Proceedings of the ASME 2018
Dynamic Systems and Control Conference
DSCC2018
September 30-October 3, 2018, Atlanta, Georgia, USA

DSCC2018-9195

PATH PLANNING FOR AUTONOMOUS CAR PARKING

Letian Lin
School of Electrical Engineering
and Computer Science
Ohio University
Athens, Ohio 45701
Email: ll728811@ohio.edu

J.Jim Zhu[*]
School of Electrical Engineering
and Computer Science
Ohio University
Athens, Ohio 45701
Email: zhuj@ohio.edu

ABSTRACT

The path planning problem for autonomous car parking has been widely studied. However, it is challenging to design a path planner that can cope with parking in tight environment for all common parking scenarios. The important practical concerns in design, including low computational costs and little human's knowledge and intervention, make the problem even more difficult. In this work, a path planner is developed using a novel four-phase algorithm. By using some switching control laws to drive two virtual cars to a target line, a forward path and a reverse path are obtained. Then the two paths are connected along the target line. As illustrated by the simulation results, the proposed path planning algorithm is fast, highly autonomous, sufficiently general and can be used in tight environment.

INTRODUCTION

The autonomous vehicle control system is composed of a mission trajectory planning system, a navigation system, and a trajectory tracking controller. Based on the mission objectives and the situational awareness data, the mission trajectory planner is responsible for generating a feasible, collision-free nominal trajectory that leads the vehicle to the target. The navigation system seeks the target (fixed or moving) and detects the environment. It also provides the real-time measurements of the vehicle states that are necessary in autonomous vehicle control. The trajectory tracking controller drives the vehicle to follow the nominal trajectory as well as accounts for tracking errors, modeling errors and disturbances. The block diagram of the overall autonomous vehicle control system is shown in Fig. 1. The mission trajectory planning system consists of a path planner and a path-trajectory converter, as shown in Fig. 2. Based on the mission objectives and by using the situational awareness data provided by the navigation system, the path planner can generate a feasible, collision-free path that connects the starting point to the destination. The path-trajectory converter is used to assign an appropriate velocity profile along the path and thus convert it to a nominal trajectory. Path planning is particularly significant and challenging in solving the autonomous car parking problem. On one hand, the motion of the vehicle at low speed is subject to the nonholonomic constraints which make maneuvering very complex. On the other hand, nowadays, as the parking spots have become narrow in cities, parking might be stressful even for experienced drivers [1].

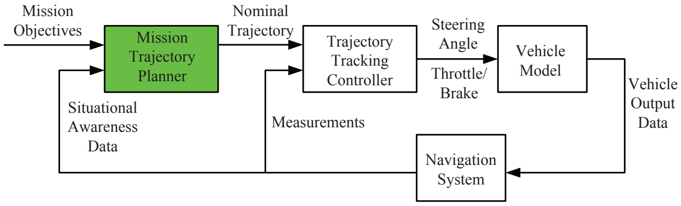

FIGURE 1. AUTONOMOUS VEHICLE CONTROL SYSTEM

According to the strategies implemented, the existing path planning techniques for autonomous car parking can be classified

[*]Address all correspondence to this author.

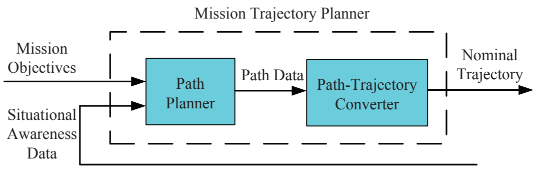
Mission Trajectory Planner
Mission Objectives
Situational Awareness Data
Path Planner
Path Data
Path-Trajectory Converter
Nominal Trajectory

FIGURE 2. MISSION TRAJECTORY PLANNING SYSTEM

in four main groups: geometric approach, three-step approach, sampling-based approach and numerical optimization approach. The geometric approach considers the path planning in car parking as a Euclidean geometry problem, that is, constructing a path by using a set of curves with basic geometrical equations that satisfy the maximum curvature constraint. Many studies of the geometric approach consider constructing a path with straight line segments and circle arcs. This type of paths was implemented in [1–3] for parallel parking and in [4] for perpendicular parking. In [5, 6], the backward motion of parallel parking is selected from a class of time-parametrized curves that are generated by applying the sinusoidal control inputs to the car. The method was extended to the head-in perpendicular parking in [7] and to back-in perpendicular parking in [8]. In [9–11], in order to reduce the dependence on human's knowledge in design, a fuzzy logic was introduced to the geometric approach to adjust the parameters of the predefined curve patterns.

In [12, 13], a three-step path planning approach for nonholonomic systems was proposed and its application in parallel parking was presented. The algorithm begins with computing a collision-free path without satisfying the nonholonomic constraints. Then, by taking into account the nonholonomic constraints and the other kinematics constraints, the path obtained in step one is approximated by a connection of segments computed by a local planner. In the last step, the path generated in step two is optimized into a shorter and smoother one with fewer cusps.

The sampling-based approach employs the sampling-based pathfinding algorithms which has been extensively studied in robotics and computer science. Instead of explicitly constructing the free space, which is time-consuming to compute [14], these algorithms probe the free space and conduct a search with a sampling strategy. The algorithms stop when a path connecting the initial and final poses is found. According to the sampling type, the sampling-based pathfinding algorithms can be classified into two categories: random-sampling-based algorithms and orderly-sampling-based algorithms. The most popular random-sampling-based algorithms is the Rapidly-exploring Random Tree (RRT) [15, 16]. RRT was applied to path planning in labyrinth environment for car-like robots in [16]. Its applications in perpendicular parking and parallel parking were shown in [17] and [18], respectively. The most notable orderly-sampling-based algorithm is A* [19] and its extensions. In [20], a path planner based on the Hybrid-State A* was used in perpendicular parking for a full-size car. Another orderly-sampling-based algorithm was developed in [21] where the possible traces of a car backing out from the parking space are sampled in order. In [22], a neural dynamics based algorithm which is essentially an orderly-sampling-based one was applied to parallel parking.

In the numerical optimization approach, by introducing an optimality objective, the motion planning problem for a nonholonomic system is reformulated as an open-loop optimal control problem and solved numerically. In [23], the minimum time paths were found for all three parking cases where the optimization problem was solved by using sequential quadratic programming (SQP). A similar study was carried out in [24] where the interior-point method was used to solve the optimization problem. In [25, 26], the path planning problem was discretized and the path is determined by solving a static optimization problem iteratively.

