Time-Optimal Maneuver Planning in Automatic Parallel Parking Using a Simultaneous Dynamic Optimization Approach

Bai Li, Kexin Wang, and Zhijiang Shao

***Abstract*—Autonomous parking has been a widely developed branch of intelligent transportation systems. In autonomous parking, maneuver planning is a crucial procedure that determines how intelligent the entire parking system is. This paper concerns planning time-optimal parallel parking maneuvers in a straightforward, accurate, and purely objective way. A unified dynamic optimization framework is established, which includes the vehicle kinematics, physical restrictions, collision-avoidance constraints, and an optimization objective. Interior-point method (IPM)-based simultaneous dynamic optimization methodology is adopted to solve the formulated dynamic optimization problem numerically. Given that near-feasible solutions have been widely acknowledged to ease optimizing nonlinear programs (NLPs), a critical region-based initialization strategy is proposed to facilitate the offline NLP-solving process, a lookup table-based strategy is proposed to guarantee the on-site planning performance, and a receding-horizon optimization framework is proposed for online maneuver planning. A series of parallel parking cases is tested, and simulation results demonstrate that our proposal is efficient even when the slot length is merely 10.19% larger than the car length. As a unified maneuver planner, our adopted IPM-based simultaneous dynamic optimization method can deal with any user-specified demand provided that it can be explicitly described.**

***Index Terms*—Autonomous vehicles, motion planning, time-optimal control, nonholonomic systems, nonlinear optimization, intelligent vehicle.**

I. Introduction

PARALLEL parking refers to stopping a car parallel to the road, in line with other parked cars. Parallel parking commonly requires initially driving past the parking spot and then reversing into that space. Subsequent position adjustment may require forward and reverse gears in handling tiny spots.

To date, urban parking spots have become narrower than before, thus rendering manual parking operations more challenging [1]. Unskilled parking operations may disturb the surrounding road users and even cause a traffic jam [2]. Besides that, inexperienced drivers who are reluctant to try tiny parking spots would have to circle around blocks, thereby contributing to additional air pollution, fuel consumption, and congestion [3]: a report indicates that vehicles along the Columbus Avenue of New York wind around for totally 590,000 km searching for parking spots every year, resulting in 325 tons of carbon dioxide emission and 50,000 wasted hours [4].

To ease the manual parking burdens, parking assist systems have been developed by automobile manufacturers. Toyota Prius is the first model equipped with an automatic parking system, which appeared on the market in 2003. Ever since the first commercial availability, productions developed by Audi, BMW, Ford, Land Rover, Mercedes-Benz, and Nissan have also hit the market [5], [6]. In spite of the prosperous developments, challenges remain in coping with harsh parking scenarios. A recent publication even alleges that no real product in this industry is known to outperform a human driver in making smart enough decisions [7].

Autonomous parking generally consists of three sequential stages, namely, circumstance perception, maneuver planning, and control implementation [8]. Circumstance perception contributes to detecting a parking slot, maneuver planning refers to the generation of parking motions according to the detected circumstance, while control implementation concerns how to execute the planned maneuvers. Prevailing research studies in this community care more about the control stage than about the maneuver planning stage, they believe that control implementation compensates for the roughly planned maneuvers but this viewpoint is misleading, especially in dealing with harsh scenarios: suppose an automobile drives along a narrow and delicate passage, a rough maneuver planner is not capable to safely conclude whether this passage is passable! In this sense, it is the maneuver planning stage that makes decisions during a parking process, i.e., to decide how intelligent an entire parking assist system is [6]. This study concerns about the maneuver planning issue in parallel parking.

Maneuver planning of wheeled vehicles has attracted interest in the community of robotics for long. Wheeled locomotion imposes non-integrable kinematic constraints on the vehicle motion, making the planning problem difficult [9]. The authors

Manuscript received December 5, 2014; revised May 28, 2015, September 15, 2015, and January 5, 2016; accepted March 12, 2016. Date of publication May 2, 2016; date of current version October 28, 2016. This work was supported in part by the 973 Program of China under Grant 2012CB720503; by the National Natural Science Foundation of China under Grant 61374167; and by the National College Students' Innovative and Entrepreneurial Training Program under Grant 201210006050. The Associate Editor for this paper was M. Da Lio.

Source codes of this work are available at https://www.researchgate.net/publication/298433602. For the convenience of understanding, a video that contains the simulated cases is available at http://www.tudou.com/programs/view/I9XRLgUaqnk/.

B. Li and K. Wang are with the College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: libai@zju.edu.cn; kxwang@iipc.zju.edu.cn).

Z. Shao is with State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: szj@zju.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2016.2546386

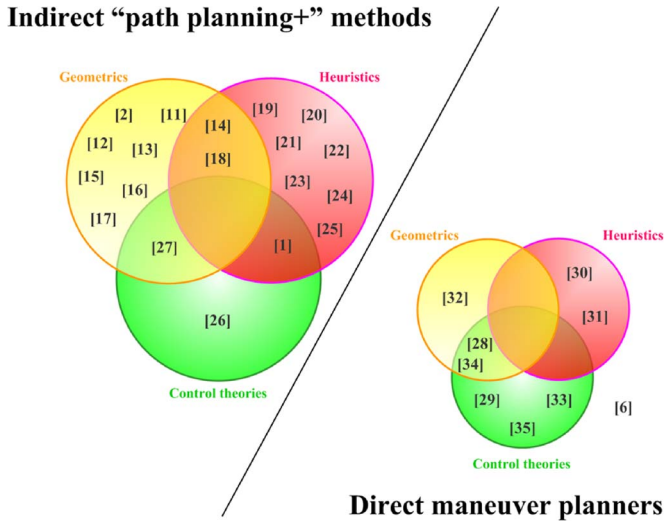

Fig. 1. Comprehensive review of prevailing maneuver planning methodologies in robotics.

classify all the maneuver planning methodologies into two categories: direct methods and indirect methods [10]. Indirect methods refer to the ones that convert the original maneuver planning scheme to other types of problems such as "path-planning + tracking" and "path-planning + trajectory generation". The prevailing "path-planning+" methodologies are broadly classified into three branches: geometric, heuristic, and analytical path planners [6]. Geometric planners originate from the pioneering works of Dubins [11] and of Reeds and Shepp [12], wherein paths consist of straight line segments and circular arcs with minimum radius [13]–[15]. In addition, smooth paths, which are expressed by splines, polynomials or clothoids, have been considered to ease the subsequent path tracking procedure [2], [16]–[18]. The second branch concerns about heuristics, including machine learning [19], sampling [20], [21], search theory [22]–[24], fuzzy logic [25], etc. The third branch involves control theories such as Pontryagin's maximum principle [26] and differential geometric control theory [27]. Compared to the aforementioned indirect methods that decouple the original problem into multiple stages, direct methods are characterized by solving the maneuver planning problem directly. Thus differential equations have been commonly used to describe the time-related dynamic process in a straightforward and clear way. Given that an analytical solution to a dynamic problem is not available in general, prevailing methodologies are commonly on the basis of specific solution structures (e.g., Reeds-Shepp based and bang-bang control based trajectories) [6], [28]–[35]. Specifically, geometrics, heuristics, and control theories are incorporated in some of the direct planners to facilitate the solving process, although the original maneuver planning problem is directly dealt with. A collection of the aforementioned maneuver planning methodologies is depicted in Fig. 1.

Determining whether one planner is better than another is difficult because they are applied in distinct scenarios and/or are on the basis of distinct descriptions. Broadly speaking, indirect methods decouple the original maneuver planning scheme into multiple stages. Although these aforementioned "path-planning+" methods are computationally cheap, they have difficulty in coping with complicated scenarios and/or time-related restrictions [10]. In essence, a "path-planning + tracking" method would shift the maneuver planning difficulties to the control stage, rather than cope with them! Given the limitations in the prevailing "path-planning+" methods, maneuver planning should be solving directly, without being separated into stages [36]. Direct methodologies are commonly involved in specific solution structures as aforementioned. Specific structures would reduce the available solution space, largely weakening the unification of a maneuver planner. For example, the heuristics-based models are usually intuitively understandable and reasonable, but they are capable of providing almost none solutions beyond human expectations/experiences. In addition, maneuver optimization is superior to planning merely feasible maneuvers, because the former nominally exploits every inch of a tiny parking spot.

This work aims to solve the parallel parking maneuver planning problem in a direct, straightforward, unified, and purely objective way. Specifically, an optimal control problem architecture is formulated, which consists of differential equations (representing vehicle kinematics), algebraic equations/inequalities (representing mechanical/environmental restrictions), and a user-specified optimization objective. Although formulating such a beneficial architecture is not difficult, computational challenges in problem solving has been a critical bottleneck that prevents most of previous studies from getting ahead. This bottleneck may become less crucial with the incessant developments in high-efficient CPUs and advanced numerical dynamic optimizers.

The remainder of this paper is organized as follows. Section II introduces an optimal control problem on the basis of the concerned parallel parking maneuver planning scheme. The aforementioned optimal control problem solver, together with facilitation strategies are proposed in Section III, followed by Section IV, wherein our proposal is tested via simulations. In-depth analyses about the simulation results are provided in Section V. Finally, conclusions are drawn in Section VI.

