Knowledge-Based Systems 86 (2015) 11–20

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles

Bai Li [a], Zhijiang Shao [a,b,*]

[a] Department of Control Science & Engineering, Zhejiang University, Hangzhou 310027, PR China
[b] State Key Laboratory of Industrial Control Technology, Hangzhou, PR China

ARTICLE INFO

Article history:
Received 17 January 2015
Received in revised form 20 March 2015
Accepted 19 April 2015
Available online 29 April 2015

Keywords:
Autonomous vehicle
Motion planning
Time optimal control
Simultaneous approach
Dynamic optimization

ABSTRACT

This paper proposes a motion planner for autonomous parking. Compared to the prevailing and emerging studies that handle specific or regular parking scenarios only, our method describes various kinds of parking cases in a unified way regardless they are regular parking scenarios (e.g., parallel, perpendicular or echelon parking cases) or not. First, we formulate a time-optimal dynamic optimization problem with vehicle kinematics, collision-avoidance conditions and mechanical constraints strictly described. Thereafter, an interior-point simultaneous approach is introduced to solve that formulated dynamic optimization problem. Simulation results validate that our proposed motion planning method can tackle general parking scenarios. The tested parking scenarios in this paper can be regarded as benchmark cases to evaluate the efficiency of methods that may emerge in the future. Our established dynamic optimization problem is an open and unified framework, where other complicated user-specific constraints/optimization criteria can be handled without additional difficulty, provided that they are expressed through inequalities/polynomial explicitly. This proposed motion planner may be suitable for the next-generation intelligent parking-garage system.

1. Introduction

Autonomous vehicles (sometimes called self-driving cars or driverless cars) refer to robotic vehicles that travel between destinations without human operators [1]. Such vehicles are expected to bring a variety of benefits, e.g., improving road network capacity and freeing up driver-occupants' time [2]. One industry analyst firm, Navigant Research, predicted that 75% of the vehicles sold in 2035 will have some sort of autonomous capability [3]. Although fully autonomous vehicles will not travel on the streets in the near future (because of the lack of legislation and mature technologies), yet the commercial availability of local vehicular automation systems (i.e., driver assistance systems and semi-autonomous systems) is increasing [4].

Autonomous parking is a critical application of driver assistance technologies. Relevant products have been designed by car manufacturers such as Audi, BMW, Ford, Land Rover, Mercedes-Benz, Nissan, and Toyota [5]. Nevertheless, these products are challenged in terms of thoroughly easing parking burdens. For instance, recognizing the environment during heavy rainstorms, inducing smart maneuvers to park in a narrow spot or grasping user preferences remains to be difficult issues [6,7]. In this sense, autonomous parking technologies deserve further investigation.

A successful autonomous parking process involves three sequential procedures: circumstance recognition, open-loop motion planning and closed-loop control execution [8]. Among these three procedures, motion planning alone is responsible for decision-making. In other words, the motion planning procedure largely determines how intelligent the entire parking system will be. Therefore, it is necessary to develop a reliable method in the motion planning phase.

Motion planning research studies in autonomous parking originated with [9], which systematically formulated a generalized autonomous parking problem for the first time. Ref. [10] categorized the prevailing motion planning algorithms into two types that are respectively applied in environments with complete or incomplete knowledge. Although many studies focus on motion planning in environments with incomplete knowledge [11], we believe that methods based on complete knowledge of the environment are not fully mature (the reasons will be presented later). This current study is based on an assumption that knowledge of the environment should be completely available before the motion planning procedure is implemented.

* Corresponding author at: Department of Control Science & Engineering, Zhejiang University, Hangzhou 310027, PR China.
E-mail addresses: libai@zju.edu.cn (B. Li), szj@zju.edu.cn (Z. Shao).

http://dx.doi.org/10.1016/j.knosys.2015.04.016
0950-7051/

The prevailing motion planning methods on the basis of complete environmental knowledge can be broadly classified into three categories: geometric-based methods, heuristic-based methods, and methods based on control theories. Geometric-based approaches commonly compute reference paths first and then generate trajectories following the obtained paths (e.g., [12–16]). Here, a path refers to a geometric curve $y = f(x)$ in the xy coordinate frame, whereas a trajectory attaches the time course along a path, i.e., the determination of $x = x(t)$ [17]. Heuristic-based methods usually seek solutions from artificial intelligence techniques, e.g., fuzzy logics [18,19], search-based methods [20,21], random sampling methods [22] and machine learning methods [23]. Commonly the heuristic methods determine merely paths rather than trajectories, thus additional efforts must be exerted to convert the computed paths into trajectories. References regarding control theories are relatively scarce [24–26]. Such analytical methods usually deal with specific cases only, lacking generalization abilities [21]. Most of the previous publications mentioned above have validated their concerned methods effective through simulations, and some of those methods have even been executed on real robots in the field (e.g., [18,19]). In spite of their success, three issues stil deserve consideration. First, many existing methods do not solve the motion control problem directly. Typically, those heuristic-based path planning methods suffer from this limitation because kinematic descriptions of the vehicle are either missing or incomplete (e.g., [15,16,19–21]). In fact, quite few works have formulated complete kinematics (e.g., [27]). Second, it is better to generate optimal/optimized motions (based on some predefined criteria) rather than generate merely feasible motions. Third, we notice that a parking spot has been assumed as a slot region (see Fig. 1(a)) in most of the previous publications. The requirement that a car should not collide with the shaded regions in Fig. 1(a) is impractical. In fact, we only need the car terminally stay inside a rectangular parking spot. That is to say, the car can temporarily "invade" a neighboring spot region during its parking maneuvers provided that no collision happens. On the other hand, even when one is reluctant to invade temporarily into others' parking regions, he may find his target parking spot partly occupied by a parked car.

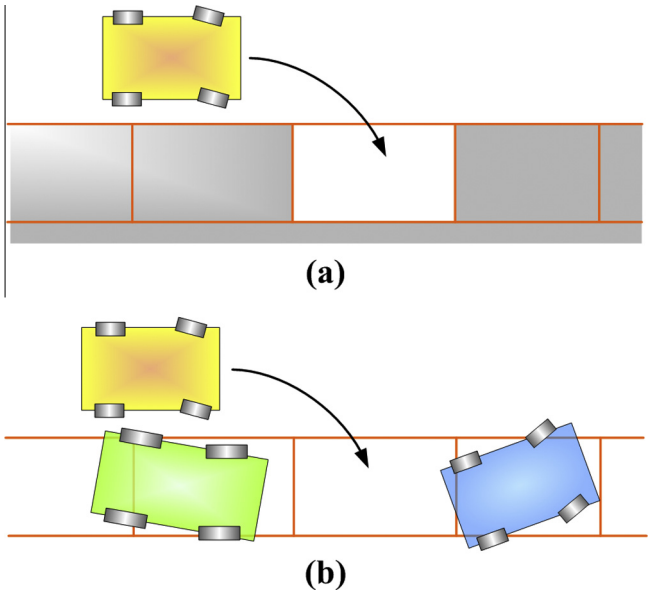

Fig. 1. Schematic of regular and irregular parking scenarios: (a) collision-free requirements in previous studies where a car should not hit the shaded regions during its parking maneuvers and (b) collision-free requirements considered in this current study, where a vehicle only needs to avoid colliding with neighboring cars during its parking maneuvers.

Such parking scenarios (see Fig. 1(b)) are irregular but indeed ordinary in our daily life. Research studies that considered general parking scenarios are scarce. Apart from Paromtchik & Laugier's three publications in the early years (i.e., [28–30]), no other reverent studies can be found, to the best of our knowledge. As a brief summary, no study has solved or can solve the aforementioned three issues altogether.

This work aims to address the original motion planning problem directly. To this end, differential equations are formulated to describe the vehicle kinematics and geometric analyses are conducted to strictly constrain the vehicle from hitting surrounding cars regardless they are regularly parked or not. We pursue for the time-optimal motions, thus formulating an optimal control problem (also can be regarded as a dynamic optimization problem) which is identical to the original parking motion planning scheme. A simultaneous approach based on interior point method (IPM) is applied to solve the formulated dynamic optimization problem.