The common criteria to evaluate a path planner include computing time, smoothness of path, number of cusps on path, tight parking capability, applicability to different scenarios and dependence on human's knowledge and intervention. Comparison of the four path planning approaches in terms of these criteria is shown in Tab. 1. It is noted that the assessment of each approach is qualitative rather than quantitative. Due to the lack of common computing and vehicle platform for performance evaluation, this comparison are inevitably inaccurate and somewhat subjective. The geometric approach is widely adopted in the current commercial automatic parking systems. Its main advantage is that, since the path is solved geometrically, the computation is fast. Also, the implementation of the algorithm is quite simple. In contrast, the other approaches are time-consuming from the practical perspective. In [26], the computing time of the optimization based path planner is reported to be around 20ms. However, the actual computing time can be much larger if the landmarks and the weighting factors in the cost function are selected improperly. Although the geometric approach is fast and simple, its performance is far from satisfactory because of some drawbacks. First, a geometric planner can only be applied to a specific parking scenario and may fail for small environmental changes. Second, the design involves a great deal of human's knowledge. Third, the driver must drive the car to a pre-specified zone to enable planning. The three-step approach often results in a path with too many cusps [27]. The sampling-based approach has a very low efficiency in tight parking and the resulted path is often jerky. In order to produce a path in reasonable time, the numerical optimization approach requires a good initial guess which may cost over 40s to compute on typical current CPUs [24]. However, the initialization must be carried out for each new parking environment that is not included in the lookup table. Therefore, it is a challenging problem to design a fast path planner that has a good performance according to these criteria.

In this paper, we propose a path planning algorithm for autonomous car parking with the following novelties:

TABLE 1. COMPARISON OF PATH PLANNING APPROACHES FOR AUTOMATIC CAR PARKING.

Approaches	Computing time	Smoothness	Number of cusps	Applicability to scenarios	Tight parking	Tight drive aisle	Human's intervention
Geometric	Very short	Good	Medium	Low	Fair	Bad	High
Three-step	Long	Good	Large	High	Good	Good	Low
Sampling-based (Random)	Medium	Bad	Medium	High	Bad	Bad	Low
Sampling-based (Orderly)	Medium	Fair	Small	High	Fair	Fair	Low
Numerical optimization	Very long (Including initialization)	Good	Small	High	Good	Good	Low

(1) The path is constructed by using a new four-phase algorithm. In phase 1 and 2, a reverse path is obtained by retrieving a virtual parked-car from the parking space and steering it to a target line in the drive aisle. In phase 3, a forward path is generated by steering a virtual car from the initial pose to the target line. In phase 4, the two paths are connected along the target line.

(2) Instead of finding a path with sufficient width to accommodate the car or computing the free-space as in [12], we pad the obstacles with a buffer zone for collision avoidance, thereby reducing the feasible path to a single line, which greatly improved the computational efficiency.

(3) A feasible path is constructed using the line-of-sight (LOS) pure-pursuit guidance technique, which produces a feasible path segments that are natural and smooth.

(4) The problem of retrieving a virtual parked-car from the parking space and the problem of steering a virtual car to a target line are formulated as a 1st-order and a 3rd-order nonlinear switching control problem respectively. For each problem, an effective switching control law is designed. To the best of our knowledge, this is the first time that the switching control theory is used to solve the path planning problem.

1 Switched system background

Before we present the main results, the following definitions and theorem for switched systems should be introduced.

Definition 1 (Switched system). *[28] [29] Given a family of continuous vector fields $\left\{f_p : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n, p \in \mathscr{P}\right\}$, where $\mathscr{P} = \{1, 2, \cdots, N\}$ is a set of positive integers, a continuous-time switched system is modeled as $\dot{\xi} = f_\sigma(\xi, \mu)$, where $\sigma : [0, +\infty) \to \mathscr{P}$, called switching signal, is a piecewise continuous integer-valued function of time that specifies the vertor field being followed at each time instant.*

Definition 2 (Dwell time). *[30] A switching signal $\sigma(t)$ is said to be with a dwell time $\tau_d > 0$ if the switching times t_1, t_2, ... satisfy the inequality $\inf_{k\geq 1}(t_{k+1} - t_k) \geq \tau_d$.*

In other words, if a switching signal has a dwell time τ_d, then the time interval between any two consecutive switchings is not smaller than τ_d. The following definitions are made for the unforced switched system

$$\dot{\xi} = f_\sigma(\xi),\ \sigma : [0, +\infty) \to \mathscr{P} \qquad (1)$$

Definition 3 (Weak invariance). *[31] A compact set $M \subset \mathbb{R}^n$ is weakly invariant with respect to (1) if for each $\xi \in M$ there exist an index $p \in \mathscr{P}$, a solution $\phi(t)$ of the vector field $f_p(\xi)$ and $T > 0$ such that $\phi(0) = \xi$ and $\phi(t) \in M$ for either $t \in [-T, 0]$ or $t \in [0, T]$.*

Definition 4 (Common weak Lyapunov function). *[31] Let Ω be an open subset of $\mathbb{R}^n$ containing the origin. A continuously differentiable function $V(\xi) : \Omega \to [0, +\infty)$ is said to be a common weak Lyapunov function for (1) if it is positive definite and satisfies $(\partial V/\partial\xi)^T f_p(\xi) \leq 0$ for each $\xi \in \Omega$, $p \in \mathscr{P}$.*

Definition 5 (Attractivity). *[31] A solution $\phi(t)$ of (1) is said to be attracted by a compact set $M \subset \mathbb{R}^n$ if for each $\varepsilon > 0$ there exists $T > 0$ such that $Dist(\phi(t), M) < \varepsilon$, $\forall t \geq T$, where $Dist(\cdot,\cdot)$ is the point-to-set distance function on $\mathbb{R}^n$ defined by $Dist(x,\ E) = \inf_{y\in E} ||x - y||$, for $x \in \mathbb{R}^n$, $\emptyset \neq E \subset \mathbb{R}^n$.*

By using these definitions, the extension of LaSalle's Invariance Principle to switched systems is stated as following.

Theorem 1.1. *[31] Let $V(\xi) : \Omega \to [0, +\infty)$ be a common weak Lyapunov function for (1). Let $c > 0$ and let Ω_c be*

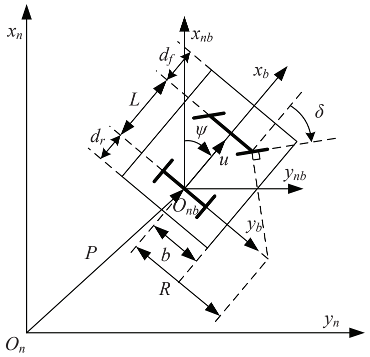

FIGURE 3. COORDINATE FRAME.

the connected component of the level set $\{\xi \in \Omega : V(\xi) < c\}$ such that $0 \in \Omega_c$. Assume that Ω_c is bounded and let $Z = \{\xi \in \Omega : \exists p \in \mathscr{P}$ such that $(\partial V/\partial \xi)^T f_p(\xi) = 0\}$. Let M be the union of all the compact, weakly invariant sets which are contained in $Z \cap \Omega_c$. If the switching signal has a dwell time, then every solution $\phi(t)$ to (1) with $\phi(0) \in \Omega_c$ is attracted by M.