II. Problem Formulation

The original maneuver planning mission is described as an unified optimal control problem in this section.

A. Vehicle Kinematics

Given the low moving speed in real-world parking [26], an automobile (excluding its four wheels) is assumed as a rigid body, i.e., coupled dynamics of vehicle suspensions are not considered. Also, tire sideslip issue is omitted. We focus on vehicles with front steering wheels. Kinematics of a car-like vehicle is expressed as

$$\begin{cases} \frac{dx(t)}{dt} = v(t)\cdot\cos\theta(t) \\ \frac{dy(t)}{dt} = v(t)\cdot\sin\theta(t) \\ \frac{dv(t)}{dt} = a(t) \\ \frac{da(t)}{dt} = \text{jerk}(t) \\ \frac{d\theta(t)}{dt} = \frac{v(t)\cdot\tan\phi(t)}{l} \\ \frac{d\phi(t)}{dt} = \omega(t) \end{cases} \tag{1}$$

where (x, y) refers to the mid-point of rear wheel axis (see the reference point P in Fig. 2), θ refers to the orientation

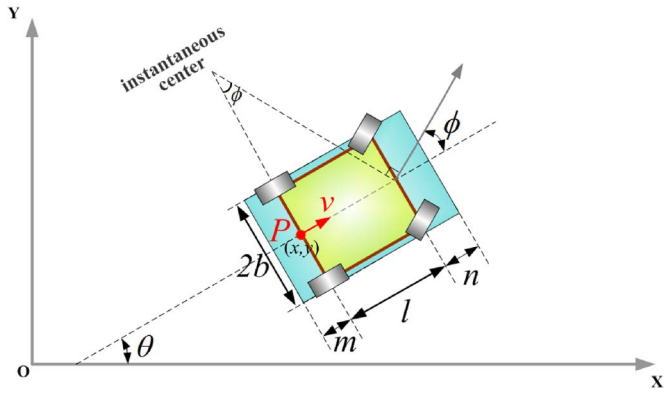

Fig. 2. Parametric notations related to vehicle size and kinematics.

angle, v stands for the linear velocity of P, a stands for the corresponding acceleration, jerk denotes the acceleration derivative, ϕ denotes the steering angle of front wheels, ω denotes the corresponding angular velocity, and l denotes the wheelbase (i.e., the distance between front and rear wheel axes) [4]. Other parametric settings related to the geometric size of an automobile include front overhang length n, rear overhang length m, and vehicle width $2b$, as depicted in Fig. 2.

B. Interior Constraints

In addition to vehicle kinematics, physical and mechanical constrains should be satisfied. In more detail,

$$\begin{cases} |a(t)| \le a_{\max} \\ |v(t)| \le v_{\max} \\ |\phi(t)| \le \Phi_{\max} \end{cases} \quad \text{for any } 0 \le t \le t_f \tag{2a}$$

where $t_f > 0$ denotes the (unknown in advance) completion time of the entire parking process. Rationales behind the aforementioned constraints are as follows: (i) changing $v(t)$ fast brings about discomfort to passengers, thus bounds are imposed on $a(t)$; (ii) a vehicle usually parks in a low speed, reserving reaction time for potential emergencies, thus bounds are imposed on $v(t)$; (iii) $\phi(t)$ is mechanically limited.

Besides (2a), the following two constraints are considered:

$$\begin{cases} |\text{jerk}(t)| \le \text{da}_{\max} \\ |\kappa'(t)| \le \text{d}\kappa_{\max}, \end{cases} \quad \text{for any } 0 \le t \le t_f \tag{2b}$$

where $\kappa'(t)$ denotes the derivative of instantaneous curvature $\kappa(t)$. $\kappa(t)$ and $\kappa'(t)$ are defined as

$$\begin{aligned} \kappa(t) &= \frac{\tan\phi(t)}{l} \\ \kappa'(t) &= \frac{\omega(t)}{l \cdot \cos^2\phi(t)}. \end{aligned} \tag{3}$$

Limitation of jerk results in smoothed actuator loads and reduced passenger discomfort [37]. Similarly, imposing bounds on $\kappa'(t)$ makes the curvature profile $\kappa(t)$ continuous and then eases the subsequent maneuver implementation procedure [38].

C. Exterior Restrictions

This subsection presents the formulation of exterior constraints (i.e., collision-free conditions). As a common practice, vehicle body is regarded as a rectangle in the 2D Euclidean space, and a parking slot is considered rectangular.

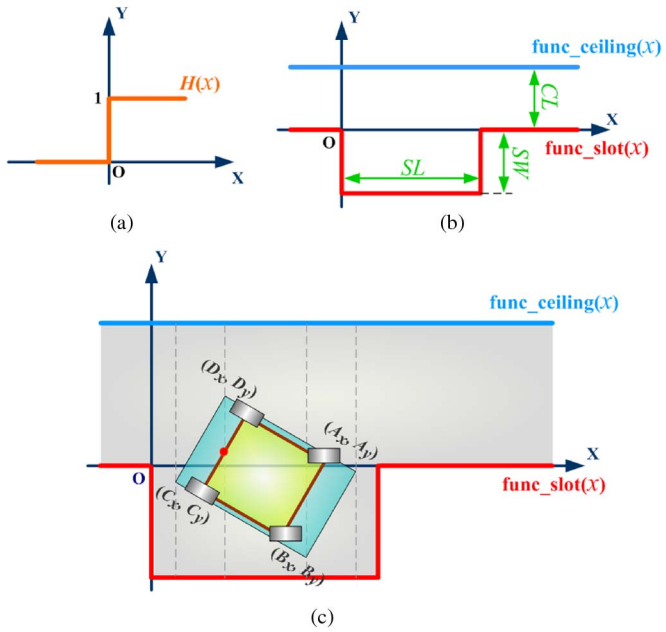

Fig. 3. Schematics on the formulation of collision-avoidance restrictions. (a) Basic Heaviside step function $H(x)$. (b) Two functions named func_slot(x) and func_ceiling(x) that describe the frontiers on both sides of the road. (c) Admissible space denoted by the shading region.

First, frontiers of a parallel parking slot is described through

$$\text{func\_slot}(x) = (-H(x) + H(x - \text{SL})) \cdot \text{SW} \tag{4}$$

where $\text{SL} > 0$ denotes parking slot length, $\text{SW} > 0$ denotes parking slot width, $H(x)$ stands for Heaviside step function (defined as $H(x) = 1$ when $x \ge 0$; otherwise, $H(x) = 0$), func_ceiling$(x) \equiv \text{CL} > 0$ defines frontiers on the other side of the road (Fig. 3). To avoid underlying collisions, a vehicle should locate above func_slot$(\cdot)$ and below func_ceiling$(\cdot)$, which is accurately described via inequalities as follows.

For the convex set properties, the necessary and sufficient collision-free condition between a rectangular car and func_ceiling$(\cdot)$ is that the four corner points (i.e. points $A(t)$, $B(t)$, $C(t)$, and $D(t)$ in Fig. 3(c)) locate below func_ceiling$(\cdot)$:

$$\begin{cases} A_y(t) \le \text{CL} \\ B_y(t) \le \text{CL} \\ C_y(t) \le \text{CL} \\ D_y(t) \le \text{CL}, \end{cases} \quad \text{for any } t \in [0, t_f]. \tag{5}$$

At any specific moment, locations of four corner points are determined provided that $x(t)$, $y(t)$, and $\theta(t)$ are given:

$$\begin{cases} A = (A_x, A_y) \\ \quad = (x + (l+n)\cdot\cos\theta - b\cdot\sin\theta,\ y + (l+n)\cdot\sin\theta + b\cdot\cos\theta) \\ B = (B_x, B_y) \\ \quad = (x + (l+n)\cdot\cos\theta + b\cdot\sin\theta,\ y + (l+n)\cdot\sin\theta - b\cdot\cos\theta) \\ C = (C_x, C_y) = (x - m\cdot\cos\theta + b\cdot\sin\theta,\ y - m\cdot\sin\theta - b\cdot\cos\theta) \\ D = (D_x, D_y) = (x - m\cdot\cos\theta - b\cdot\sin\theta,\ y - m\cdot\sin\theta + b\cdot\cos\theta). \end{cases} \tag{6}$$

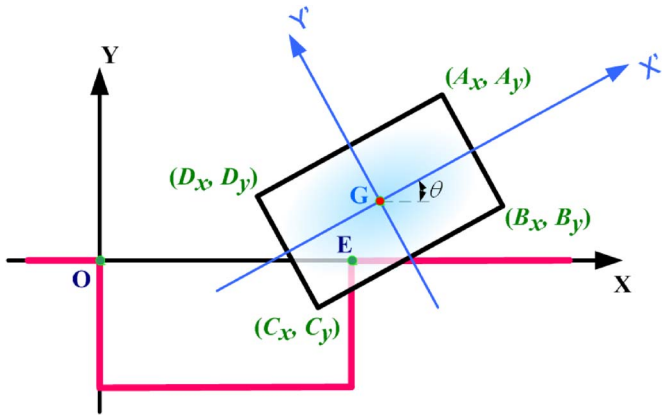

Fig. 4. Schematics on the necessary and sufficient condition of remaining above func_slot(·).