The rest of this paper is organized as follows. In Section 2, the kinematics of an autonomous vehicle and the collision-free requirements are presented so as to formulate a dynamic optimization problem. In Section 3, the IPM-based simultaneous approach is introduced. In Section 4, simulations on several parking scenarios are presented, followed by Section 5, where detailed analyses on the simulation results are provided. Finally in Section 6, our conclusions are drawn.

2. Dynamic optimization problem formulation

This section formulates a dynamic optimization problem on the basis of the original parking motion planning mission. Detailedly, the vehicle kinematics, mechanical constraints and collision-free constraints will be introduced respectively. At the end of this section, we will show the overall formulation.

2.1. Kinematics of a car-like vehicle

The kinematics of a concerned front-steering autonomous vehicle can be expressed by

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

where $t \in [0, t_f]$ refers to time, t_f indicates the terminal moment of the entire dynamic process, (x, y) refers to the mid-point of the front

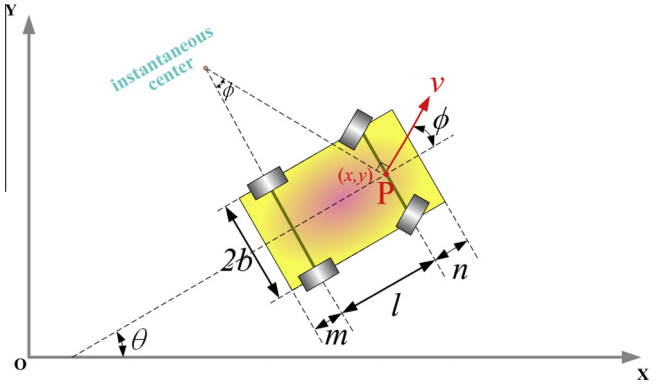

Fig. 2. Parametric notations related to vehicle size and kinematics.

wheel axis (see the reference point P in Fig. 2), θ refers to the orientation angle, v refers to the linear velocity of point P, a refers to the corresponding acceleration, ϕ refers to the steering angle of front wheels and ω refers to the corresponding angular velocity. Moreover, l denotes the wheelbase length, n denotes the front overhang length, m denotes the rear overhang length and $2b$ denotes the car width.

Here, $\omega(t)$ and $a(t)$ are selected as control variables and the remaining five variables (i.e., $x(t), y(t), v(t), \theta(t)$ and $\phi(t)$) are regarded as state variables. Given their initial values, the state variables can be determined successfully one after another through integral once $\omega(t)$ and $a(t)$ are known.

2.2. Mechanical and physical constraints

Aside from the vehicle kinematics described through differential equations in the preceding subsection, the bounded constraints on state/control variables should be considered as well. In detail, we express these mechanical/physical constraints as

$$\begin{cases} |a(t)| \leqslant \mathrm{a}_{\max} \\ |v(t)| \leqslant \mathrm{v}_{\max} \\ |\phi(t)| \leqslant \Phi_{\max} \\ |\omega(t)| \leqslant \mathrm{w}_{\max} \end{cases}, \quad \forall t \in [0, t_f]. \tag{2}$$

The reasons for the imposition of boundaries on $a(t), v(t)$ and $\phi(t)$ are obvious. Imposing bounds on $\omega(t)$ has been widely applied with the aim of planning continuous-curvature trajectories in a number of previous publications (e.g., [12,15,16,27]). The rationale behind this issue is that, the instantaneous curvature $\kappa(t) = \frac{\sin\phi(t)}{l}$ as well as its derivative $\frac{\mathrm{d}\kappa(t)}{\mathrm{d}t} = \frac{\omega(t)\cdot\cos\phi(t)}{l}$ should be bounded so as to avoid generating non-smooth trajectories. Commonly, non-smooth trajectories are not recommended due to the resulting undesirable wear of tires [12]. Given that $\phi(t)$ is mechanically limited, if there is no boundaries imposed on $\omega(t)$, one cannot guarantee that $\frac{\mathrm{d}\kappa(t)}{\mathrm{d}t}$ is bounded, and then the continuous-curvature property cannot be guaranteed.

Till now, the mechanical and physical constraints associated with the vehicle have been formulated. Besides that, there are also collision-avoidance conditions that should be satisfied when an autonomous vehicle moves in the environment, which is introduced in the next subsection.

2.3. Collision-free restrictions in the environment

This subsection concerns the environmental collision-avoidance descriptions through strict geometrics. Unlike previous works that assume the obstacles form ideal slots, we merely require the to-be-parked car to avoid colliding with other parked cars during its maneuvers. Assuming that cars are rectangular, this subsection first introduces how to precisely describe one rectangle locates outside the other. Then, the collision-free constraints are formulated.

All of the scenarios in which one rectangle collides with another can be divided into two categories. One refers to the cases in which at least one edge point of a rectangle is located within the region of the other rectangle (see Fig. 3(a)). The other case refers to the possibility that no such edge point is involved (see Fig. 3(b)). Nonetheless, the second possibility always originates from the first one. Therefore, if we require that none of the four edge points on one rectangle remaining within the other rectangular region during the entire dynamic process, we can guarantee that the two rectangles do not collide. Here, a question that would arise is, how to determine whether or not each edge point is located inside

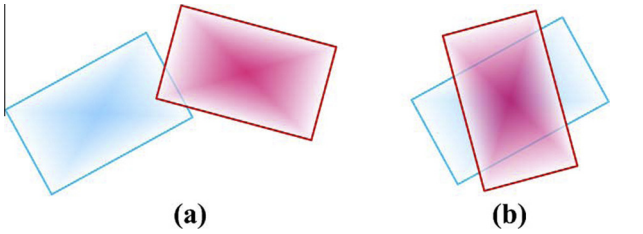

Fig. 3. Schematic of two possibilities in which one rectangle collides with the other.

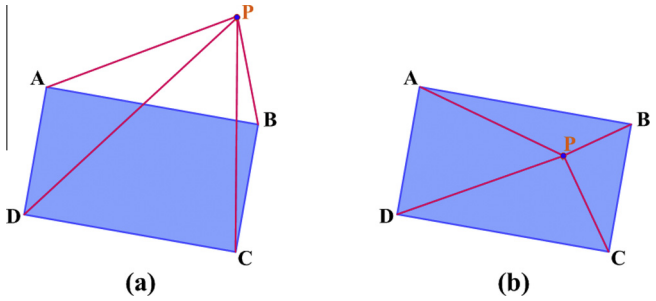

Fig. 4. Schematic of a "triangle area criterion", which judges whether or not point P is located within the rectangle region ABCD.

a given rectangle region? The following formula addresses this issue when Fig. 4 is taken as an example:

$$S_{\Delta PAB} + S_{\Delta PBC} + S_{\Delta PCD} + S_{\Delta PDA} > S_{\square ABCD}, \tag{3}$$

where S_Δ denotes the triangle area and $S_\square$ denotes the rectangle area. This judgment can be proven analytically (through simple mathematical knowledge), but we omit this part so as to avoid losing focus of this paper.

Collision-free constraints can be formulated on the basis of the triangle area judgment mentioned above. In detail, when we expect two rectangles not collide, all the four edge points on one rectangle should remain outside the other rectangular region. Therefore, if there are N_{car} parked cars in the environment, there will be as many as $8N_{car}$ inequalities that should be satisfied during the entire dynamic parking process so as to avoid collision.

To briefly summarize here, compared to the prevailing methods that can only deal with specific cases (e.g., [13,15,16,18,20,21,26,27]), our formulated model describes various kinds of cases in a unified way regardless they are regular parking scenarios (i.e., parallel, perpendicular or echelon parking cases) or irregular ones.

2.4. Terminal conditions

In addition to the kinematics and constraints, there are terminal conditions that should be met at the moment $t = t_f$.

First, a parking process should end with a full stop, that is, $v(t_f) = 0$.