2 Problem Formulation

2.1 Kinematic model and parking environment

We employ the following coordinate frames to describe the kinematics of the car: a ground-fixed North-East-Down (NED) navigation frame (n-frame), a body-carried NED frame (nb-frame) and a body-fixed frame (b-frame). The n-frame has its origin O_n located at a fixed point of interest on the ground, with x_n pointing to the north, y_n pointing to the east, and z_n pointing down. We treat the n-frame as a flat and inertial frame. The nb-frame which is parallel to the n-frame has its origin O_{nb} located at the midpoint of the rear axle of the car. The b-frame has its origin O_b overlapped with O_{nb}, with x_b along the longitude of the car pointing forward, y_b pointing to the driver's right side of the car and z_b pointing down. The reference frames and notations are shown in Fig. 3, where L is the wheelbase, d_f is the front overhang, d_r is the rear overhang, b is the half-width of the car, ψ is the orientation of the car in the nb-frame with respect to the x_{nb}-axis, δ is the steering angle relative to the x_b-axis, u is the longitudinal speed in the b-frame, R is the turning radius.

Consider the following kinematics equation of a car

$$\begin{aligned} \dot{x}_n(t) &= \cos\psi(t)\cdot u(t) \\ \dot{y}_n(t) &= \sin\psi(t)\cdot u(t) \\ \dot{\psi}(t) &= \kappa(t)\cdot u(t) = \frac{1}{L}\tan\delta(t)\cdot u(t) \end{aligned} \tag{2}$$

where κ can be interpreted geometrically as the curvature of the car's path in x_n-y_n plane. The car is assumed to drive at low speed on level ground while parking, thus it can be assumed that there is no skidding between the tires and the ground. It is also assumed that δ is bounded by $|\delta(t)| \leq \delta_{\max}$. This constraint can be characterized by using saturated κ. We thus rewrite the last equation in (2) as $\dot{\psi}(t) = \text{sat}_{\kappa^*}(\kappa(t))\cdot u(t)$, where sat$(\cdot)$ is the saturation function defined by

$$\text{sat}_a(x) = \begin{cases} -a & \text{for } x < -a \\ x & \text{for } |x| \leq a \\ a & \text{for } x > a \end{cases}, \text{ for all } a > 0 \tag{3}$$

and κ^* is the saturation limit that satisfies $\kappa^* = (\tan\delta_{\max})/L$. Let s be the traveled distance of the car from a given initial position. Since a path is a curve in the x_n-y_n plane, it is independent of u. Thus, without loss of generality, we can normalize u as U satisfying $|U| = 1$. The normalization can be carried out by a change of time variable $t \to s$. Then, the path is parameterized as $(x_n(s), y_n(s), \psi(s))$ and U can be interpreted as the traveling direction of the car. In the rest of this paper, we consider the kinematics equation represented in the normalized time s

$$\begin{aligned} \dot{x}_n(s) &= \cos\psi(s)\cdot U \\ \dot{y}_n(s) &= \sin\psi(s)\cdot U \\ \dot{\psi}(s) &= \text{sat}_{\kappa^*}(\kappa(s))\cdot U \end{aligned} \tag{4}$$

Notice that all the derivatives in (4) are with respect to s. Without any ambiguity, the parameter s will be omitted throughout the rest of this paper. Since the normalized time s is also the traveled distance, the words "path" and "trajectory" will be used interchangeably and s will be simply called "time".

The parking space is commonly structured in a quadrangular form. There are three types of parking structures: "lane", "diagonal" and "row" [6]. The lane structure is usually used in parking along the street where the parallel parking maneuver is required. The diagonal and row structures are typical for squares or garages. For parking in such places, angle parking or perpendicular parking maneuver is involved.

Without loss of generality, we assume that (i) the parking space is on the right side of the driver, (ii) the origin O_n of the n-frame is located at the final position of the car in the parking space and (iii) the drive aisle is parallel to the y_n axis as shown in Fig. 4. For the general cases where the assumptions (i), (ii) and (iii) are violated, the question can be put in the same form by an inertial frame transform. It is also assumed that in x_n and y_n directions there are restrictions that the car body can not cross.

2.2 Path planning problem

The goal is to develop a unified path planning algorithm which, for a given car, a parking space and restrictions on x_n and

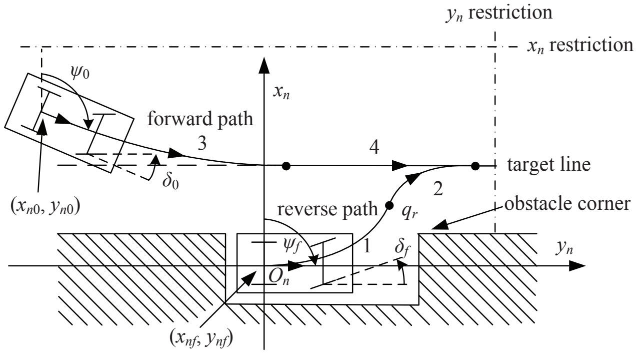

FIGURE 4. FOUR PHASES PATH PLANNING SCHEME.

y_n, generates a feasible, collision-free path that connects the initial pose $q_0 = (x_{n0}, y_{n0}, \psi_0)$ and the final pose $q_f = (x_{nf}, y_{nf}, \psi_f)$. By the assumptions previously made, the final pose in our research is taken by $q_f = (0, 0, \pi/2)$.

3 Path planning algorithm for car parking

3.1 Four-phase algorithm

In this research, the path is constructed by using a four-phase algorithm. Without loss of generality, the idea is illustrated by the parallel parking scenario in Fig. 4. In Phase 1, a virtual parked-car is retrieved from the parking space to a ready position q_r where the obstacle corner can be avoided. In Phase 2, the virtual parked-car is steered to a target line parallel to the drive aisle. A reverse path is obtained by recording the trace of the virtual parked-car in Phase 1 and 2. Note that this path can be used later for retrieval of the car from the parking space provided that the obstacle corner remains unchanged. In Phase 3, a forward path is generated by steering a virtual car from the initial pose to the target line. In Phase 4, the two paths are connected along the target line.

3.2 Steering the virtual car to the target line

In this subsection, the problem of steering a virtual car to a target line is formulated and solved under the switching control framework.

