conditions and the maneuver still worked in a desired way suggests a yet unexplored stability property for the RCP. Moreover, in the linearized model, the procedure ended up aligning all the cars to stop at virtually the same x-coordinate, even though their starting x-coordinates were up to 0.5m apart. We will investigate this apparent stability property in a future paper.

The construction of an appropriate state space $\mathcal{M}'$ is another limitation of our current approach. While the $\mathcal{M}'$ that we chose is guaranteed to be invariant under piecewise affine feedback, it is too large to theoretically ensure that the vehicle will stop in a correct position. A potential solution would be dynamic generation of the set $\mathcal{M}' = \mathcal{P}_1 \cup \ldots \cup \mathcal{P}_m$: when the car reaches a point at an exit facet of polytope $\mathcal{P}_i$, a smaller $\mathcal{P}_{i+1}$ could be constructed on the fly using the information on the current position of the car. This is another future research topic.

REFERENCES

Audi UK (2013). The Audi TT RS pricing and specification guide, edition 3.0 10/13, 2014 model year.

Belta, C. and Habets, L.C.G.J.M. (2006). Controlling a class of nonlinear systems on rectangles. *IEEE Transactions on Automatic Control*, 51(11), 1749–1759.

Chand, A., Kawanishi, M., and Narikiyo, T. (2015). Application of sigmoidal Gompertz curves in reverse parallel parking for autonomous vehicles. *International Journal of Advanced Robotic Systems*, 12.

Cheng, S.W., Dey, T.K., and Shewchuk, J.R. (2013). *Delaunay Mesh Generation*. CRC Press.

De Luca, A., Oriolo, G., and Samson, C. (1998). Feedback control of a nonholonomic car-like robot. In J.P. Laumond (ed.), *Robot Motion Planning and Control*, 171–253. Springer.

Demirli, K. and Khoshnejad, M. (2009). Autonomous parallel parking of a car-like mobile robot by a neuro-fuzzy sensor-based controller. *Fuzzy Sets and Systems*, 160(19), 2876–2891.

Girard, A. and Martin, S. (2012). Synthesis for constrained nonlinear systems using hybridization and robust controllers on simplices. *IEEE Transactions on Automatic Control*, 57(4), 1046–1051.

Habets, L.C.G.J.M., Collins, P.J., and van Schuppen, J.H. (2006). Reachability and control synthesis for piecewise-affine hybrid systems on simplices. *IEEE Transactions on Automatic Control*, 51(6), 938–948.

Habets, L.C.G.J.M. and van Schuppen, J.H. (2004). A control problem for affine dynamical systems on a full-dimensional polytope. *Automatica*, 40(1), 21–35.

Helwa, M.K. and Broucke, M.E. (2013). Monotonic reach control on polytopes. *IEEE Transactions on Automatic Control*, 58(10), 2704–2709.

Kloetzer, M. and Belta, C. (2008). A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control*, 53(1), 287–297.

Liang, Z., Zheng, G., and Li, J. (2012). Automatic parking path optimization based on Bezier curve fitting. In *2012 IEEE International Conference on Automation and Logistics*, 583–587.

Lyon, D. (1992). Parallel parking with curvature and nonholonomic constraints. In *Intelligent Vehicles '92 Symposium*, 341–346.

Martino, M. and Broucke, M.E. (2014). A reach control approach to bumpless transfer of robotic manipulators. In *53rd IEEE Conference on Decision and Control*, 25–30.

Murray, R.M. and Sastry, S.S. (1993). Nonholonomic motion planning: steering using sinusoids. *IEEE Transactions on Automatic Control*, 38(5), 700–716.

Ornik, M., Moarref, M., and Broucke, M.E. (2017). Numerical values used for simulations in "An automated parallel parking strategy using reach control theory". URL `http://bit.ly/2ntJmd9`.

Pohl, J., Sethsson, M., Degerman, P., and Larsson, J. (2009). A semi-automated parallel parking system for passenger cars. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 220(1), 53–65.

Roszak, B. and Broucke, M.E. (2006). Necessary and sufficient conditions for reachability on a simplex. *Automatica*, 42(11), 1913–1918.

Tilbury, D., Sørdalen, O.J., Bushnell, L., and Sastry, S.S. (1995). A multisteering trailer system: conversion into chained form using dynamic feedback. *IEEE Transactions on Robotics and Automation*, 11(6), 807–818.

Vukosavljev, M. and Broucke, M.E. (2014). Control of a gantry crane: a reach control approach. In *53rd IEEE Conference on Decision and Control*, 3609–3614.

Vukosavljev, M., Jansen, I., Broucke, M.E., and Schoellig, A.P. (2016). Safe and robust robot maneuvers based on reach control. In *2016 IEEE International Conference on Robotics and Automation*, 5677–5682.