Second, with regard to terminal location, we require a car to park itself within a given region. According to the box region depicted in Fig. 5, we define the following:

$$\begin{cases} A_x(t_f) \in [x_0, x_0 + BL] \\ B_x(t_f) \in [x_0, x_0 + BL] \\ C_x(t_f) \in [x_0, x_0 + BL] \\ D_x(t_f) \in [x_0, x_0 + BL] \end{cases} \text{and} \begin{cases} A_y(t_f) \in [y_0, y_0 + BW] \\ B_y(t_f) \in [y_0, y_0 + BW] \\ C_y(t_f) \in [y_0, y_0 + BW] \\ D_y(t_f) \in [y_0, y_0 + BW] \end{cases}, \tag{4}$$

where

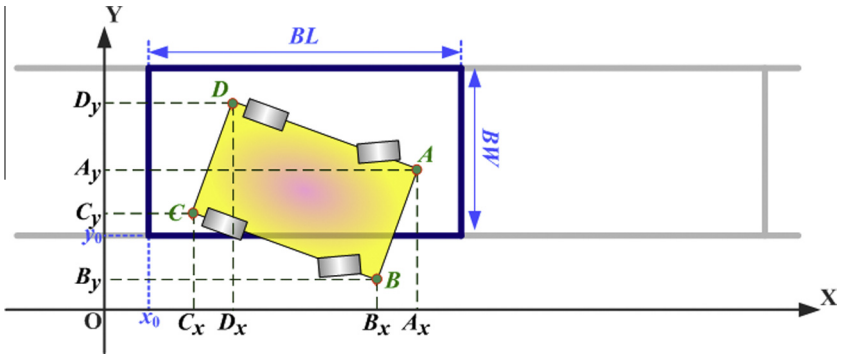

Fig. 5. Parametric notations related to size and location of the box region where the vehicle is terminally parked.

$$\begin{cases} A_x = x + n\cdot\cos\theta - b\cdot\sin\theta \\ A_y = y + n\cdot\sin\theta + b\cdot\cos\theta \\ B_x = x + n\cdot\cos\theta + b\cdot\sin\theta \\ B_y = y + n\cdot\sin\theta - b\cdot\cos\theta \\ C_x = x - (m+l)\cdot\cos\theta + b\cdot\sin\theta \\ C_y = y - (m+l)\cdot\sin\theta - b\cdot\cos\theta \\ D_x = x - (m+l)\cdot\cos\theta - b\cdot\sin\theta \\ D_y = y - (m+l)\cdot\sin\theta + b\cdot\cos\theta \end{cases} \quad (5)$$

Third, compared to some studies that additionally required $\theta(t_f) = 0$ and/or $\phi(t_f) = 0$, our formulation does not contain such terminal conditions. At this point, we believe that once the concerned vehicle fully stops inside a desired parking spot, it needs not worry about other issues. In fact, it is unrealistic to park a car exactly in line with the spot frontiers or to let the terminal steering angle be zero.

2.5. Overall dynamic optimization formulation

In this work, we aim to find the motions associated with minimum t_f. In other words, min-time control motions are expected. Our dynamic optimization formulation is not a specific optimization model but a unified optimization framework, which can cater for different optimization demands/conditions/constraints. Fig. 6 schematically illustrates our formulated optimization framework. The unification of that framework lies in the following few aspects: (i) it does not make any distinction between regular and irregular parking scenarios; (ii) it utilizes complete kinematics (i.e., Eq. (1)) to provide sufficient and necessary principles about how a concerned vehicle moves; (iii) extra user-specific conditions/requirements can be considered in our framework provided that they can be explicitly described through algebraic equalities/inequalities; and (iv) although we pursue for time-optimal motions, this framework is capable of considering other optimization criterions with ease.

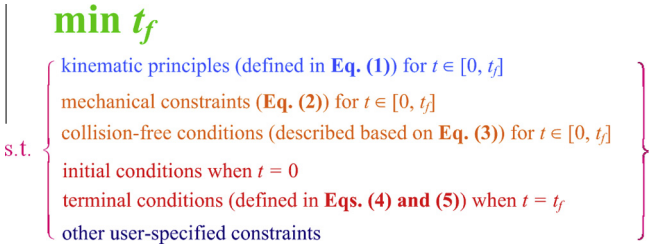

Fig. 6. Schematic of our formulated unified dynamic optimization framework.

Till now we have formulated an optimization framework for the parking motion planning scheme. It is far beyond the analytical methods' ability to solve such a complicated dynamic optimization problem in general, thus the adoption of numerical methods may be the only way to provide solutions. In the next section, the IPM-based simultaneous approach we utilize to solve the formulated dynamic optimization problem will be introduced.

3. IPM-based simultaneous approach for dynamic optimization

In this section, the IPM-based simultaneous approach is presented to solve the dynamic optimization problem formulated in the preceding section.

The IPM-based simultaneous approach consists of two phases, namely the discretization phase and programming phase. First, in the discretization phase, all the state and control profiles in time are discretized through the collocation of finite elements. In this way, the original dynamic optimization problem is transformed into a nonlinear programming (NLP) formulation. Thereafter, the resulting NLP problem is solved in the second phase. The simultaneous method utilized in the discretization phase is equivalent to a fully implicit Runge–Kutta method with high order accuracy and excellent stability. It possesses various merits over its competitors when tackling dynamic optimization problems, especially ones with complicated constraints and with input instabilities. Interested readers may consult [31] for a detailed review of the simultaneous method. On the other hand, the resulting NLP problem is usually in large scale (because an infinite-dimensional dynamic optimization problem has been converted into a finite-dimension programming problem in the first phase), thus a highly efficient NLP solver is needed in the second phase. IPM is such an efficacious large-scale NLP solver, which applies a logarithmic barrier strategy to handle inequality constraints, solves a set of equality constrained optimization problems for a monotonically decreasing sequence of the barrier parameter, and quickly converges to the solution of the given NLP. In fact, IPM is capable of solving an NLP with up to several million variables, constraints and degrees of freedom [32]. The remainder of this section is organized as follows. First, we present the principles of the two phases respectively. Then, we focus on how this approach can be utilized to solve the unified dynamic optimization problem we have formulated.

3.1. Discretization phase

Without loss of generality, we consider the following general dynamic optimization problem [31]:

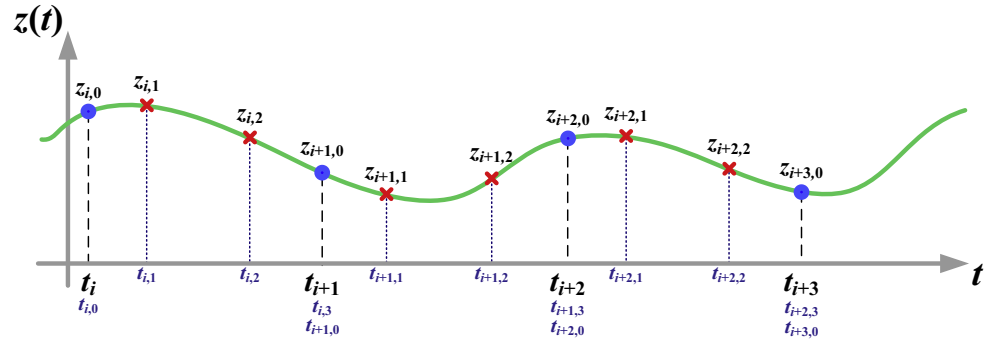

Fig. 7. Collocation of finite elements and interpolation points for differential state variables.