To guarantee that the concerned vehicle remains above func_slot(·), we have

$$\begin{cases} A_y \geq \text{func\_slot}(A_x) \\ B_y \geq \text{func\_slot}(B_x) \\ C_y \geq \text{func\_slot}(C_x) \\ D_y \geq \text{func\_slot}(D_x) \end{cases} \tag{7}$$

which only serves as a necessary condition. That is because the vehicle can still hit the function func_slot(·) in two cusps (i.e., points $O = (0, 0)$ and $E = (\text{SL}, 0)$ in Fig. 4) even when the four corner points locate above func_slot(·). Fig. 4 depicts an example wherein point E hits in the vehicle through edge $\overline{\text{BC}}$ whereas the four corner points remain above func_slot(·). Thus we need additional efforts to prevent that points O and E are inside the rectangular region ABCD. To begin with, the geometric center of rectangle ABCD is denoted as point G, which locates at $(x+(((l+n-m)\cdot\cos\theta)/2), y+(((l+n-m)\cdot\sin\theta)/2))$. Then, a new frame of axes is established, wherein the new origin is G and X' axis points the vehicle orientation (Fig. 4). Thereafter, the coordinates of O and E are transformed from XOY frame to X′GY′ frame. Given that the X′GY′ frame is obtained through translation (according to $\overrightarrow{\text{OG}}$) and rotation (according to θ in anticlockwise) of the XOY frame, coordinates of O and E in the X′GY′ frame can be presented by

$$\begin{aligned} O_{\text{X}'\text{GY}'} &= \left(O'_x, O'_y\right) \\ &= \left(-x\cdot\cos\theta - y\cdot\sin\theta - \frac{l+n-m}{2}, \right. \\ &\qquad \left. x\cdot\sin\theta - y\cdot\cos\theta \right) \end{aligned} \tag{8a}$$

$$\begin{aligned} E_{\text{X}'\text{GY}'} &= \left(E'_x, E'_y\right) \\ &= \left(-x\cdot\cos\theta - y\cdot\sin\theta - \frac{l+n-m}{2} \right. \\ &\qquad \left. + \text{SL}\cdot\cos\theta,\ x\cdot\sin\theta - y\cdot\cos\theta - \text{SL}\cdot\sin\theta \right). \end{aligned} \tag{8b}$$

Thus, O and E locate outside the rectangular vehicle region if and only if

$$|O'_x| \geq \frac{l+m+n}{2}, \text{ when } \left|O'_y\right| \leq b \tag{9a}$$

$$|E'_x| \geq \frac{l+m+n}{2}, \text{ when } \left|E'_y\right| \leq b. \tag{9b}$$

In summary, the necessary and sufficient collision-avoidance condition for a vehicle that is above func_slot(·) is a satisfaction of (7)–(9).

D. Initial and Terminal Conditions

Initial conditions refer to the initial status of a vehicle when parking begins, and they should be specified via localization and perception ahead of time. Terminal conditions, on the other hand, stand for the criteria for terminating the concerned parking process: (i) a parking process ends with a full stop, i.e., $v(t_f) = 0$ and $a(t_f) = 0$; (ii) an inside-slot terminal requirement is defined as

$$\begin{cases} A_y(t_f) \leq 0 \\ B_y(t_f) \leq 0 \\ C_y(t_f) \leq 0 \\ D_y(t_f) \leq 0. \end{cases} \tag{10}$$

Requirement (ii) implies that a vehicle is not required to be parked exactly parallel to the road, and it caters for true needs: once the vehicle is fully inside the slot, it is nominally free of collision risks.

E. Overall Dynamic Optimization Framework

The aforementioned kinematics and restrictions, together with a specified optimization objective, form an optimal control problem. Specifically, completion time t_f is chosen as the minimization objective in this work.

If the concerned off-line optimal control problem is solved online, it becomes the receding horizon optimization stage in nonlinear model predictive control (NMPC). Both on-line and off-line optimization schemes are collectively called dynamic optimization problems. In addition to the aforementioned constraints and conditions, any other user-specified demand can be incorporated in our formulated dynamic optimization problem, thereby formulating a unified framework, rather than dealing with a few special cases.

III. Dynamic Optimization Problem Solver

Describing the parallel parking maneuver planning scheme as a unified dynamic optimization problem is easy, nonetheless, the problem-solving process is so difficult as to have hindered most of research efforts from investigating further. A high-efficient dynamic optimizer, simultaneous dynamic optimization methodology, is adopted to solve our formulated problem. Specifically, the original infinite-dimensional dynamic optimization problem is discretized into a finite-dimensional nonlinear programming (NLP) problem, which is solved via

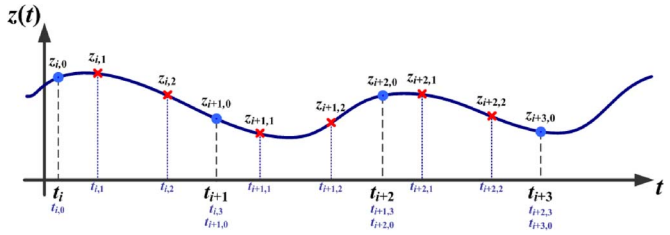

Fig. 5. Collocation of finite elements and interpolation points for differential state variables.

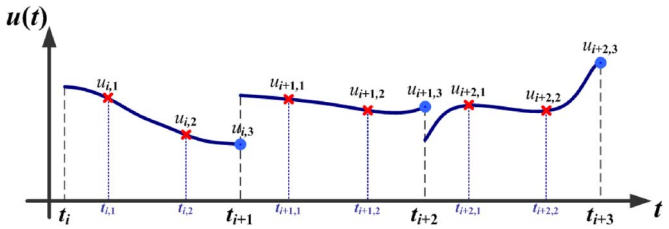

Fig. 6. Collocation of finite elements and interpolation points for control variables.

interior-point method (IPM) thereafter. Strategies that facilitate the NLP-solving process are preliminarily proposed.

A. Dynamic Optimization Problem Solver

Simultaneous dynamic optimization method is featured by discretizing not only control variables but state variables as well, thereby being equivalent to a fully implicit Runge-Kutta method with excellent stability and high order accuracy [39].

Our concerned dynamic optimization problem in Section II can be generalized as

$$\min\ \varphi\left(z(t_f)\right) \quad \text{s.t.} \begin{cases} \frac{dz(t)}{dt} = F\left(z(t), \zeta(t), u(t)\right) \\ G\left(z(t), \zeta(t), u(t)\right) \le 0 \\ z(0) = z_0 \\ z(t_f) = z_{t_f} \\ t \in [0, t_f] \end{cases} \tag{11}$$

wherein $\varphi(\cdot)$ refers to the minimization criterion, $z(t)$ refers to differential state variables, $\zeta(t)$ refers to the algebraic state variables, and $u(t)$ refers to the control variables. The decision variables in (11) include $z(t)$, $\zeta(t)$, $u(t)$, and t_f. First, the time domain $[0, t_f]$ is divided into N_{fe} equidistant intervals $\{[t_{i-1}, t_i] | i = 1, 2, \ldots, N_{\text{fe}}\}$, where $t_0 = 0$ and $t_{N_{\text{fe}}} = t_f$. Thus, duration of each element is written as $h_i = t_i - t_{i-1} = t_f / N_{\text{fe}},\ i = 1, 2, \ldots, N_{\text{fe}}$. For the consistency with previous studies in this community, those intervals are referred to as "finite elements" in the remainder of this paper. Second, in each finite element, $(K+1)$ collocation points $\{z_{ij} | j = 0, 1, 2, \ldots, K\}$ (see Fig. 5) are taken as decision variables whereby to approximate the differential states $z(t)$ via piecewise Lagrange polynomial:

$$z(t) = \sum_{j=0}^{K} \left(z_{ij} \cdot \prod_{k=0, \neq j}^{K} \frac{(\tau - \tau_k)}{(\tau_j - \tau_k)} \right) \tag{12}$$

where $t = t_{i-1} + h \cdot \tau$ and $\tau \in [0, 1]$. Radau or Gaussian points $\tau_K = 1$ and $\tau_k\ (k = 1, \ldots, K-1)$ are chosen to satisfy orthogonal properties when K is determined. The Lagrange polynomial represented in (12) has a desirable property that $z(t_{ij}) = z_{ij}$, where $t_{ij} = t_{i-1} + h \cdot \tau_j$. In addition, continuity of the differential state variables at element boundaries is considered by enforcing $\tau_0 = 0$ and

$$z_{i+1,0} = \sum_{j=0}^{K} \left(z_{ij} \cdot \prod_{k=0, \neq j}^{K} \frac{(1 - \tau_k)}{(\tau_j - \tau_k)} \right) = z_{i,K},\ i = 1, 2 \ldots, N_{\text{fe}} - 1. \tag{13}$$