3.2.1 Problem formulation Without loss of generality, consider a target line defined in the n-frame by $l_{tgt} = \left\{(x_n, y_n) \in \mathbb{R}^2 : x_n = x_{tgt}\right\}$, where x_{tgt} is a real constant. In the general cases, a target line that is not parallel to the y_n-axis can be put in the above form by a coordinate rotation. For a virtual car of half-width b driving on a drive aisle of width W (normally $W > 2.5b$), in order to guarantee that the target line is achievable with a small number of directional changes and with a reasonable traveled distance, a rule of thumb for the selection of x_{tgt} is $2.3b < x_{tgt} < \min\{W, 3.5b\}$. It is desirable to change the steering angle of the virtual car continuously while driving. Otherwise, when the trace of the virtual car is used as the path for a real car, the car may have to stop at the positions where sudden steering angle change occurs . A straightforward way to consider the continuity of the steering angle is to augment the kinematics equation (4) by introducing $\dot{\kappa} = \rho$, where ρ replaces κ as the control input to (4). It can be seen that, if ρ is piecewise continuous, κ would be continuous. Then the continuity of δ would be guaranteed. The augmented kinematics equation is written as

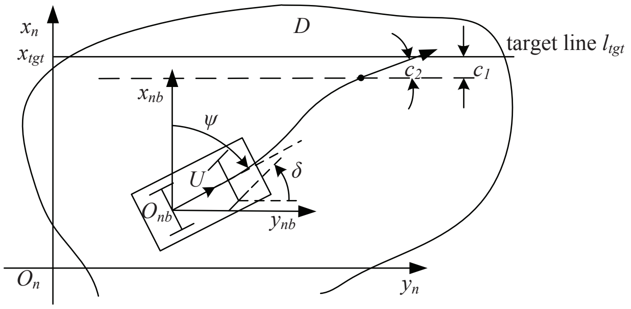

FIGURE 5. STEERING A CAR TO THE TARGET LINE.

$$\begin{aligned} \dot{x}_n &= \cos\psi \cdot U \\ \dot{y}_n &= \sin\psi \cdot U \\ \dot{\psi} &= \mathrm{sat}_{\kappa^*}(\kappa) \cdot U \\ \dot{\kappa} &= \rho \end{aligned} \tag{5}$$

A bounded region $D \subset \mathbb{R}^2$ is defined on the x_n-y_n plane due to obstacles and restrictions on x_n axis and y_n axis. The virtual car's body must be inside D while driving. In order to guarantee that the virtual car remains in D, the driving direction U must be changed when the virtual car reaches the boundary of D. Thus, (5) can be treated as a switched system with U as the switching signal whose sign changes when the boundary of D is reached. Notice that the corresponding switching logic dependents on the virtual car's pose and the boundary of D.

In Phase 2 and 3, we want to steer a virtual car to the vicinity of the target line. It is not necessary to align the virtual car to the target line exactly. Let $c_1 > 0$ be a tolerance of the final distance to the target line in x_n coordinate. Let $c_2 > 0$ be a tolerance of the final angle between the car's orientation and the positive direction of y_n-axis. Let $I_{y_n} = \{y_n \in \mathbb{R} : y_{\min} \le y_n \le y_{\max}\}$, where $y_{\min}$ and $y_{\max}$ are constants, be a finite interval on y_n-axis that contains all the possible values of y_n when the virtual car is inside D. Our goal is designing a switched control law to drive a virtual car characterized by (5) from an initial pose (x_{nA}, y_{nA}, ψ_A) to a final pose (x_{nB}, y_{nB}, ψ_B) for which $|x_{nB} - x_{tgt}| < c_1$, $y_{nB} \in I_{y_n}$ and $|\psi_B - \pi/2| < c_2$. The control problem is shown in Fig. 5. The nominal trajectories of (5) correspond to the motions of the vir-

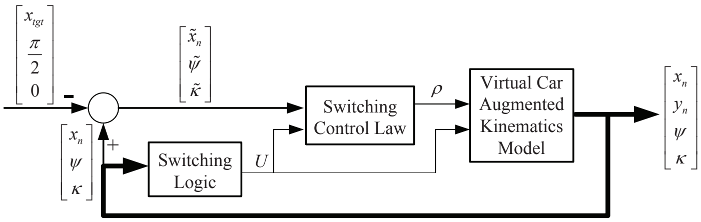

FIGURE 6. CONTROL DIAGRAM FOR STEERING THE VIRTUAL CAR TO THE TARGET LINE.

tual car traveling inside D along the target line l_{tgt} with car heading to the positive direction of y_n. They are given by $\bar{x}_n = x_{tgt}$, $\dot{\bar{y}}_n = \bar{U}$, $\bar{\psi} = \pi/2$, $\bar{\kappa} = 0$. The nominal control input is given by $\bar{\rho} = 0$.

In order to steer the virtual car to the target line, the switching control design is conducted in the error coordinates. Let $\tilde{x}_n = x_n - \bar{x}_n$, $\tilde{y}_n = y_n - \bar{y}_n$, $\tilde{\psi} = \psi - \bar{\psi}$, $\tilde{\kappa} = \kappa - \bar{\kappa}$, $\tilde{\rho} = \rho - \bar{\rho}$, the error dynamics are given by

$$\begin{aligned}\dot{\tilde{x}}_n &= -\sin\tilde{\psi}\cdot U\\ \dot{\tilde{y}}_n &= \cos\tilde{\psi}\cdot U - \bar{U}\\ \dot{\tilde{\psi}} &= \mathrm{sat}_{\kappa^*}(\tilde{\kappa})\cdot U\\ \dot{\tilde{\kappa}} &= \tilde{\rho}\end{aligned} \tag{6}$$

where U and $\bar{U}$ are the switching signals for (5) and the nominal trajectories respectively, $\tilde{\rho}$ is the control input. The aforementioned requirement on the final pose is translated as $|\tilde{x}_n(s_B)| < c_1$, $|\tilde{y}_n(s_B)| \leq |y_{\max} - y_{\min}|$, $|\tilde{\psi}(s_B)| < c_2$, for some finite time s_B. It should be noted that $\tilde{y}_n$ cannot be fully controlled by using $\tilde{\rho}$ because the sign of $\dot{\tilde{y}}_n$ is dominated by $-\bar{U}$. However, the constraint on $\tilde{y}_n(s_B)$ is guaranteed by the switching logic of U and $\bar{U}$. Since $\tilde{y}_n$ has no effect on the other three state variables, the problem is reduced to designing a switching control law for the following reduced order error dynamics

$$\begin{aligned}\dot{\tilde{x}}_n &= -\sin\tilde{\psi}\cdot U\\ \dot{\tilde{\psi}} &= \mathrm{sat}_{\kappa^*}(\tilde{\kappa})\cdot U\\ \dot{\tilde{\kappa}} &= \tilde{\rho}\end{aligned} \tag{7}$$

The switching control block diagram is shown in Fig. 6.

3.2.2 Switching control law and attractivity theorem

The following two assumptions are made for the switching signal U to facilitate the switching control design.

Assumption 1. *U in (7) is time-dependent.*

Assumption 2. *U has a user-specified dwell-time.*

Assumption 1 is made because the dependence of U on the boundary of D and the pose of the virtual car is difficult to characterize mathematically by using a switching logic rule. Therefore, we treat U as a time-dependent switching signal with uncertain switching times. Assumption 2 is feasible because the dwell-time for U is determined by the minimum drivable space and it can be provided by the user in practice. For (7), we design the switching control law $\tilde{\rho} = K_1\tilde{x}_n - K_2\sin\tilde{\psi}\cdot U - K_3\tilde{\kappa}$, where K_1, K_2, $K_3 > 0$, then the closed-loop reduced order error dynamics are give by

$$\begin{aligned}\dot{\tilde{x}}_n &= -\sin\tilde{\psi}\cdot U\\ \dot{\tilde{\psi}} &= \mathrm{sat}_{\kappa^*}(\tilde{\kappa})\cdot U\\ \dot{\tilde{\kappa}} &= K_1\tilde{x}_n - K_2\sin\tilde{\psi}\cdot U - K_3\tilde{\kappa}\end{aligned} \tag{8}$$

The following theorem guarantees that the trajectories of (8) are attracted by the origin of the error coordinate.