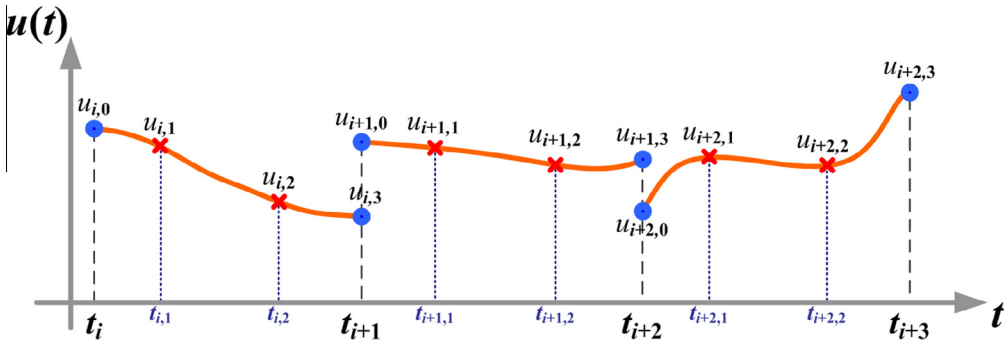

Fig. 8. Collocation of finite elements and interpolation points for algebraic state variables or control variables.

$$\min \varphi(z(t_f)),\quad s.t.\begin{cases}\frac{dz(t)}{dt}=F(z(t),y(t),u(t))\\ G(z(t),y(t),u(t))\leqslant 0\\ z(0)=z_0\\ z(t_f)=z_{t_f}\\ t\in[0,t_f]\end{cases}, \tag{6}$$

where $z(t)$ refers to differential state variables, $y(t)$ temporarily denotes the algebraic state variables in this subsection and $u(t)$ denotes the control variables. The unknowns in the concerned optimization problem (i.e., Eq. (6)) include $z(t), y(t)$ and $u(t)$. They are functions of the scalar time parameter t.

First, the time domain $[0, t_f]$ is divided into Nfe finite intervals $\{[t_{i-1}, t_i] \mid i = 1, 2, \ldots, Nfe\}$, where $t_0 = 0$ and $t_{Nfe} = t_f$. Then the duration of each element can be written as $h_i = t_i - t_{i-1}$, $i = 1, 2, \ldots, Nfe$. This work considers equidistant division in $[0, t_f]$, thus the duration h of every interval equals t_f/Nfe. For consistency with the previous studies in the area of computational science, we refer to such intervals as "finite elements".

Second, we choose $(K+1)$ interpolation points in the ith element and approximate the state using the following Lagrange polynomial:

$$z(t) = \sum_{j=0}^{K}\left(z_{ij}\cdot\prod_{k=0,\neq j}^{K}\frac{(\tau-\tau_k)}{(\tau_j-\tau_k)}\right), \tag{7}$$

where $t = t_{i-1} + h\cdot\tau, \tau\in[0,1], \tau_0 = 0$ and $0 < \tau_i \leqslant 1$ $(j = 1, 2, \ldots, K)$. Each τ_i refers to either Gauss or Radau points, which can be calculated off-line according to Gaussian Quadrature Accuracy Theorem when K is determined. Interested readers can consult [33] for the computation of τ_i.

As illustrated in Fig. 7, the Lagrange polynomial representation in Eq. (7) possesses a desirable property that $z(t_{ij}) = z_{ij}$, where $t_{ij} = t_{i-1} + h\cdot\tau_j$. In other words, the polynomial in Eq. (7) passes the $(K+1)$ interpolation points directly once they are determined on the ith finite element. This merit largely simplifies the optimization procedure in the next phase.

In addition, since $z(t)$ refer to states to be differentiated, continuity of the differential state variables across element boundaries should be guaranteed; otherwise, the derivative will not exist. Thus we have

$$z_{i+1,0} = \sum_{j=0}^{K}\left(z_{ij}\cdot\prod_{k=0,\neq j}^{K}\frac{(1-\tau_k)}{(\tau_j-\tau_k)}\right),\quad i = 1, 2, \ldots, Nfe-1, \tag{8a}$$

$$z_{t_f} = \sum_{j=0}^{K}\left(z_{Nfe,j}\cdot\prod_{k=0,\neq j}^{K}\frac{(1-\tau_k)}{(\tau_j-\tau_k)}\right), \tag{8b}$$

$$z_0 = z_{1,0}. \tag{8c}$$

Similarly, the control variables $u(t)$ and algebraic states $y(t)$ can be represented by Lagrange interpolation profiles at K collocation points:

$$u(t) = \sum_{j=1}^{K}\left(u_{ij}\cdot\prod_{k=1,\neq j}^{K}\frac{(\tau-\tau_k)}{(\tau_j-\tau_k)}\right), \tag{9}$$

and

$$y(t) = \sum_{j=1}^{K}\left(y_{ij}\cdot\prod_{k=1,\neq j}^{K}\frac{(\tau-\tau_k)}{(\tau_j-\tau_k)}\right). \tag{10}$$

Unlike (8a) for differential states $z(t)$, here $u(t)$ and $y(t)$ can be discontinuous at finite element boundaries in Eq. (9) or (10). In other words, we do not enforce $u_{i+1,0} = u_{i,K}$ or $y_{i+1,0} = y_{i,K}$ $(i = 1, 2, \ldots, Nfe-1)$ as depicted in Fig. 8.

By substituting Eqs. (8)–(10) into Eq. (6), we yield:

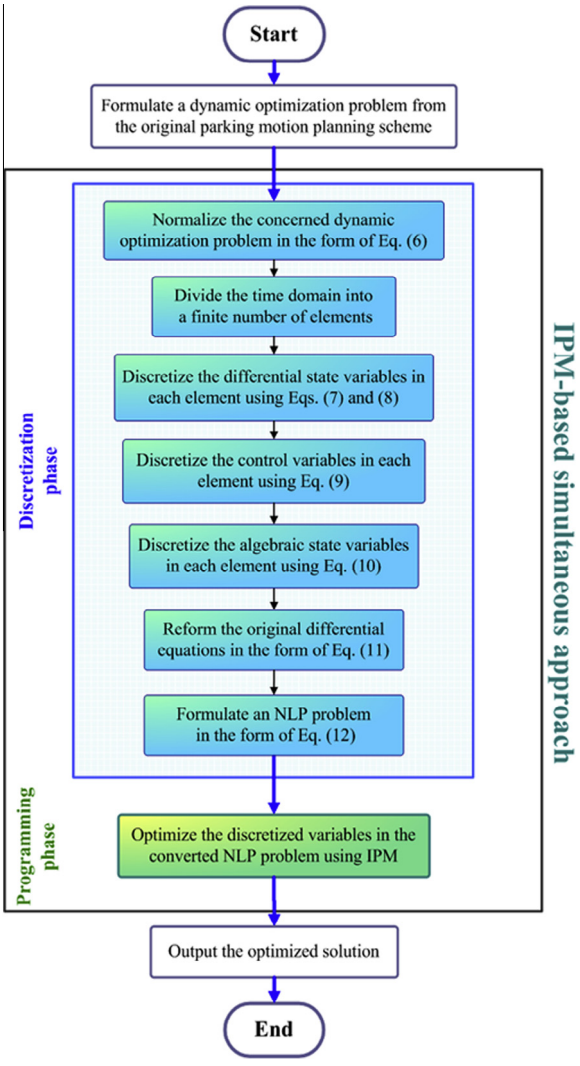

Fig. 9. Flowchart of the IPM-based simultaneous approach for dynamic optimization problems.

Table 1
User-specific parametric settings.

Parameter	Description	Setting
ε_{tol}	Convergence tolerance in IPM	10^{-12}
κ_ε	Minimum absolute distance from the initial point to bound in IPM	10^{-4}
$K+1$	Interpolation point number in the simultaneous approach	4
Nfe	Number of finite elements in the simultaneous approach	20
BL	Terminal parking box region length	–
BW	Terminal parking box region width	–
n	Front overhang length	0.960
l	Distance between the front and back wheel axes	2.800
m	Rear overhang length	0.929
$2b$	Car width	1.942

$$\sum_{j=0}^{K}\left(\left.\frac{\mathrm{d}\varsigma_j(\tau)}{\mathrm{d}\tau}\right|_{\tau=\tau_k}\cdot z_{ik}\right)-h\cdot F(z_{ik},y_{ik},u_{ik})=0,\quad i=1,2,\ldots,Nfe,k=1,2,\ldots,K, \tag{11}$$

where $\varsigma_j(\tau)=\prod_{k=0,\neq j}^{K}\frac{(\tau-\tau_k)}{(\tau_j-\tau_k)}$.