Similarly, control variables $u(t)$ and algebraic states $\zeta(t)$ can be represented by piecewise Lagrange polynomials:

$$u(t) = \sum_{j=1}^{K} \left(u_{ij} \cdot \prod_{k=1, \neq j}^{K} \frac{(\tau - \tau_k)}{(\tau_j - \tau_k)} \right) \tag{14}$$

$$\zeta(t) = \sum_{j=1}^{K} \left(\zeta_{ij} \cdot \prod_{k=1, \neq j}^{K} \frac{(\tau - \tau_k)}{(\tau_j - \tau_k)} \right). \tag{15}$$

Herein, $u(t)$ and $\zeta(t)$ can be discontinuous at finite element boundaries, as depicted in Fig. 6. Substituting (12)–(15) into (11) yields an NLP formulation:

$$\min_{z_{ij}, \zeta_{ij}, u_{ij}}\ \varphi\left(z(t_f)\right) \quad \text{s.t.} \begin{cases} \sum_{k=0}^{K} \left(\left. \frac{d\psi_j(\tau)}{d\tau} \right|_{\tau = \tau_k} \cdot z_{ik} \right) - h \cdot F(z_{ij}, \zeta_{ij}, u_{ij}) = 0 \\ \psi_j(\tau) = \prod_{k=0, \neq j}^{K} \frac{(\tau - \tau_k)}{(\tau_j - \tau_k)} \\ h = \frac{t_f}{N_{\text{fe}}} \\ G(z_{ij}, \zeta_{ij}, u_{ij}) \le 0 \\ z_{i_1+1,0} = z_{i_1,K} \\ z_{1,0} = z_0 \\ z_{N_{\text{fe}},K} = z_{t_f} \end{cases}$$
$$i = 1, 2 \ldots, N_{\text{fe}},\ i_1 = 1, 2 \ldots, N_{\text{fe}} - 1,\ j = 1, 2, \ldots, K. \tag{16}$$

After the discretization, IPM is utilized to compute the decision variables, i.e., z_{ij}, ζ_{ij}, and u_{ij} as well as t_f, so as to minimize $\varphi(z(t_f))$. As a gradient-based optimizer, IPM can handle NLP problems with intricate equalities and bound constraints. Specifically, bound constraints are incorporated in the optimization objective in barrier-parameter-multiplied logarithmic forms, and the original NLP is transformed into an equality constrained barrier problem. For each specified barrier parameter, the barrier problem is solved inside the feasible domain. As the barrier parameter approaches zero, solutions of the barrier problems gradually converge to that of the original NLP problem. Interested readers may consult [40] for details.

B. Connections to Parking Maneuver Planning Scheme

Principle of the IPM-based simultaneous method in solving a generalized dynamic optimization problem (11) is briefly presented in the preceding subsection. This subsection targets on how the IPM-based simultaneous method solves our concerned maneuver planning problem.

First, the minimization criterion $\varphi(z(t_f))$ is equal to t_f since time-optimal maneuvers are expected in this study. Second, $z(t)$

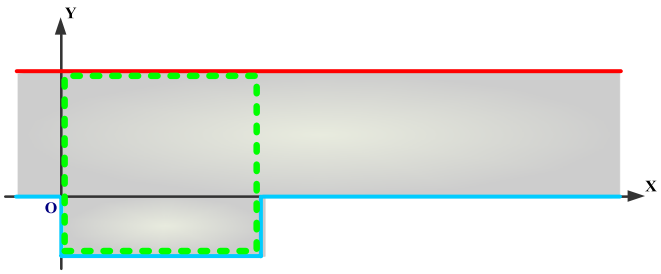

Fig. 7. Schematics on the concept of "critical region" in our offline initialization strategy.

represents all the differential state variables in (1), namely, $x(t)$, $y(t)$, $v(t)$, $a(t)$, $\theta(t)$, and $\phi(t)$. Third, $\zeta(t)$ represents all the state variables that are not differentiated, e.g., $A_x(t)$, $B_x(t)$, and $C_x(t)$. Fourth, $u(t)$ represents the control variables, i.e., $\omega(t)$ and jerk(t). Fifth, $G(z(t), \zeta(t), u(t)) \leq 0$ in (16) cover all the algebraic equalities/inequalities formulated in Section II.

C. Off-Line Initialization of IPM

Although IPM is capable of dealing with NLPs with millions of decision variables and constraints [39], yet the optimization process may converge slowly or even converge to infeasibility in solving complicated problems. Given that near-feasible initial guess has been widely acknowledged to facilitate the solving of a complicated NLP problem [41], [42], this subsection concerns about how to generate smart initial guess so as to facilitate the off-line NLP-solving process. Later, the action to generate efficient initial guess generation is called initialization of NLP.

Our proposed initialization strategy is inspired by the fact that a vehicle would adjust its configurations locally during the period of time right before the parking process terminates, especially when the slot is tiny. To begin with, a "critical region" is defined as the dashed box region in Fig. 7. Thereafter, a number of "new" problems, which are distinct from the original one, are pre-solved one after another. During that sequential pre-solving process, each optimized solution is taken as the initial guess in pre-solving the next problem. The aforementioned "new" problems contain additional restriction that the vehicle should remain within the critical region during some specified period of time right before the parking scheme is accomplished. Length of the "some specified period of time" is determined by integer $N_\chi \in [1, N_{\text{fe}}]$, thus, the additional restriction is described as

$$\begin{cases} -\text{SW} \leq A_y(t) \leq \text{CL} \\ -\text{SW} \leq B_y(t) \leq \text{CL} \\ -\text{SW} \leq C_y(t) \leq \text{CL} \\ -\text{SW} \leq D_y(t) \leq \text{CL} \\ 0 \leq A_x(t) \leq \text{SL} \\ 0 \leq B_x(t) \leq \text{SL} \\ 0 \leq B_x(t) \leq \text{SL} \\ 0 \leq B_x(t) \leq \text{SL}, \end{cases} \quad \text{for any } t \in [h \cdot N_\chi,\ t_f]. \tag{17}$$

Evidently, when N_χ is set small, restriction (17) would take effect during a relatively large proportion of time. Conversely, if $N_\chi = N_{\text{fe}}$, then the "new" problem is identical to the original one, because (17) is completely ineffective. We aim to pre-solve a number of "new" problems, wherein N_χ increases from 1 to N_{fe}.

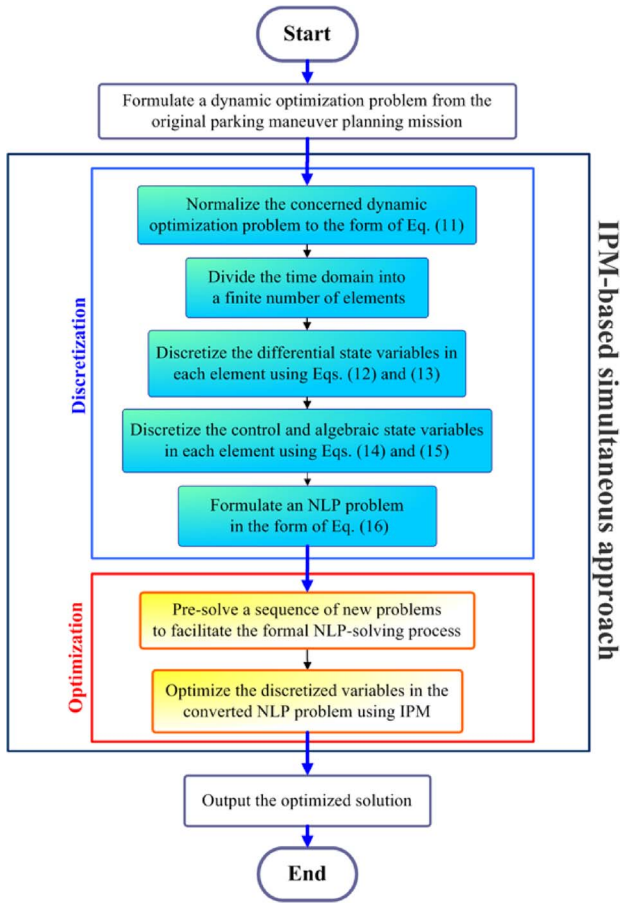

Fig. 8. Flowchart of IPM-based simultaneous methodology for parallel parking maneuver planning problems.

Readers may wonder why it makes sense to consider such a sequence of increasingly constrained problems. The rationale behind this proposal is that, once a car stays in the critical region, the collision avoidance conditions mentioned in Section II-C (i.e., (5), (7) and (9)) are replaced by (17). In other words, (17) is identical to the combination of (5), (7) and (9), $\forall\, t \in [h \cdot N_\chi, t_f]$. Given that func_slot$(\cdot)$ in (7) contains two non-differentiable points, and (9) contains condition judgments, (7) and (9) add to the difficulties in solving the original problem via a gradient-based method, IPM. This indicates that, the collision-avoidance constrains are largely simplified on $t \in [h \cdot N_\chi, t_f]$, although the additionally imposed restriction during that period of time appears to complicate the problem. Therefore, $N_\chi = 1$ represents the most simplified one among all the "new" problems. We begin the pre-solving process from the easiest one (without initial guess), and utilize the optimized solution as the initial guess to facilitate the subsequent "new" problem, i.e. the one with $N_\chi = 2$. This sequential process continues until $N_\chi = N_{\text{fe}}$, when the original problem is formally solved in the end.