Theorem 3.1. *Consider the system (8). Let the constraint on $\tilde{x}_n$ be $|\tilde{x}_n| < x_{\max}$. If the switching signal U has a dwell-time, and K_1, K_2, K_3 satisfy $K_2K_3 > K_1$, $\kappa^* > (K_2K_3 - K_1)/(4K_3^2)$, $2(K_2K_3 - K_1)/(K_1K_2) > x_{\max}^2$, then the trajectories starting from $\Omega_0 = \{(\tilde{x}_n, \tilde{\psi}, \tilde{\kappa}) : |\tilde{x}_n| < x_{\max}, |\tilde{\psi}| \leq \pi/2, \tilde{\kappa} = 0\}$ are attracted by the origin.*

Proof. Let $H = K_2K_3 - K_1$. Consider the common weak Lyapunov function candidate $V(\tilde{x}_n, \tilde{\psi}, \tilde{\kappa}) = K_1H\tilde{x}_n^2 + 2K_3H(1 - \cos\tilde{\psi}) + (K_1\tilde{x}_n - K_3\tilde{\kappa})^2$. The derivative of V along the trajectories of (8) is given by

$$\dot{V} = -2K_1^2K_3\tilde{x}_n^2 - 2K_3|\tilde{\kappa}|\left(K_3^2|\tilde{\kappa}| - H\sin\tilde{\psi}\,(\tilde{\kappa} - \mathrm{sat}_{\kappa^*}(\tilde{\kappa}))/|\tilde{\kappa}|\right) \tag{9}$$

For $|\tilde{\kappa}| \leq \kappa^*$, $\mathrm{sat}_{\kappa^*}(\tilde{\kappa}) = \tilde{\kappa}$, then $\dot{V} = -2K_1^2K_3\tilde{x}_n^2 - 2K_3^3\tilde{\kappa}^2 \leq 0$. Now consider the case when $|\tilde{\kappa}| > \kappa^*$. By the condition $K_2K_3 > K_1$, we have $H > 0$. Thus, (9) yields $\dot{V} \leq -2K_1^2K_3\tilde{x}_n^2 - 2K_3|\tilde{\kappa}|\left(K_3^2|\tilde{\kappa}| - H(1 - \kappa^*/|\tilde{\kappa}|)\right)$. By the condition $\kappa^* > (K_2K_3 - K_1)/(4K_3^2)$, the term in the bracket satisfies $K_3^2|\tilde{\kappa}| - H(1 - \kappa^*/|\tilde{\kappa}|) \geq 2\sqrt{K_3^2H\kappa^*} - H > 0$. Therefore, we have $\dot{V} < 0$. Note that this holds for any neighborhood Ω_c of the origin. Let $\Omega_c = \{(\tilde{x}_n, \tilde{\psi}, \tilde{\kappa}) : V(\tilde{x}_n, \tilde{\psi}, \tilde{\kappa}) < 4K_3H\}$. Then, we have $|\tilde{\psi}| < \pi$ in Ω_c and the subset Z where $\dot{V} = 0$ is given by $Z = \{(\tilde{x}_n, \tilde{\psi}, \tilde{\kappa}) \in \Omega_c : \tilde{x}_n = 0, \tilde{\kappa} = 0\}$. It can be seen that the unique compact, weakly invariant set contained in Z is the origin. Moreover, by condition $2(K_2K_3 - K_1)/(K_1K_2) > x_{\max}^2$, we have $\Omega_0 \subset \Omega_c$. Therefore, by Theorem 1.1, the origin is attractive for all trajectories starting from Ω_0.

Consequently, for all $c_1 > 0$, $c_2 > 0$, every trajectory starting from Ω_0 enters and remains in the region $\Omega = \{(\tilde{x}_n, \tilde{\psi}, \tilde{\kappa}) : |\tilde{x}_n| < c_1, |\tilde{\psi}| < c_2\}$ in some finite time.

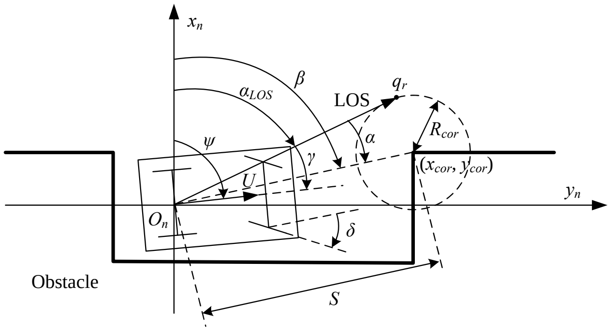

FIGURE 7. RETRIEVING A CAR FROM THE PARKING SPACE

3.3 Retrieving a virtual car from the parking space

3.3.1 Problem formulation Consider an available parking space. While retrieving a virtual parked-car, the corners of the parking space should be avoided. We apply the following steps: (i) around each corner, draw an avoidance circle with a radius $R_{cor} > b$, (ii) maneuver the virtual parked-car until the car's longitude pointing out the parking space without passing through the circles, (iii) drive the car straightly out of the parking space and stop at a ready position q_r where the corner is guaranteed to be avoid. Step (ii) can be conducted by selecting an obstacle corner and maneuver the car until the longitude is tangent to the avoidance circle, as shown in Fig. 7. A rule of thumb of choosing R_{cor} is $R_{cor} = 1.4b$.

Let the LOS be the tangent line that points out the parking space and passes through the midpoint of the rear axle. The LOS angle, denoted by γ, is defined by the acute angle between LOS and the longitude of the car. The goal can be stated as designing a switching control law under which, for a given tolerance $c > 0$, the LOS angle γ satisfies $|\gamma| < c$ in finite time. We first derive the differential equation for γ. By Fig. 7, $\alpha_{LOS} = \psi - \gamma = \beta - \alpha$, where $\beta = \arctan\left[(y_{cor} - y_n)/(x_{cor} - x_n)\right]$, $\alpha = \arcsin(R_{cor}/S)$, in which $S = \sqrt{(x_{cor} - x_n)^2 + (y_{cor} - y_n)^2}$. By the kinematic equation (4), We thus have $\dot{\gamma} = \dot{\psi} - \dot{\alpha}_{LOS} = \mathrm{sat}_{\kappa^*}(\kappa) \cdot U - \dot{\beta} + \dot{\alpha}$, where $\dot{\beta}$ along the trajectories of (4) is given by $\dot{\beta} = \sin(\alpha - \gamma) \cdot U/S$, and $\dot{\alpha}$ along the trajectories of (4) is given by $\dot{\alpha} = \tan\alpha \cos(\alpha - \gamma) \cdot U/S$. Substituting these expressions for $\dot{\beta}$ and $\dot{\alpha}$, we obtain $\dot{\gamma} = \mathrm{sat}_{\kappa^*}(\kappa) \cdot U + \left(1/\sqrt{S^2 - R_{cor}^2}\right) \sin\gamma \cdot U$. When maneuvering a car in a tight parking space, directional changes are unavoidable. However, a minimal number of directional changes is desired. This means that we must allow the steering angle to be discontinuous along the path. Thus, the switching control law can be directly designed for κ. By treating S as a time-varying parameter, written as $S(s)$, and by taking the switching control law as $\kappa = -K\gamma U$, the closed-loop system is given by

$$\dot{\gamma} = -\mathrm{sat}_{\kappa^*}(K\gamma) + \left(1/\sqrt{S^2(s) - R_{cor}^2}\right) \sin\gamma \cdot U \qquad (10)$$

3.3.2 Switching control law and stability theorem

The stability of (10) is presented by the following theorem.