Given a fixed number of elements, the NLP formulation originated from the original dynamic optimization problem (i.e., Eq. (6)) can be written in the form of Eq. (12):

$$\min_{z_{ij},y_{ij},u_{ij}}\phi(z(t_f)),$$
$$\text{s.t.}\begin{cases}\sum_{k=0}^{K}\left(\left.\frac{\mathrm{d}\varsigma_j(\tau)}{\mathrm{d}\tau}\right|_{\tau=\tau_k}\cdot z_{ik}\right)-h_i\cdot F(z_{ij},y_{ij},u_{ij})=0\\ G(z_{ij},y_{ij},u_{ij})\leqslant 0\\ z_{i_1+1,0}=\sum_{j=0}^{K}\left(z_{i_1j}\cdot\prod_{k=0,\neq j}^{K}\frac{(1-\tau_k)}{(\tau_j-\tau_k)}\right)\\ z_{1,0}=z_0\\ z_{Nfe,K}=z_{t_f}\end{cases},\quad i=1,2,\ldots,Nfe,i_1=1,2,\ldots,Nfe-1,j=1,2,\ldots,K. \tag{12}$$

Through this, the original dynamic optimization problem is converted into an NLP formulation in the discretization phase.

3.2. Programming phase

In this phase, IPM is adopted to optimize the discretized variables z_{ij},y_{ij} and u_{ij} $(i=1,2,\ldots,Nfe,j=0,2,\ldots,K)$ in the converted NLP formulation (i.e., Eq. (12)). The details of IPM are not discussed here for reasons of length and the focus of our paper. Interested readers should refer to [32].

3.3. Overall approach for dynamic optimization

The preceding two subsections present the two phases in the IPM-based simultaneous approach. This subsection mainly focuses on how this IPM-based simultaneous approach is used to solve the unified dynamic optimization problem we have formulated in Section 2.

It is worthwhile to notice that our formulated dynamic optimization problem (see Fig. 6) can be directly expressed in the form of Eq. (6). First, now that the time-optimal parking motions are pursued, the minimization criterion $\varphi(z(t_f))$ refers to t_f in our scheme. Second, $z(t)$ represents all the differential state variables in Eq. (1) (i.e., $x(t),y(t),v(t),\theta(t)$ and $\phi(t)$ in Section 2). Third, the variable $y(t)$ emerges in Section 3 stands for all the state variables that are not differentiated (i.e., $A_x(t),B_x(t),C_x(t)$, etc.). Fourth, $u(t)$ represents the control variables (i.e., $\omega(t)$ and $a(t)$). Fifth, $G(z(t),y(t),u(t))\leqslant 0$ in Eq. (6) can cover those inequalities in Section 2. At this point, it should be noted that equalities are special cases of inequalities (because any an equality $g_1=0$ can be converted into a combination of two inequalities: $g_1\geqslant 0$ and $g_1\leqslant 0$). Sixth, Eq. (6) also contains two-point boundary conditions that should be met (e.g., $v(t_f)=0$).

Through the analyses mentioned above, we manage to fit the adopted method into our formulated model. At the end of this whole section, an overall flowchart of the IPM-based simultaneous approach is illustrated in Fig. 9.

4. Simulation results

This section presents the results of simulations that were conducted in "A Mathematical Programming Language" (AMPL) environment [34] and executed on an Intel Core 2 Duo CPU with 2 GB RAM that runs at 2.53 GHz under Microsoft Windows XP. We employed version 3.11.9 of IPOPT (a commercial software package of IPM) [32]. The notations and settings of critical parameters are listed in Table 1.

In this study, we designed four scenarios to reflect the unified capability of our formulated model (as well as the optimization

Table 2
Details of four parking motion planning cases.

Case no.	Two-point boundary conditions	Constraints ($\forall t \in [0, t_f]$)	Parametric settings[a]
1	$\begin{cases} x(0) = -10 \\ y(0) = 3 \\ \theta(0) = 0 \\ v(0) = 0 \end{cases}$, $v(t_f) = 0$ and inside-box terminal condition	$\begin{cases} -2 \leqslant a(t) \leqslant 1.5 \\ \lvert v(t) \rvert \leqslant 2 \\ \lvert \phi(t) \rvert \leqslant 0.714 \\ \lvert w(t) \rvert \leqslant 1 \end{cases}$ and collision-free requirements	$\begin{cases} P_{1Ax} = -3.390, P_{1Ay} = 0.601 \\ P_{1Bx} = -3.577, P_{1By} = -1.094 \\ P_{1Cx} = -8.100, P_{1Cy} = -0.594 \\ P_{1Dx} = -7.913, P_{1Dy} = 1.100 \end{cases}$, $\begin{cases} P_{2Ax} = 7.156, P_{2Ay} = 1.193 \\ P_{2Bx} = 7.576, P_{2By} = -0.646 \\ P_{2Cx} = 2.844, P_{2Cy} = -1.727 \\ P_{2Dx} = 2.424, P_{2Dy} = 0.111 \end{cases}$ and $\begin{cases} x_0 = -3 \\ y_0 = -1.25 \\ BL = 6 \\ BW = 2.5 \end{cases}$
2	$\begin{cases} x(0) = -10 \\ y(0) = 3 \\ \theta(0) = 0 \\ v(0) = 0 \end{cases}$, $v(t_f) = 0$ and inside-slot terminal requirement	$\begin{cases} -2 \leqslant a(t) \leqslant 1.5 \\ \lvert v(t) \rvert \leqslant 2 \\ \lvert \phi(t) \rvert \leqslant 0.714 \\ \lvert w(t) \rvert \leqslant 1 \end{cases}$ and collision-free requirements	$\begin{cases} P_{1Ax} = -5.998, P_{1Ay} = -1.462 \\ P_{1Bx} = -7.431, P_{1By} = -2.385 \\ P_{1Cx} = -9.894, P_{1Cy} = 1.441 \\ P_{1Dx} = -8.461, P_{1Dy} = 2.364 \end{cases}$, $\begin{cases} P_{2Ax} = 7.492, P_{2Ay} = 1.034 \\ P_{2Bx} = 7.348, P_{2By} = -0.846 \\ P_{2Cx} = 2.508, P_{2Cy} = -0.474 \\ P_{2Dx} = 2.652, P_{2Dy} = 1.406 \end{cases}$ and $\begin{cases} x_0 = -3 \\ y_0 = -1.25 \\ BL = 6 \\ BW = 2.5 \end{cases}$
3	$\begin{cases} x(0) = -10 \\ y(0) = 6 \\ \theta(0) = 0 \\ v(0) = 0 \end{cases}$, $v(t_f) = 0$ and inside-slot terminal requirement	$\begin{cases} -2 \leqslant a(t) \leqslant 1.5 \\ \lvert v(t) \rvert \leqslant 2 \\ \lvert \phi(t) \rvert \leqslant 0.714 \\ \lvert w(t) \rvert \leqslant 1 \end{cases}$ and collision-free requirements	$\begin{cases} P_{1Ax} = -1.944, P_{1Ay} = -2.353 \\ P_{1Bx} = -3.650, P_{1By} = -2.355 \\ P_{1Cx} = -3.656, P_{1Cy} = 2.195 \\ P_{1Dx} = -1.951, P_{1Dy} = 2.197 \end{cases}$, $\begin{cases} P_{2Ax} = 2.175, P_{2Ay} = -2.585 \\ P_{2Bx} = 0.375, P_{2By} = -2.021 \\ P_{2Cx} = 1.825, P_{2Cy} = 2.611 \\ P_{2Dx} = 3.625, P_{2Dy} = 2.047 \end{cases}$ and $\begin{cases} x_0 = -3 \\ y_0 = -2.75 \\ BL = 6 \\ BW = 5.5 \end{cases}$
4	$\begin{cases} x(0) = 8 \\ y(0) = 6 \\ \theta(0) = -0.2 \\ v(0) = 0 \end{cases}$, $v(t_f) = 0$ and inside-slot terminal requirement	$\begin{cases} -2 \leqslant a(t) \leqslant 1.5 \\ \lvert v(t) \rvert \leqslant 2 \\ \lvert \phi(t) \rvert \leqslant 0.714 \\ \lvert w(t) \rvert \leqslant 1 \end{cases}$ and collision-free requirements	$\begin{cases} P_{1Ax} = -4.750, P_{1Ay} = 0.917 \\ P_{1Bx} = -4.702, P_{1By} = -0.787 \\ P_{1Cx} = -9.250, P_{1Cy} = -0.917 \\ P_{1Dx} = -9.298, P_{1Dy} = 0.787 \end{cases}$ $\begin{cases} P_{2Ax} = 6.193, P_{2Ay} = 2.425 \\ P_{2Bx} = 7.443, P_{2By} = 1.012 \\ P_{2Cx} = 3.807, P_{2Cy} = -2.204 \\ P_{2Dx} = 2.557, P_{2Dy} = -0.791 \end{cases}$ $\begin{cases} P_{3Ax} = -0.845, P_{3Ay} = 7.420 \\ P_{3Bx} = 0.620, P_{3By} = 8.291 \\ P_{3Cx} = 2.945, P_{3Cy} = 4.380 \\ P_{3Dx} = 1.480, P_{3Dy} = 3.509 \end{cases}$ $\begin{cases} P_{4Ax} = 1.471, P_{4Ay} = -1.679 \\ P_{4Bx} = 1.377, P_{4By} = -3.563 \\ P_{4Cx} = -3.471, P_{4Cy} = -3.321 \\ P_{4Dx} = -3.377, P_{4Dy} = -1.437 \end{cases}$ and $\begin{cases} x_0 = -3 \\ y_0 = -1.25 \\ BL = 6 \\ BW = 2.5 \end{cases}$