Through the analyses mentioned above, we manage to facilitate the off-line maneuver planning process via a critical region based sequential strategy. An overall flowchart of this approach is demonstrated in Fig. 8.

D. On-Site Initialization

Although an off-line initialization strategy is proposed to facilitate the NLP-solving process, yet the computation is not cheap enough in real-world applications. In other words, an automobile may render congestion if it has to wait for the computation results before taking any action. To preliminarily address this issue, an on-site initialization procedure via look-up tables is introduced in this subsection.

In the first step, all of the parking cases should be represented by a finite number of scenarios. This preliminary study only considers distinctions in $x(0)$, $y(0)$, $\theta(0)$, and SL. For example, parameter SL that fall in the interval $5 \leq \text{SL} \leq 6$ can be represented by scenarios in the set {5.0, 5.2, 5.4, 5.6, 5.8, 6.0}. Second, each representative scenario is optimized off-line using the IPM-based simultaneous dynamic optimization approach, together with the sequential initialization strategy introduced in Section III-C. Thereafter, the optimized results of the representative cases are recorded in a database.

In dealing with a real-world parking maneuver planning scheme, solution to the closest recorded case is extracted from the database and utilized as the on-site initial guess. Through this, initial guess can be generated without involving the time-consuming sequential strategy. The maneuver optimization result, when available, is regarded as an "initial reference" utilized in the subsequent on-line replanning process.

E. On-Line Receding-Horizon Replanning

On-line replanning intends to react to potential scenario changes and compensate for errors in low-level controllers during the parking process.

Suppose that the receding-horizon replanning procedure executes N_{rep} times. Thus, time interval is divided into N_{rep} control horizons $\{T_{Ci} | i = 1, 2, \ldots, N_{\text{rep}}\}$. To begin with, the vehicle executes the initial reference maneuvers during the first control horizon T_{C1}; simultaneously, a receding-horizon replanning procedure starts at $t = 0$. Note that the maneuver replanning concerns part of the original problem, specifically, from $t = T_{C1}$ to $t = t_f$. In other words, maneuver replanning considers a scheme from where the vehicle is "predicted" to be at $t = T_{C1}$, the end of the first control horizon. Moving obstacles, if exist, locate at where they are "predicted" to be at $t = T_{C1}$. Herein, predicting future locations of the vehicle and the moving obstacles is not easy because of sensor/controller noises, kinematics mismatches, and exterior uncertainties. Capable filtering methodologies are needed to make reliable predictions of the future situations. In this preliminary study, we assume that the state estimation of each subsequent control horizon is accurate, therefore stochastic/system errors do not accumulate as the parking process continues. Ever since $t = T_{C1}$, the vehicle involves in the second control horizon and executes the replanned maneuvers until $t = T_{C1} + T_{C2}$. This recursive process continues until the end the N_{rep}th control horizon.

Regarding the setting of control horizon lengths, we consider utilizing the finite element structure in the simultaneous dynamic optimization method by setting $N_{\text{rep}} = N_{\text{fe}}$. In more detail, when $t = 0$, T_{C1} is set to t_f^*/N_{fe}, where t_f^* originates from the initial reference; during $t \in [0, T_{C1}]$, a reduced NLP problem with the latter $(N_{\text{fe}} - 1)$ finite elements is solved, and the corresponding part in the initial reference is taken as an on-line initial guess; the replanned result replaces the on-line initial guess; when $t = T_{C1}$, T_{C2} is set to $t_f'/(N_{\text{fe}} - 1)$, where t_f' originates from the on-line replanning result in the first control horizon; during $t \in [T_{C1}, T_{C1} + T_{C2}]$, an NLP problem with $(N_{\text{fe}} - 2)$ finite elements is solved, and the corresponding part in the current initial guess is utilized to facilitate the NLP-solving process. The subsequent process continues in a similar manner, and finally $t_f = \sum_{i=1}^{N_{\text{rep}}} T_{Ci}$ stands for the completion time of entire the parking process. Herein, on-line initialization is adopted to guarantee that each replanning procedure accomplishes before the corresponding control horizon ends. In contrast to the NMPC architectures with fixed control horizon lengths, our proposal smartly incorporates the receding-horizon replanning with our discretization formulation.

TABLE I
NOTATION OF USER-SPECIFIC PARAMETERS AND SETTINGS

Parameter	Description	Setting
ε_{tol}	Convergence tolerance in IPM	10^{-8}
κ_ε	Minimum absolute distance from the initial point to bound in IPM	10^{-4}
N_{fe}	Finite element number in the simultaneous approach	40
K	Interpolation point number in the simultaneous approach	3
SL	Slot length	-
SW	Slot width	2.0 m
CL	Road width	3.5 m
n	Front overhang length	0.839 m
l	Distance between front and back wheel axes	2.588 m
m	Rear overhang length	0.657 m
$2b$	Car width	1.771 m
Φ_{max}	Maximum steering angle	33°
a_{max}	Bound of acceleration	0.75 m/s^2
v_{max}	Bound of velocity	2.0 m/s
da_{max}	Bound of jerk	0.5 m/s^3
$\text{d}\kappa_{\text{max}}$	Bound of curvature derivative	0.6
N_{rep}	Control horizon number in on-line replanning	10

IV. SIMULATION RESULTS

To understand the behavior of IPM-based simultaneous methodology on maneuver planning, a series of simulations were conducted in "A Mathematical Programming Language" (AMPL) environment [43] and executed on an Intel Core i7-4710MQ CPU with 4 GB RAM that runs at 2.50 GHz. IPOPT, an open-source software package of IPM, in the 3.8.0 version [40] was utilized with default options. Concerned parameters and their settings are listed in Table I, where the car-size parameters originated from [2]. Regarding τ_i mentioned in Section III-A, we set $K = 3$ and adopted Radau points $\tau_1 = 0.1551$, $\tau_2 = 0.6450$, and $\tau_3 = 1$ [39].

Specifically, for the first six cases, the initial conditions include $x(0) = \text{SL} + m$, $y(0) = 1.5$, $v(0) = 0$, $a(0) = 0$, $\theta(0) = 0$, and $\phi(0) = 0$ while the terminal conditions include $v(t_f) = 0$,

TABLE II
MANEUVER PLANNING CASES 1–10

Case no.	Parking slot size	Special requirement(s)	t_f (sec)	Computation time (sec)
1	$SL = 6.0$ m	-	7.521	41.868
2	$SL = 5.5$ m	-	9.242	47.152
3	$SL = 5.0$ m	-	11.905	59.726
4	$SL = 4.5$ m	-	33.849	101.411
5	$SL = 6.0$ m	same with Case 1 except that $N_{fe} = 80$	7.522	136.400
6	$SL = 6.0$ m	same with Case 1 except that $da_{max} = 0.25$ m/s^3	9.344	182.394
7	$SL = 5.0$ m	same with Case 3 except that $\begin{cases} x(0) = -5.14 \text{ m} \\ y(0) = 1.41 \text{ m} \\ \theta(0) = 13.18° \end{cases}$	18.426	193.578
8	$SL = 5.0$ m	same with Case 3 except that $\begin{cases} x(0) = 8.97 \text{ m} \\ y(0) = 2.17 \text{ m} \\ \theta(0) = -16.06° \\ v(0) = 0.36 \text{ m/s} \end{cases}$	16.131	178.845
9	$SL = 5.5$ m	same with Case 2 except that $\begin{cases} x(0) = 6.15 \text{ m} \\ y(0) = 1.55 \text{ m} \text{ and} \\ \theta(0) = 5° \end{cases}$ result of Case 2 is taken as the initial guess directly	9.483	0.332
10	SL changes	same with Case 1 except that result of Case 1 is taken as the initial reference and SL reduces continuously from 6.0 m to 5.0 m during the first 5.74 sec.	12.002	-

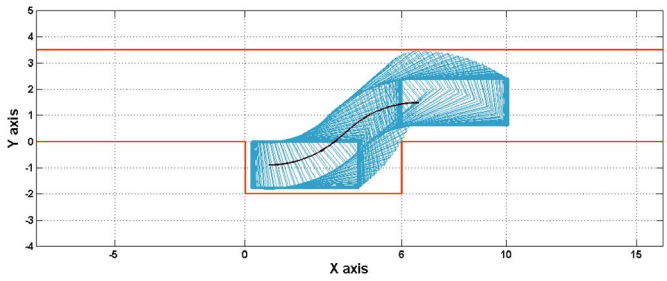

Fig. 9. Optimized motion in Case 1.