Theorem 3.2. *For the system (10), if (i) there exists $d > 0$ such that $d < \sqrt{S^2(s) - R_{cor}^2}$ holds in a sufficiently large finite positive time interval, (ii) $\gamma(0)$ satisfies $|\sin\gamma(0)| < \kappa^* d$, (iii) the gain K is selected to satisfy $K > 1/d$, then, for a given tolerance $c > 0$, the trajectory of γ enters the region $|\gamma| < c$ in finite time.*

Proof. Consider the quadratic function $V = \gamma^2/2$. Now the problem is studied in two cases.

For $|\gamma| > \kappa^*/K$, we have $\mathrm{sat}_{\kappa^*}(K\gamma) = \kappa^* \mathrm{sgn}(\gamma)$, where $\mathrm{sgn}(\cdot)$ is the sign function. Then the derivative of V along the trajectories of (10) is given by $\dot{V} = \dot{\gamma}\gamma = -\kappa^*|\gamma| + \left(1/\sqrt{S^2(s) - R_{cor}^2}\right) \sin(\gamma)\gamma U$. Suppose that the condition (i) in the theorem holds, then we have $\dot{V} \leq -\kappa^*|\gamma| + \frac{1}{d}\sin(\gamma)\gamma = -\left(\kappa^* - |\sin\gamma|/d\right)|\gamma|$. Suppose that the condition (ii) also holds, then $\dot{V} \leq -\left(\kappa^* - |\sin(\gamma(0))|/d\right)|\gamma| < 0$ holds in the set $|\gamma| \leq |\gamma(0)|$. Therefore, the set $|\gamma| \leq |\gamma(0)|$ is invariant and $|\gamma| = \sqrt{2V}$ decrease monotonically with respect to time. Moreover, the above inequality for $\dot{V}$ can be rewritten as $\dot{V} \leq -\left(\kappa^* - |\sin(\gamma(0))|/d\right)\sqrt{2V} < 0$. Then, by using the comparison principle, it can be shown that $|\gamma| = \sqrt{2V} < |\gamma(0)| - \left(\kappa^* - |\sin(\gamma(0))|/d\right)s$. Therefore, the trajectory of γ starting from $|\gamma(0)| > \kappa^*/K$ will enter the region $|\gamma| \leq \kappa^*/K$ within time $T_1 = \dfrac{|\gamma(0)| - \kappa^*/K}{\kappa^* - |\sin(\gamma(0))|/d}$.

For $|\gamma| \leq \kappa^*/K$, we have $\mathrm{sat}_{\kappa^*}(K\gamma) = K\gamma$. Then the derivative of V along the trajectories of the system (10) is given by $\dot{V} = \dot{\gamma}\gamma = -K\gamma^2 + \left(1/\sqrt{S^2(s) - R_{cor}^2}\right) \sin(\gamma)\gamma U$. Suppose that condition (i) in the theorem holds, then we have $\dot{V} \leq -K\gamma^2 + \sin(\gamma)\gamma/d$. Moreover, by condition (iii), $\dot{V} \leq -\left(K - 1/d\right)\gamma^2 = -2\left(K - 1/d\right)V < 0$ holds for all $|\gamma| \leq \kappa^*/K$, $\gamma \neq 0$. Therefore, the set $|\gamma| \leq \kappa^*/K$ is invariant and $|\gamma| = \sqrt{2V}$ will decay to 0 in an exponential convergence rate. Consequently, for any $c > 0$, the trajectory of γ will enter the region $|\gamma| < c$ in finite time.

3.4 Forward and reverse paths connection

Due to the tolerance for the ending poses at the target line, the forward and reverse paths cannot be simply connected by a segment of the target line. Hence, we consider using a 5th-order polynomial curve to connect the two paths. Let $(x_{nBF}, y_{nBF}, \psi_{BF})$ and $(x_{nBR}, y_{nBR}, \psi_{BR})$ be the final poses for the forward and reverse paths respectively. Let δ_{BF} and δ_{BR} be the final steering angles corresponding to the forward and reverse paths respectively. Then, the two paths can be connected by using the 5th-order polynomial function defined on $\left[\min\{y_{nBF}, y_{nBR}\}, \max\{y_{nBF}, y_{nBR}\}\right]$ in the form of $x_n(y_n) = a_5 y_n^5 + a_4 y_n^4 + a_3 y_n^3 + a_2 y_n^2 + a_1 y_n + a_0$, where $\{a_i\}$ can be solved by the endpoint condition: $x_n(y_{nBF}) = x_{nBF}$, $\frac{dx_n}{dy_n}(y_{nBF}) = \cot\psi_{BF}$, $x_n(y_{nBR}) = x_{nBR}$, $\frac{dx_n}{dy_n}(y_{nBR}) =$

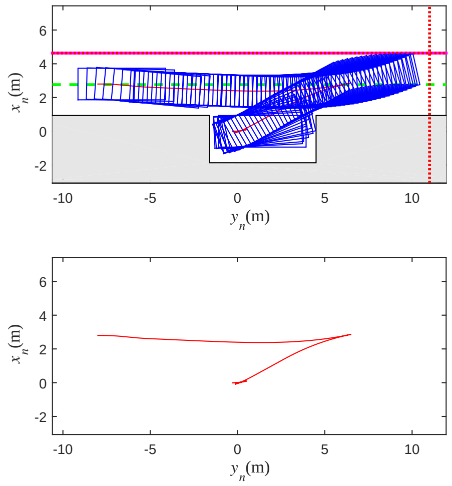

FIGURE 8. PARALLEL PARKING (TIGHT)

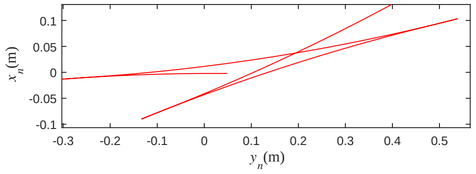

FIGURE 9. ZOOM-IN PATH

TABLE 2. PARKING ENVIRONMENT.