[a] x_0 and y_0 originate from Fig. 5. These two parameters, together with BL and BW, determine the location and size of the terminal parking box region. (P_{iAx}, P_{iAy}), (P_{iBx}, P_{iBy}), (P_{iCx}, P_{iCy}) and (P_{iDx}, P_{iDy}) respectively denote the four (clockwise) edge points of the ith rectangular obstacle in the environment.

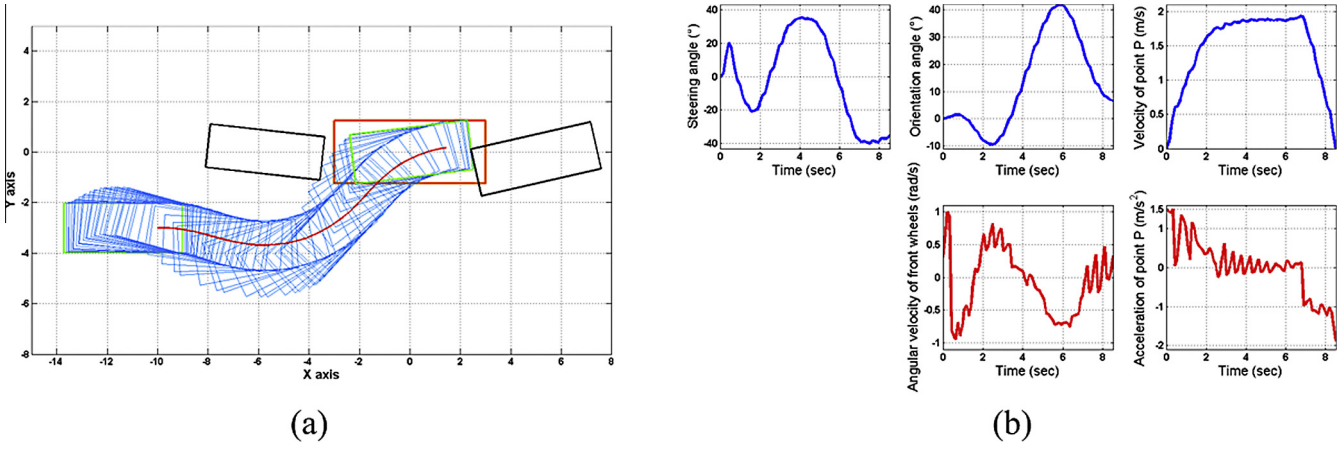

Fig. 10. Optimization results for Case 1: (a) optimized parking motions and (b) optimized control/state variables associated with the optimized motions. The optimized objective $t_f = 8.515$. Note that the solid curve connecting the starting and terminal configurations denotes the trajectory of point P (which is located at the mid-point of front wheel axis).

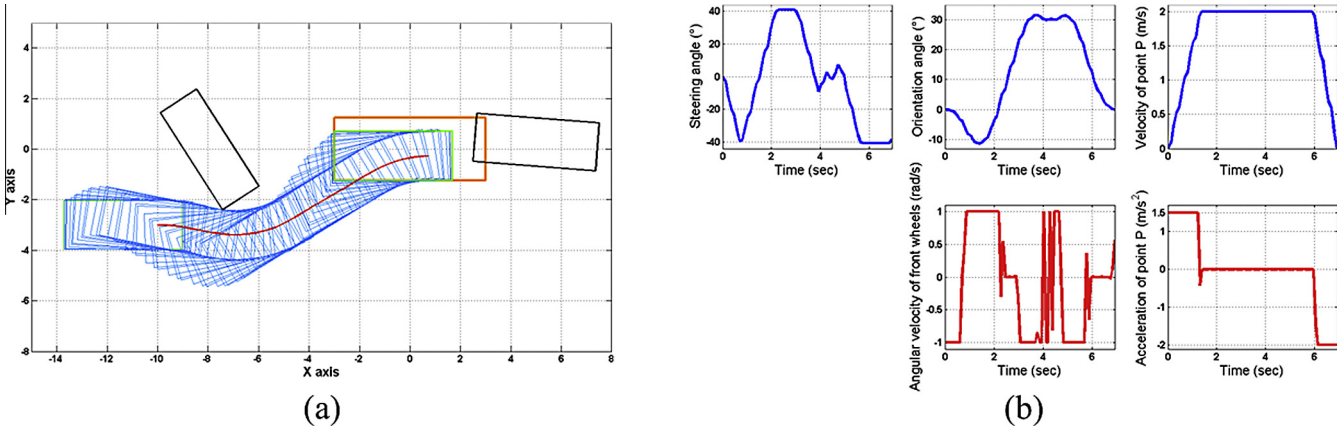

Fig. 11. Optimization results for Case 2: (a) optimized parking motions and (b) optimized control/state variables associated with the optimized motions. The optimized objective $t_f = 6.919$.

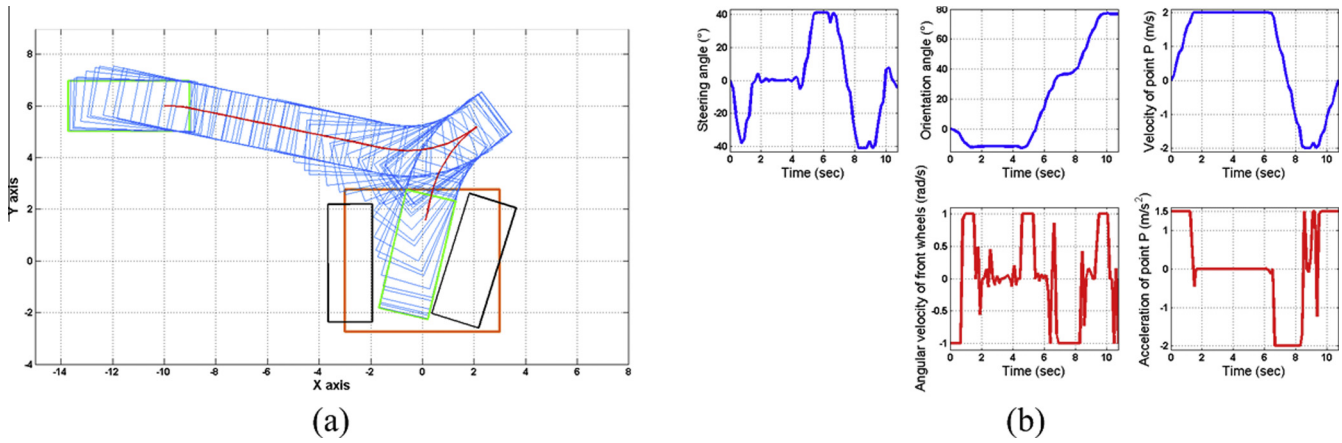

Fig. 12. Optimization results for Case 3: (a) optimized parking motions and (b) optimized control/state variables associated with the optimized motions. The optimized objective $t_f = 10.708$.