$a(t_f) = 0$ and the inside-slot restriction (10). In Cases 7 and 8, the initial conditions are irregular. Compared with the first eight cases that are optimized without initial guess, Case 9 is optimized based on a near-feasible initial guess. In Case 10, SL reduces continuously from 6.0 m to 5.0 m at the beginning 5.74 seconds. This case concerns a real-time maneuver planning mission with moving obstacle(s). Details of all the simulation cases are listed in Table II. The obtained parking maneuvers are partly plotted in Figs. 9–14, together with the optimized profiles in Figs. 15–19.

V. DISCUSSIONS

This section provides in-depth analyses behind the obtained simulation results.

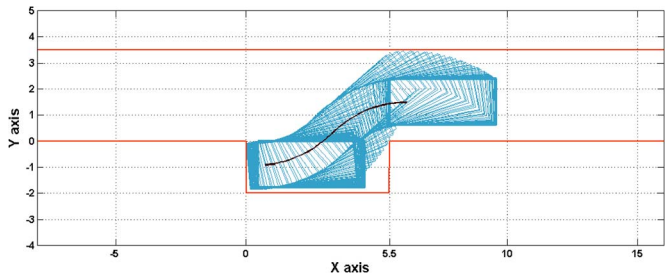

Fig. 10. Optimized motion in Case 2.

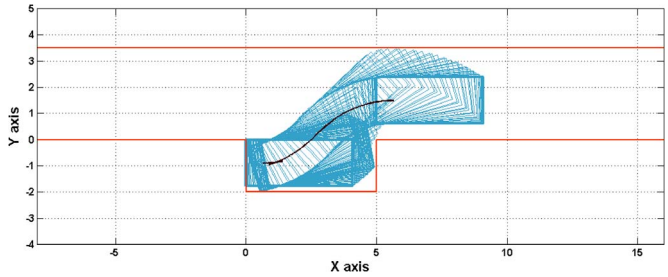

Fig. 11. Optimized motion in Case 3.

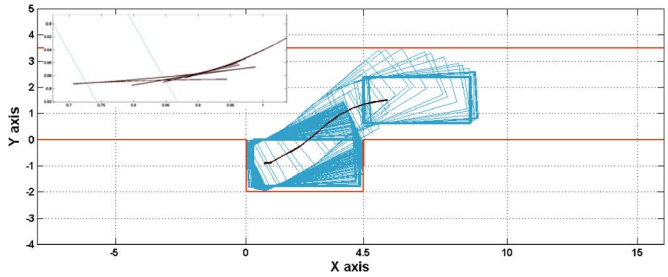

Fig. 12. Optimized motion in Case 4. The terminal maneuvers are zoomed in at the upper left corner.

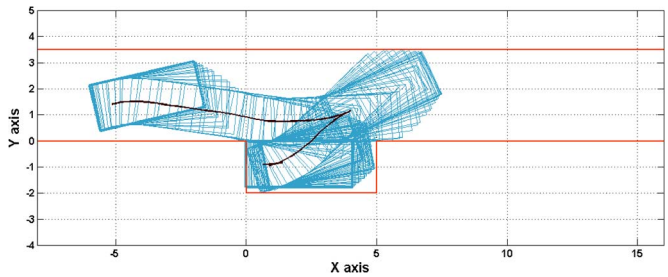

Fig. 13. Optimized motion in Case 7.

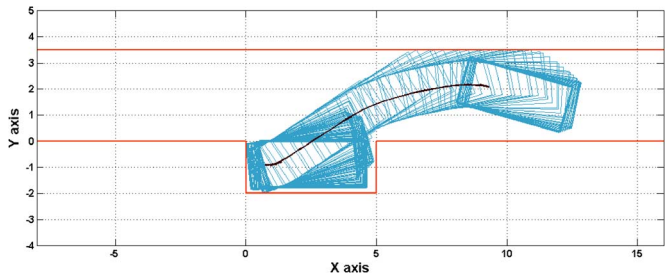

Fig. 14. Optimized motion in Case 8.

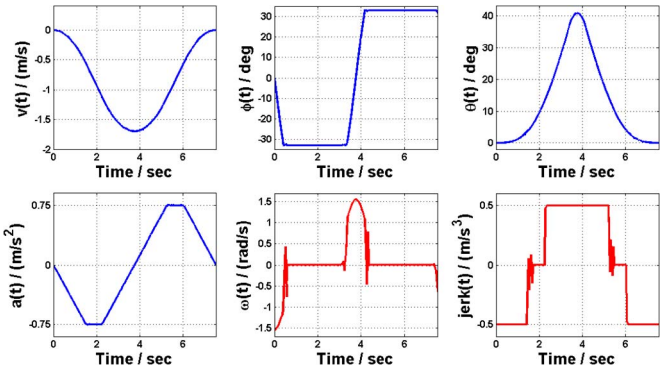

Fig. 15. Optimized control/state profiles in Case 1.

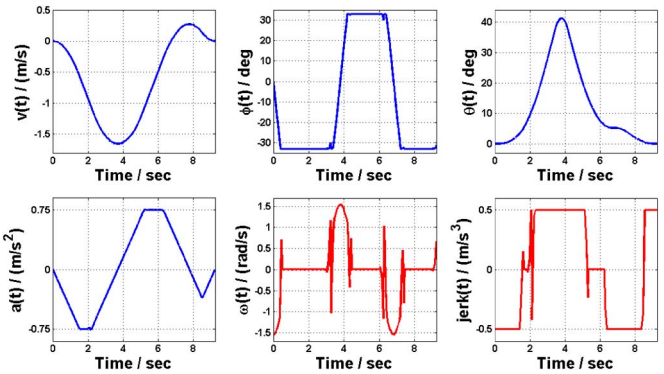

Fig. 16. Optimized control/state profiles in Case 2.

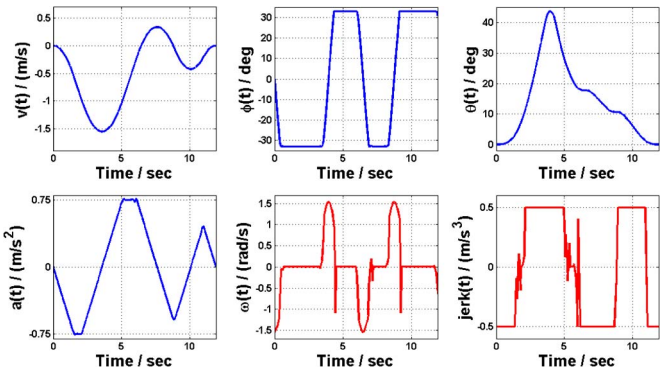

Fig. 17. Optimized control/state profiles in Case 3.

A. On the Optimized Maneuvers

The only distinct parameter among Cases 1–4 is SL, which varies from 6.0 m to 4.5 m, with the slot-car length ratio reduces from 1.47 to 1.10. When SL decreases, the car has to locally adjust with more maneuvers before fully completely enter the parallel parking slot. Here, the number of maneuvers can be reflected by the number of cusps in the path. When SL $= 4.5$ (Case 4), the slot length is merely 10.19% larger than the car length.

B. On the Optimized Profiles

The aforementioned number of maneuvers is also reflected in the optimized profile $v(t)$. In more detail, counting the times that $v(t)$ passes through $v = 0$ when $0 < t < t_f$ leads to the

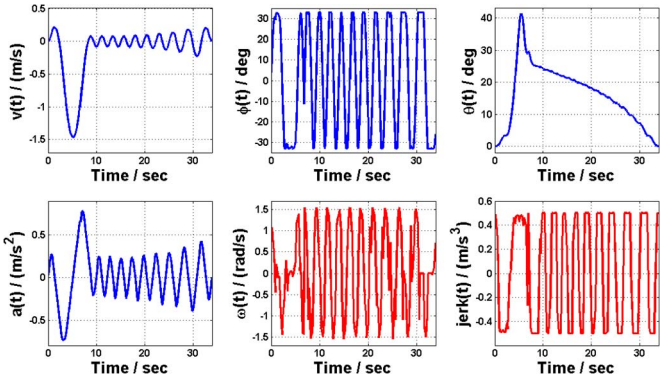

Fig. 18. Optimized control/state profiles in Case 4.

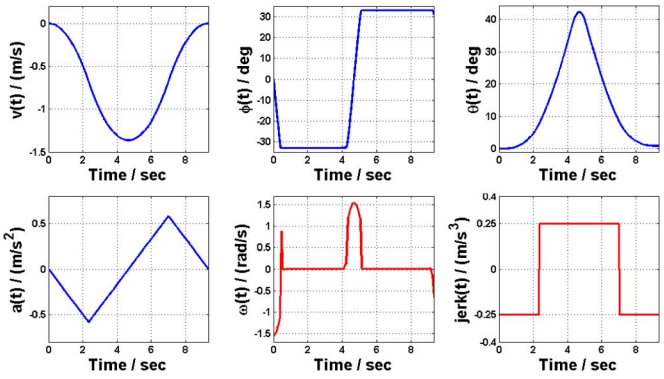

Fig. 19. Optimized control/state profiles in Case 6.

number of maneuvers. Besides that, the maneuver number issue reflects limitations of the utilized kinematic model: when the vehicle passes from backward motion to a forward motion, the acceleration profile $a(t)$ is nonzero, which is in conflict with common practice. This indicates that a kinematic model deserves improvements so as to be realistic.