Scenario	W_l (m)	W_s (m)	x_n (m)	y_n (m)
Parallel (tight)	6.10	2.80	<4.63	<11
Parallel (wide)	7.15	2.80	<4.63	<12
Perpendicular (head-in)	5.50	2.74	<11.2	<4.5
Perpendicular (back-in)	5.50	2.74	<11.2	<6.5

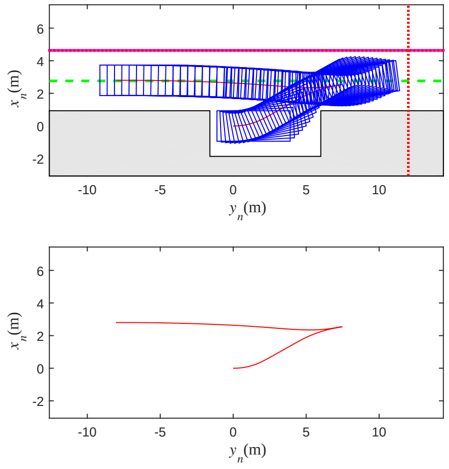

FIGURE 10. PARALLEL PARKING (WIDE)

$\cot\psi_{BR}$, and the $\delta(s)$-continuity condition: $\frac{d^2x_n}{dy_n^2}(y_{nBF}) = \tan\delta_{BF}/(L|\sin\psi_{BF}|^3)$, $\frac{d^2x_n}{dy_n^2}(y_{nBR}) = \tan\delta_{BR}/(L|\sin\psi_{BR}|^3)$.

3.5 Collision detection and switching

In the algorithm, collision detection is used while the virtual car is driving. When an upcoming collision between the car and the boundary of D is detected, to guarantee that the car remains in D, we change the sign of the switching signal. Any existing collision detection technique (e.g. those in [15, 24]) can be employed. If the car is far away from the boundary of D, we can turn off collision detection for a moment and reactivate it later, thereby greatly reducing the computational costs. If there are available sensors on the car for real-time collision detection, our algorithm can also work in a closed-loop manner.

4 Simulation results

In order to verify the effectiveness of the proposed path planning algorithm, four common parking scenarios are studied by simulation. The car's parameters which comes from the common mid-size car are $L = 2.91$m, $d_r = 0.97$m, $d_f = 1.14$m, $b = 0.93$m and $\delta_{\max} = 0.55$rad. The dimensions of the parking space and the restrictions on x_n and y_n in the different parking scenarios are given in Tab. 2, where W_l is the length of the long side of the parking space, W_s is the length of the short side of the parking space. The width of the drive aisle is 3.66m. In all scenarios, the radius of the avoidance circle is taken by the rule of thumb as $1.4b = 1.3$m. By the aforementioned rule of thumb, the middle line of the drive aisle is selected as the target line. In parallel parking and back-in perpendicular parking, the ready position q_r is selected as the position where the geometric center of the car is on the avoidance circle. In head-in perpendicular parking, q_r is selected as the position where the car's front end is on the avoidance circle. The parameters of the switching controllers are chosen as $K_1 = 0.9$, $K_2 = 4.5$, $K_3 = 15$, $K = 10$. It can be checked that the conditions in Theorem 3.1 and 3.2 are satisfied for all scenarios. The initial and final steering angles are 0 rad in all cases. The simulation was conducted in MATLAB with an Intel Core i5-2450M 2.5GHz CPU.

The simulation results are shown in Fig. 8-12, where the blue boxes illustrate the poses of the car along the path, the grey

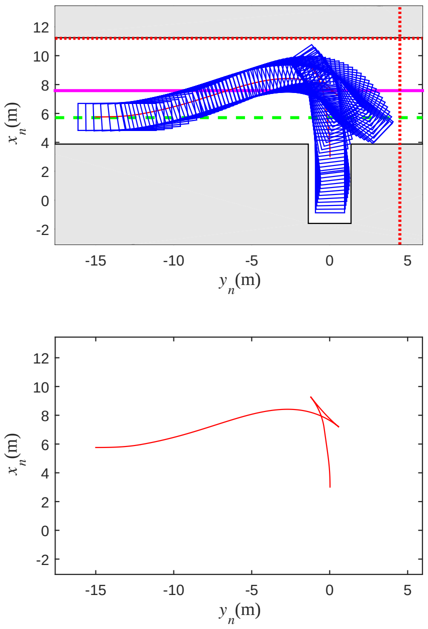

FIGURE 11. PERPENDICULAR PARKING (HEAD-IN)

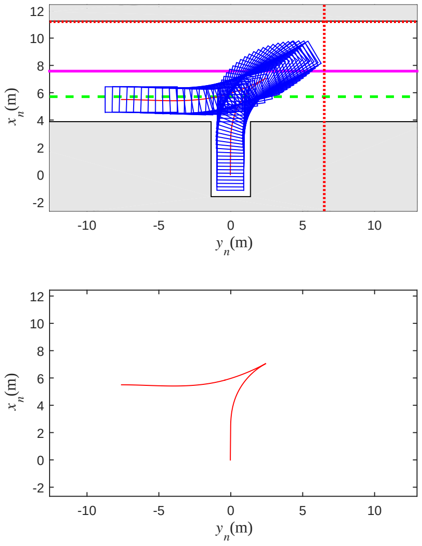

FIGURE 12. PERPENDICULAR PARKING (BACK-IN)

shadowed area stands for obstacles that the car must avoid, the red dotted lines are the restrictions on x_n and y_n, the green dashed line represent the target line, the magenta solid line represents the line that separates two drive aisles. The results for parallel parking in tight parking space and wide parking space are shown in Fig. 8 and Fig. 10 respectively. From Fig. 10, we can see that the algorithm is able to produce a one-maneuver path in the wide parking space. Fig. 8 shows that, in order to park a car in the tight parking space, several forward and backward moves are required. The details of the path in the parking space are shown in Fig. 9. Fig. 11 illustrates the head-in perpendicular parking scenario. Fig. 12 shows the back-in perpendicular parking scenario. Notice that head-in parking requires a larger maneuvering space than back-in parking does. The computing time is around 80ms for the parallel parking scenarios and around 60ms for the perpendicular parking scenarios. Comparing to the geometric planner, the computing time is on the same order of magnitude with the same class of CPUs. In the other approaches [18,24,32], it took more than 1s (inclouding initialization time) for parallel parking with the same class of CPUs. We point out that about half of the computing time of our algorithm spends on collision detection, which could be greatly shortened or eliminated if collision detection sensors are used in a closed-loop manner. Our algorithm can produce the entire path in 4 phases automatically whereas the geometric approach requires human's intervention in practical use, because it only automates phases 1 and 2. Unlike the geometric approach, our algorithm has high applicability to different scenarios and can cope with environmental changes. The smoothness of the path is guaranteed by the algorithm and attested by the simulation results. We can also see that, although the number of cusps is not explicitly considered in the design, it is small compared with other algorithms [1,24] for tight parking space. In addition, comparing to the existing algorithms, the proposed path planner can generate paths with relative short length.