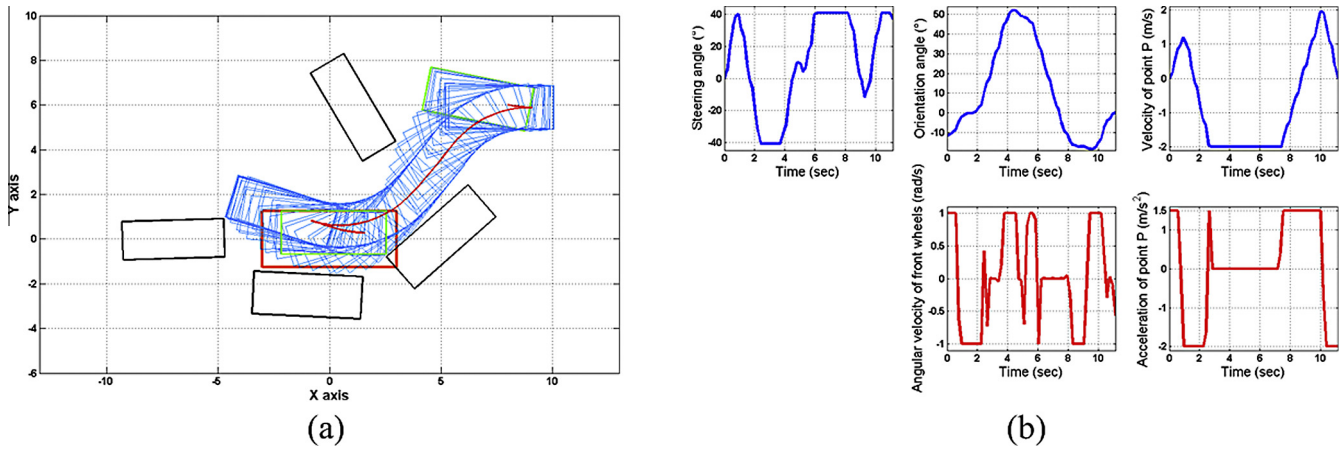

Fig. 13. Optimization results for Case 4: (a) optimized parking motions and (b) optimized control/state variables associated with the optimized motions. The optimized objective $t_f = 11.121$.

approach) to handle parking motion planning problems. The parking scenario details are listed in Table 2. The optimization results are illustrated in Figs. 10–13.

5. Discussions

This section focuses on some analyses behind the obtained simulation results.

5.1. On the performance of the simulation results

Case 1 is often regarded as a parallel parking scenario. Unlike most of the previous studies that consider circumstances with regularly placed obstacles around, Case 1 is concerned with the possibility that a target parking region is occupied in part by a parked car. As shown in Fig. 10(a), our autonomous car manages to find a trajectory toward the destination. Interestingly, the car fully stops as soon as its last one edge point locates on the terminal parking box's borderline. This observed phenomenon is easy to understand: once all the terminal conditions are met, if the car still moves, t_f increases. Recall that t_f is the minimization objective, a relatively "economic" trajectory should be arranged in the optimization procedure. Similar phenomenon can be easily found in the remaining three cases. Case 2 considers a slightly different scenario in which one parked car is in the way of the to-be-parked car. As illustrated in Fig. 11(a), motions are computed accordingly to avoid collision. Case 3 involves a perpendicular parking scenario with two irregularly parked cars. The obtained motions presented in Fig. 12(a) accord with common sense movements. In Case 4, we placed more obstacles in the environment to complicate the parking mission. Given the optimized motions in Fig. 13(a), two maneuvers are necessary. In the first maneuver, the car reverses in an appropriate direction. The second maneuver assists the car inside the box region completely.

The optimized control/state profiles deserve discussion also. Taking Case 1 for example (see Fig. 10(b)), we find that those three state variables depicted in the first row (i.e., $\phi(t), \theta(t)$ and $v(t)$) are smoother than the control profiles plotted in the second row (i.e., $\omega(t)$ and $a(t)$). The rationale behind this phenomenon is follows. First, since there are only bounded constraints imposed on the control variables, the control profiles are not prohibited to present an oscillation form. Those state variables, on the other hand, should be differentiable in addition to satisfying the bounded constraints, thus the state profiles should be continuous. Another way to comprehend is, those state variables are computed through integral, which reduces the oscillation that exists in the integrand (i.e., $\omega(t)$ or $a(t)$). Besides the smoothness issue, one can also notice that $\phi(t_f)$ does not necessarily equal zero. That is because we do not impose that sort of unrealistic

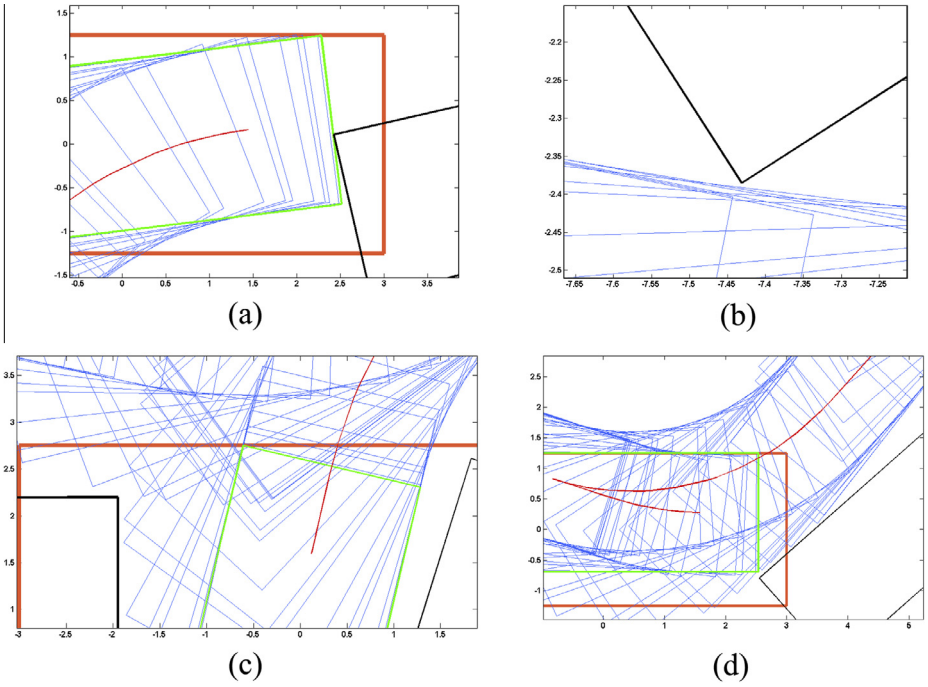

Fig. 14. Local perspectives of the optimized motions in four parking scenarios.

constraints in our formulation. In the authors' viewpoint, if the autonomous vehicle finally stops in a target box region, no other issues deserve consideration.

Viewing the four parking scenarios, we may notice that there is not any parked car (i.e., obstacle) that locates exactly parallel/vertical to the x coordinate. This is common. In fact, there are far more parking scenario categories than the three prevailing ones (i.e., parallel/perpendicular/echelon parking) that deserve investigation. Instead of developing motion generation principles for more classified parking scenario categories, we view all the possible parking scenarios in a same framework and then try to find solutions in a same manner. This is where the unification of our model/method lies in.

Besides that, in all four cases, the car does not collide with the parked cars during its movements because the environment description has been formulated accurately in Section 2 without any abstraction or approximation. This accuracy can be validated with the magnified view in Fig. 14.

Due to the unification of our model, a strong optimization solver is required to solve the formulated dynamic optimization problem. An analytical method that solves such generalized optimal control problems has not been developed. Thus, a numerical method is adopted in this work. However, numerical methods commonly discretize the continuous variables into pieces, resulting in a large number of discretized variables to optimize and constraints to consider. In fact, given that $Nfe = 20$, there are as many as 1167 variables to optimize in the transformed NLP formulation, where 2458 equality/inequality constraints exist as well (for Cases 1–3). In Case 4, the number of equality/inequality constraints soars to 3722 (because the obstacle number doubles). The successful motion planning results indicate that IPM is an efficient large-scale NLP solver. In spite of the success, it is still possible what we have achieved are local optimal solutions. For the convenience of future comparisons that might be conducted by other researchers, we provide all the scenario details in Table 2, making those four scenarios benchmark cases for subsequent methods to compare with.