Viewing the depicted profiles in Figs. 15–19, one may notice that the obtained state variables are smoother than the control variables (i.e., jerk(t) and $\omega(t)$), because of the requirements (13) in Section III-A. The jerk profiles generally form a bang-singular-bang mode, which is in accord with the theories in [44]. Given that an automobile can only steer the front wheels within a finite speed [45], [46], it appears that $\omega(t)$ profiles vary within a finite amplitude (especially in Fig. 18), although boundary constraints are not imposed on $\omega(t)$. Rationale behind this phenomenon is that, we have required that $\kappa'(t)$ should be bounded; definition of $\kappa'(t)$ in (3) directly renders $|\omega(t)| \leq d\kappa_{\max} \cdot l \cdot \cos^2 \phi(t)$, which implies that $|\omega(t)|$ is naturally bounded.

Moreover, it is interesting to note that $\phi(t_f)$ does not necessarily equal to zero, because the formulated terminal conditions do not include $\phi(t_f) = 0$. Herein, we consider that once a vehicle is safe the moment it is fully in a parallel parking slot, thus no extra issues deserve consideration. Letting $\phi(t_f)$ free not only eases the parking process, but also prevents unnecessary tire wears.

C. On the Off-Line Initialization Process

In Cases 1–4, there are as many as 2154 decision variables to optimize with 3114 constraints considered simultaneously in

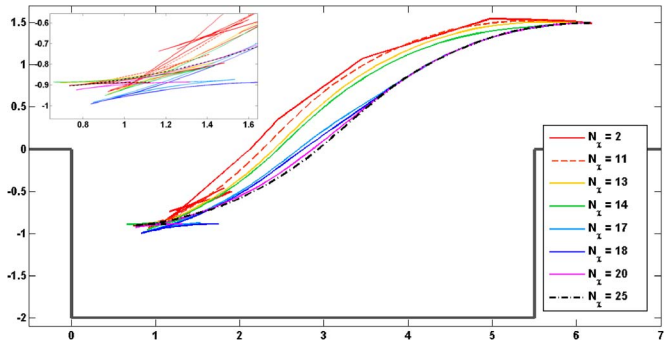

Fig. 20. Illustration on the performance of offline initialization strategy. The curves denote selected paths of point P during the parallel parking process (Case 2). The terminal maneuvers are zoomed in at the upper left corner.

our concerned NLP problem when $N_{\mathrm{fe}} = 40$. Without initial guess, it is almost beyond the ability of IPM to converge to optimums. Fig. 20 shows how our sequential initialization strategy works when Case 2 is taken for instance: a sequential initialization process begins with $N_\chi = 1$ and terminates when $N_\chi = 40$. Specifically, the beginning "new" problem with $N_\chi = 1$ additionally requests that the vehicle should stay in the critical region during the later $(40-1)/40 = 97.5\%$ period of the entire $[0, t_f]$. Suppose that the automobile needs at least 2 sec to enter the critical region, $N_\chi = 1$ would render that at least $2/(1-97.5\%) = 80$ sec to accomplish the parking mission. Thus, $N_\chi = 2$ yields the most painstaking efforts and longest completion time $t_f = 85.615$ than the other ones, as depicted in Fig. 20. When N_χ increases, the optimized completion time gradually decreases until solution to the original problem may be obtained at some specific threshold.

D. On the Unification of Our Proposal

Cases 5–8 are slightly different from Cases 1–4 for comparison. Case 5 investigates the discretization accuracy of the adopted numerical dynamic optimizer. A comparison between Cases 1 and 5 shows that t_f varies 0.001 sec when N_{fe} is doubled, reflecting that (i) setting $N_{\mathrm{fe}} = 40$ is acceptable at a level of 0.001, and (ii) our proposed methodology can uniformly cater for different discretization accuracy demands. By halving the jerk bounds in Case 6, the acceleration profile is smoother than that in Case 3 (Figs. 17 and 19). A smaller $\mathrm{da}_{\max}$ yields less agile maneuvers and leads to a 24.24% increase in t_f. Cases 7 and 8 evaluate the algorithm ability to cope with irregular scenarios. As a brief summary of this subsection, Cases 5–8 show the algorithm unification and efficiency through comparisons.

E. On the Real-Time Performance of the Maneuver Planner

A maneuver planner would not be useful if it executes slowly, in spite of the merits in accuracy and unification. This subsection analyzes the performance of our on-site initialization strategy and on-line replanning process.

Case 9 investigates the performance of on-site initialization. Suppose that the result of Case 1 is obtained off-line and then stored in a look-up table. When Case 9 is assigned, the on-site initialization method first consults the look-up table and finds

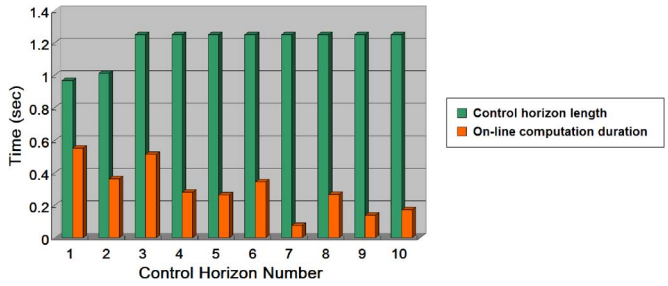

Fig. 21. Control horizon lengths and online computation duration periods in receding-horizon replanning process (Case 10).

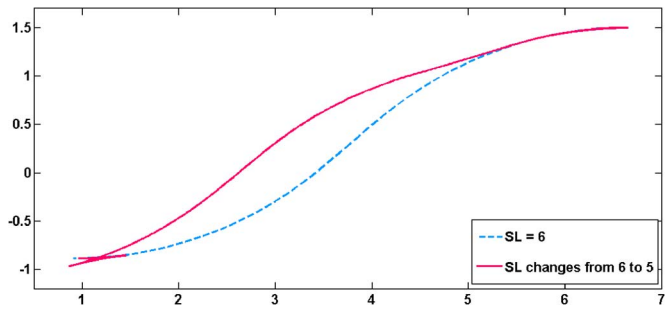

Fig. 22. Comparison between Case 1 and Case 10.

that Case 1 is the most similar case; thereafter, the recorded optimization result of Case 1 is extracted from the look-up table and utilized as initial guess in solving Case 9. As it turns out, the on-site maneuver planning of Case 9 is accomplished in less than 1 sec, which is satisfactory in real-world applications.

Case 10 investigates the on-line replanning capability. Details of the receding-horizon replanning process are shown in Fig. 21. During each control horizon (see the green long pillars in Fig. 21), the replanning computation (see the orange pillars adjacent to those green ones) takes up a small portion. As a general trend, the computational time reduces as the parking process continues, because the scales of the on-line replanning problems are gradually decreasing (see Section III-E). A comparison between the paths of Case 1 and Case 10 is shown in Fig. 22, wherein both trajectories are nearly the same at the beginning but as the vacant parking spaces are reducing in Case 10, more efforts are needed before the parking scheme is completed.

VI. Conclusion

In this paper, we have investigated how to plan time-optimal maneuvers for autonomous parallel parking. The underlying highlights lie in the following aspects.

(i) We advocate the benefits in using direct maneuver planners after providing a comprehensive review of the maneuver planning methods in robotics.
(ii) Our proposal is an open and unified dynamic optimization framework, and any user-specified constraint can be incorporated in this framework provided that it is explicitly described via equality/inequality. Moreover, our proposal contains anything but unjustifiable subjective knowledge or experiences, which would largely limit the full utilization of objective conditions.

(iii) The proposed on-site initialization via a look-up table and receding-horizon replanning process indicate this maneuver planner is promising to meet real-time demands.

In spite of the bright sides, future efforts are needed to polish the proposed receding horizon replanning architecture in NMPC, and to carry out experiments on a real-world automobile platform. Moreover, dynamic models, instead of kinematic ones, should be established to describe the movement principles of a vehicle.

ACKNOWLEDGMENT

The authors are grateful to the anonymous reviewers and Dr. Y. Jiang for their precious comments and suggestions.

REFERENCES

[1] P. Zips, M. Böck, and A. Kugi, "Optimisation based path planning for car parking in narrow environments," *Robot. Auton. Syst.*, vol. 79, pp. 1–11, May 2016.

[2] H. Vorobieva, S. Glaser, N. Minoiu-Enache, and S. Mammar, "Automatic parallel parking in tiny spots: Path planning and control," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 396–410, Feb. 2015.

[3] T. Rajabioun and P. Ioannou, "On-street and off-street parking availability prediction using multivariate spatiotemporal models," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2913–2924, Oct. 2015.

[4] B. Li and Z. Shao, "Autonomous parking: A unified motion planning framework based on simultaneous dynamic optimization," in *Proc. 34th Chin. Control Conf.*, 2015, pp. 5913–5918.