5 Conclusion

In this paper, we proposed a four-phase path planning algorithm for autonomous car parking. The simulation results demonstrate that the algorithm is fast and able to cope with different tight parking scenarios in a unified manner. It is also shown that the resulted path is smooth and has very few cusps. The algorithm is highly automatic because little human's knowledge or intervention is involved in the design. In the future, the algorithm will be extended to path planning for on-road driving through both static and dynamic obstacles. We will also extend the algorithm to autonomous parking for a car with n-trailers.

REFERENCES

[1] Vorobieva, H., Glaser, S., Minoiu-Enache, N., and Mammar, S., 2015. "Automatic parallel parking in tiny spots: Path planning and control". *IEEE Trans. Intell. Transp. Syst.*, ***16***(1), pp. 396–410.

[2] Lo, Y., Rad, A., Wong, C., and Ho, M., 2003. "Automatic

parallel parking". In Intelligent Transportation Systems, 2003. Proc.. 2003 IEEE, Vol. 2, IEEE, pp. 1190–1193.

[3] Sungwoo, C., Boussard, C., and d'Andréa Novel, B., 2011. "Easy path planning and robust control for automatic parallel parking". *IFAC Proc. Volumes,* ***44***(1), pp. 656–661.

[4] Wang, C., Zhang, H., Yang, M., Wang, X., Ye, L., and Guo, C., 2014. "Automatic parking based on a bird's eye view vision system". *Adv. Mech. Eng.,* ***6***, p. 847406.

[5] Paromtchik, I. E., and Laugier, C., 1996. "Autonomous parallel parking of a nonholonomic vehicle". In Intelligent Vehicles Symposium, 1996., Proceedings of the 1996 IEEE, IEEE, pp. 13–18.

[6] Paromtchik, I. E., and Laugier, C., 1996. "Motion generation and control for parking an autonomous vehicle". In Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on, Vol. 4, IEEE, pp. 3117–3122.

[7] Paromtchik, I. E., 2003. "Planning control commands to assist in car maneuvers". In Proc. of the 11th IEEE Int. Conf. on Advanced Robotics, pp. 1308–1313.

[8] Pradalier, C., Vaussier, S., and Corke, P., 2005. "Path planning for a parking assistance system: Implementation and experimentation". *Australian Robotics and Automation Association, Sydney, Australia.*

[9] Gómez-Bravo, F., Cuesta, F., and Ollero, A., 2001. "Parallel and diagonal parking in nonholonomic autonomous vehicles". *Eng. Appl. Artif. Intell.,* ***14***(4), pp. 419–434.

[10] Li, T.-H., and Chang, S.-J., 2003. "Autonomous fuzzy parking control of a car-like mobile robot". *IEEE Trans. Syst., Man, Cybern. A, Syst. ,Humans,* ***33***(4), pp. 451–465.

[11] Razinkova, A., Cho, H.-C., and Jeon, H.-T., 2012. "An intelligent auto parking system for vehicles". *Int. J. Fuzzy Log. Intell. Syst.,* ***12***(3), pp. 226–231.

[12] Laumond, J.-P., Jacobs, P. E., Taix, M., and Murray, R. M., 1994. "A motion planner for nonholonomic mobile robots". *IEEE Trans. Robot. Autom.,* ***10***(5), pp. 577–593.

[13] Laumond, J., Sekhavat, S., and Vaisset, M., 1994. "Collision-free motion planning for a nonholonomic mobile robot with trailers". *IFAC Proceedings Volumes,* ***27***(14), pp. 171–177.

[14] Jiang, K., Seneviratne, L. D., and Earles, S., 1999. "A shortest path based path planning algorithm for nonholonomic mobile robots". *J. Intell. Robot. Syst.,* ***24***(4), pp. 347–366.

[15] LaValle, S. M., and Kuffner Jr, J. J., 2000. "Rapidly-exploring random trees: Progress and prospects".

[16] Pepy, R., Lambert, A., and Mounier, H., 2006. "Path planning using a dynamic vehicle model". In Information and Communication Technologies, 2006. ICTTA'06. 2nd, Vol. 1, IEEE, pp. 781–786.

[17] 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. Control Syst. Technol.,* ***17***(5), pp. 1105–1118.

[18] Han, L., Do, Q. H., and Mita, S., 2011. "Unified path planner for parking an autonomous vehicle based on rrt". In Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE, pp. 5622–5627.

[19] Nilsson, N. J., 2014. *Principles of artificial intelligence*. Morgan Kaufmann.

[20] Dolgov, D., Thrun, S., Montemerlo, M., and Diebel, J., 2008. "Practical search techniques in path planning for autonomous driving". *Ann Arbor,* ***1001***(48105), pp. 18–80.

[21] Kim, D., and Chung, W., 2008. "Motion planning for car-parking using the slice projection technique". In Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, IEEE, pp. 1050–1055.

[22] Yang, S. X., and Meng, M.-H., 2003. "Real-time collision-free motion planning of a mobile robot using a neural dynamics-based approach". *IEEE Trans. Neural Netw.,* ***14***(6), pp. 1541–1552.

[23] Kondak, K., and Hommel, G., 2001. "Computation of time optimal movements for autonomous parking of nonholonomic mobile platforms". In Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on, Vol. 3, IEEE, pp. 2698–2703.

[24] Li, B., Wang, K., and Shao, Z., 2016. "Time-optimal maneuver planning in automatic parallel parking using a simultaneous dynamic optimization approach". *IEEE Trans. Intell. Transp. Syst.,* ***17***(11), pp. 3263–3274.

[25] Zips, P., Bock, M., and Kugi, A., 2013. "A fast motion planning algorithm for car parking based on static optimization". In Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ Int. Conf. on, IEEE, pp. 2392–2397.

[26] Zips, P., Böck, M., and Kugi, A., 2016. "Optimisation based path planning for car parking in narrow environments". *Robotics and Autonomous Systems,* ***79***, pp. 1–11.

[27] Sekhavat, S., and Chyba, M., 1999. "Nonholonomic deformation of a potential field for motion planning". In Robotics and Automation, 1999. Proc.. 1999 IEEE Int. Conf. on, Vol. 1, IEEE, pp. 817–822.

[28] Kamgarpour, M., and Tomlin, C., 2012. "On optimal control of non-autonomous switched systems with a fixed mode sequence". *Automatica,* ***48***(6), pp. 1177–1181.

[29] Lin, H., and Antsaklis, P. J., 2009. "Stability and stabilizability of switched linear systems: a survey of recent results". *IEEE Trans. Autom. Control,* ***54***(2), pp. 308–322.

[30] Liberzon, D., 2012. *Switching in systems and control*. Springer Science & Business Media.

[31] Bacciotti, A., and Mazzi, L., 2005. "An invariance principle for nonlinear switched systems". *Systems & Control Letters,* ***54***(11), pp. 1109–1119.

[32] Kwon, H., and Chung, W., 2014. "Performance analysis of path planners for car-like vehicles toward automatic parking control". *Intelligent Service Robotics,* ***7***(1), pp. 15–23.