[5] J. M. Blosseville and M. Flonneau, "From traffic signal control systems to automated driving: A review of ITS systems based on a cross historical and technical perspective," in *Proc. 5th TRA Conf.*, 2014, pp. 1–7.

[6] B. Li and Z. Shao, "A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles," *Knowl.-Based Syst.*, vol. 86, pp. 11–20, Sep. 2015.

[7] M. Czubenko, Z. Kowalczuk, and A. Ordys, "Autonomous driver based on an intelligent system of decision-making," *Cogn. Comput.*, vol. 7, no. 5, pp. 569–581, Oct. 2015.

[8] C. Lee, C. Lin, and B. Shiu, "Autonomous vehicle parking using hybrid artificial intelligent approach," *J. Intell. Robot. Syst.*, vol. 56, no. 3, pp. 319–343, Jan. 2009.

[9] B. Li and Z. Shao, "Simultaneous dynamic optimization: A trajectory planning method for nonholonomic car-like robots," *Adv. Eng. Softw.*, vol. 87, pp. 30–42, Sep. 2015.

[10] B. Li, K. Wang, and Z. Shao, "Time-optimal trajectory planning for tractor-trailer vehicles via simultaneous dynamic optimization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Hamburg, Germany, Sep. 2015, pp. 3844–3849.

[11] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *Amer. J. Math.*, vol. 79, no. 3, pp. 497–516, Jul. 1957.

[12] J. Reeds and L. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pac. J. Math.*, vol. 145, no. 2, pp. 367–393, Oct. 1990.

[13] T. Fraichard and A. Scheuer, "From Reeds and Shepp's to continuous-curvature paths," *IEEE Trans. Robot.*, vol. 20, no. 6, pp. 1025–1035, Nov. 2004.

[14] X. Du and K. Tan, "Autonomous reverse parking system based on robust path generation and improved sliding mode control," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1225–1237, Jun. 2015.

[15] P. Soueres and J. P. Laumond, "Shortest paths synthesis for a car-like robot," *IEEE Trans. Autom. Control*, vol. 41, no. 5, pp. 672–688, May 1996.

[16] F. Gómez-Bravo, F. Cuesta, A. Ollero, and A. Viguria, "Continuous curvature path generation based on β-spline curves for parking manoeuvres," *Robot. Auton. Syst.*, vol. 56, no. 4, pp. 360–372, Apr. 2008.

[17] T. C. Liang, J. S. Liu, G. T. Hung, and Y. Z. Chang, "Practical and flexible path planning for car-like mobile robot using maximal-curvature cubic spiral," *Robot. Auton. Syst.*, vol. 52, no. 4, pp. 312–335, Sep. 2005.

[18] F. Lamiraux and J. P. Lammond, "Smooth motion planning for car-like vehicles," *IEEE Trans. Robot. Autom.*, vol. 17, no. 4, pp. 498–501, Aug. 2001.

[19] T. Fujii, Y. Arai, H. Asama, and I. Endo, "Multilayered reinforcement learning for complicated collision avoidance problems," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1998, vol. 3, pp. 2186–2191.

[20] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, Jun. 2011.

[21] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

[22] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *Int. J. Robot. Res.*, vol. 29, no. 5, pp. 485–501, Apr. 2010.

[23] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1994, pp. 3310–3317.

[24] Q. Wang, M. Wulfmeier, and B. Wagner, "Voronoi-based heuristic for nonholonomic search-based path planning," in *Intelligent Autonomous Systems*, vol. 13, E. Menegatti, N. Michael, K. Berns, and H. Yamaguchi, Eds. Cham, Switzerland: Springer-Verlag, 2016, pp. 445–458.

[25] Y. Zhao and E. Collins, "Robust automatic parallel parking in tight spaces via fuzzy logic," *Robot. Auton. Syst.*, vol. 51, no. 2/3, pp. 111–127, May 2005.

[26] W. N. Patten, H. C. Wu, and W. Cai, "Perfect parallel parking via Pontryagin's principle," *Trans. ASME, J. Dyn. Syst. Meas. Control*, vol. 116, no. 4, pp. 723–728, Dec. 1994.

[27] H. J. Sussmann and G. Tang, "Shortest paths for the Reeds-Shepp car: A worked out example of the use of geometric techniques in nonlinear optimal control," Rutgers Center Syst. Control, Piscataway, NJ, USA, Tech. Rep. SYCON–91-10, 1991.

[28] H. Wang, Y. Chen, and P. Souères, "A geometric algorithm to compute time-optimal trajectories for a bidirectional steered robot," *IEEE Trans. Robot.*, vol. 25, no. 2, pp. 399–413, Apr. 2009.

[29] S. Fleury, P. Soueres, J. P. Laumond, and R. Chatila, "Primitives for smoothing mobile robot trajectories," *IEEE Trans. Robot. Autom.*, vol. 11, no. 3, pp. 441–448, Jun. 1995.

[30] K. Kondak and G. Hommel, "Computation of time optimal movements for autonomous parking of non-holonomic mobile platforms," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2001, vol. 3, pp. 2698–2703.

[31] D. Maravall and J. de Lope, "Multi-objective dynamic optimization with genetic algorithms for automatic parking," *Soft. Comput.*, vol. 11, no. 3, pp. 249–257, Feb. 2007.

[32] Y. Kim and B. K. Kim, "Efficient time-optimal two-corner trajectory planning algorithm for differential-driven wheeled mobile robots with bounded motor control inputs," *Robot. Auton. Syst.*, vol. 64, pp. 35–43, Feb. 2015.

[33] M. Renaud and J. Fourquet, "Minimum time motion of a mobile robot with two independent, acceleration-driven wheels," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1997, vol. 3, pp. 2608–2613.

[34] D. J. Balkcom and M. T. Mason, "Time optimal trajectories for bounded velocity differential drive vehicles," *Int. J. Robot. Res.*, vol. 21, no. 3, pp. 199–217, Mar. 2002.

[35] R. M. Murray and S. S. Sastry, "Steering nonholonomic systems using sinusoids," in *Proc. 29th IEEE Conf. Decision Control*, 1990, pp. 2097–2101.

[36] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, "Path planning and trajectory planning algorithms: A general overview," in *In Motion and Operation Planning of Robotic Systems: Background and Practical Approaches*, vol. 29, G. Carbone and F. Gomez-Bravo, Eds. Cham, Switzerland: Springer-Verlag, 2015, pp. 3–27.

[37] S. Macfarlane and E. A. Croft, "Jerk-bounded manipulator trajectory planning: Design for real-time applications," *IEEE Trans. Robot. Autom.*, vol. 19, no. 1, pp. 42–52, Feb. 2003.

[38] A. Scheuer and C. Laugier, "Planning sub-optimal and continuous-curvature paths for car-like robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1998, vol. 1, pp. 25–31.

[39] L. T. Biegler, *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*, vol. 10. Philadelphia, PA, USA: *SIAM*, 2010.

[40] A. Wächter and L. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Programm.*, vol. 106, no. 1, pp. 25–57, Mar. 2006.

[41] B. H. Li and C. T. Chang, "A simple and efficient initialization strategy for optimizing water-using network designs," *Ind. Eng. Chem. Res.*, vol. 46, no. 25, pp. 8781–8786, Dec. 2007.

[42] B. Li and Z. Shao, "An incremental strategy for tractor-trailer vehicle global trajectory optimization in the presence of obstacles," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Dec. 2015, pp. 1447–1452.

[43] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*. San Francisco, CA, USA: Scientific, 2003.

[44] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Trans. Autom. Control*, vol. AC-30, no. 6, pp. 531–541, Jun. 1985.
[45] H. Vorobieva, N. Minoiu-Enache, S. Glaser, and S. Mammar, "Geometric continuous-curvature path planning for automatic parallel parking," in *Proc. IEEE ICNSC*, 2013, pp. 418–423.
[46] H. Vorobieva, S. Glaser, N. Minoiu-Enache, and S. Mammar, "Automatic parallel parking with geometric continuous-curvature path planning," in *Proc. IEEE Intell. Veh. Symp.*, 2014, pp. 465–471.

Bai Li received the B.S. degree from Beihang University (formerly Beijing University of Aeronautics and Astronautics), Beijing, China, in 2013. He is currently working toward the Ph.D. degree with the College of Control Science and Engineering, Zhejiang University, Hangzhou, China.

His research interests include computational planning and control methodologies with the autonomous driving backdrop.

Kexin Wang received the M.S. degree from Shandong University, Jinan, China, in 2004 and the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2008.

She is currently with the College of Control Science and Engineering, Zhejiang University. Her research interests include theories and applications of dynamic optimization and nonlinear programming.

Zhijiang Shao received the B.S. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 1992 and 1997, respectively. He was a Lecturer from 1997 to 1999 and an Associate Professor from 1999 to 2004 with Zhejiang University, where he has been a Professor since 2004. He is currently with State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University. His research interests include large-scale dynamic optimization, real-time optimization, and supervisory control.