5.2. On the prospect of the concerned technique

Some readers may wonder how this technique can be applied in the industry. Two questions that may be raised are (i) how to capture the environmental obstacles accurately before a car intends to park along the road side? (ii) even when a car manages to park itself, it may be difficult for the passengers to open the car doors to get off (e.g., see Fig. 12(a)).

Instead of roadside parking, our proposed technique is designed for parking an autonomous car in an indoor intelligent garage. The passengers get off the car at the garage gate before the car drives itself into the garage and then finds a vacant spot in a fully autonomous manner. Moreover, there will be a sufficient number of sensors distributed on the garage ceiling/floor so as to collect a complete knowledge of the environment. Audi revealed the concept of such a parking-garage system at 2013 Consumer Electronics Show in Las Vegas. We believe that this concept will be widely applied soon. At that time, our concerned motion planning technique will be more useful.

6. Conclusions and future work

In this paper, we have proposed a motion planner for the autonomous parking missions. The potential highlights of this work are listed as follows.

First, our proposed technique targets directly on the industrial frontier and caters for true needs. When a car enters such an intelligent garage, it makes decisions in a fully automated manner. We believe that such a parking-garage system helps form a "small world" of intelligent transportation. Investigations on this small world act as a preliminary but critical step to establishing an urban intelligent transportation system with fully autonomous cars travelling in the streets.

Second, our formulated dynamic optimization model aims to describe the parking missions in a unified, accurate, realistic and flexible way. In detail, (i) various kinds of parking scenarios are

described in a unified manner; (ii) no abstraction or approximation exists in the dynamic optimization model; (iii) the inside-box terminal condition and the thought to let $\phi(t_f)$ free cater for true needs; and (iv) other user-specific constraints/conditions/criteria we did not mention or consider can still be added in this framework provided that they can be described explicitly through equalities/inequalities.

Third, the strong capability of the IPM-based simultaneous approach guarantees the success to solve the formulated unified dynamic optimization problems. Although no analytical solutions are achieved, the formulated dynamic optimization problem can be solved at a precision of 10^{-12}. In fact, this is the first time that the high-efficient simultaneous approach has been applied in this research area.

Most importantly, our proposed technique is based on a complete and objective knowledge of the world (because all the necessary knowledge, i.e., the kinematic principle and environment information are accurately described in our formulation). In this sense, compared to the prevailing methods which are based on the assimilated (subjective) human parking knowledge, our unique thought enriches the concept of knowledge-based systems and opens a brand-new gate for this industry.

Our proposed technique will be implemented on a real robot. Then, providing real-time solutions efficiently (like that in [35]) is a critical issue, which will be our future work.

Acknowledgements

The authors are sincerely grateful to the anonymous reviewers, Ms. R. Zhou, and Dr. K. Wang for their valuable comments and suggestions, as well as to the editor for the efficient and professional processing of this paper. This work is supported by the 973 Program of China under Grant 2012CB720503, National Nature Science Foundation under Grant 61374167 and the 6th National College Students' Innovation & Entrepreneurial Training Program under Grant 201210006050.

References

[1] G. Ullrich, Automated Guided Vehicle Systems, Springer, 2015.
[2] S. Le Vine, A. Zolfaghari, J. Polak, Autonomous cars: the tension between occupant experience and intersection capacity, Transport. Res. Part C: Emerg. Technol. 52 (2015) 1–14.
[3] M. Swan, Connected car: quantified self becomes quantified car, J. Sensor Actuator Networks 4 (1) (2015) 2–29.
[4] U. Kiencke, L. Nielsen, R. Sutton, K. Schilling, M. Papageorgiou, H. Asama, The impact of automatic control on recent developments in transportation and vehicle systems, Annu. Rev. Control 30 (1) (2006) 81–89.
[5] J.M. Blosseville, M. Flonneau, From traffic signal control systems to automated driving, a review of ITS systems based on a cross historical and technical perspective, in: Proceedings of 5th Conference on Transport Research Arena, TRA 2014, 2014, pp. 1–7.
[6] A. Sekmen, P. Challa, Assessment of adaptive human–robot interactions, Knowl.-Based Syst. 42 (2013) 49–59.
[7] W.Y. Kwon, I.H. Suh, Planning of proactive behaviors for human–robot cooperative tasks under uncertainty, Knowl.-Based Syst. 72 (2014) 81–95.
[8] C.K. Lee, C.L. Lin, B.M. Shiu, Autonomous vehicle parking using hybrid artificial intelligent approach, J. Intell. Robot. Syst. 56 (3) (2009) 319–343.
[9] A. Ohata, M. Mio, Parking control based on nonlinear trajectory control for low speed vehicles, in: Proceedings of 1991 International Conference on Industrial Electronics, Control and Instrumentation, IECON'91, 1991, pp. 107–112.
[10] Y. Zhu, T. Zhang, J. Song, X. Li, A new hybrid navigation algorithm for mobile robots in environments with incomplete knowledge, Knowl.-Based Syst. 27 (2012) 302–313.
[11] D. Dolgov, S. Thrun, M. Montemerlo, J. Diebel, Path planning for autonomous vehicles in unknown semi-structured environments, Int. J. Robot. Res. 29 (5) (2010) 485–501.
[12] H. Vorobieva, N. Minoiu-Enache, S. Glaser, S. Mammar, Geometric continuous-curvature path planning for automatic parallel parking, in: Proceedings of 10th IEEE International Conference on Networking, Sensing and Control, ICNSC 2013, 2013, pp. 418–423.
[13] A.A. Zhdanov, D.M. Klimov, V.V. Korolev, A.E. Utemov, Modeling parallel parking a car, Int. J. Comput. Syst. Sci. 47 (6) (2008) 907–917.
[14] A. Bicchi, G. Casalino, C. Santilli, Planning shortest bounded-curvature paths for a class of nonholonomic vehicles among obstacles, J. Intell. Robot. Syst. 16 (4) (1996) 387–405.
[15] B. Muller, J. Deutscher, S. Grodde, Continuous curvature trajectory design and feedforward control for parking a car, IEEE Trans. Control Syst. Technol. 15 (3) (2007) 541–553.
[16] F. Gómez-Bravo, F. Cuesta, A. Ollero, A. Viguria, Continuous curvature path generation based on β-spline curves for parking manoeuvres, Robot. Auton. Syst. 56 (4) (2008) 360–372.
[17] Y. Wang, M.P. Cartmell, Trajectory generation for a four wheel steering tractor–trailer system: a two-step method, Robotica 16 (4) (1998) 381–386.
[18] Y. Zhao, E.G. Collins Jr., Robust automatic parallel parking in tight spaces via fuzzy logic, Robot. Auton. Syst. 51 (2) (2005) 111–127.
[19] T.H. Li, S.J. Chang, Autonomous fuzzy parking control of a car-like mobile robot, IEEE Trans. Syst. Man Cybern., Part A: Syst. Hum. 33 (4) (2003) 451–465.
[20] L. Han, Q.H. Do, S. Mita, Unified path planner for parking an autonomous vehicle based on RRT, in: Proceedings of 2011 IEEE International Conference on Robotics and Automation, ICRA 2011, pp. 5622–5627.
[21] H. Kwon, W. Chung, Performance analysis of path planners for car-like vehicles toward automatic parking control, Intell. Serv. Robot. 7 (1) (2014) 15–23.
[22] B. Park, J. Choi, W.K. Chung, Sampling-based retraction method for improving the quality of mobile robot path planning, Int. J. Control Automat. Syst. 10 (5) (2012) 982–991.
[23] Q. Huy, M.I.T.A. Seiichi, K. Yoneda, A practical and optimal path planning for autonomous parking using fast marching algorithm and support vector machine, IEICE Trans. Inform. Syst. 96 (12) (2013) 2795–2804.
[24] M.F. Hsieh, A parking algorithm for an autonomous vehicle, in: Proceedings of 2008 IEEE Intelligent Vehicles Symposium, 2008, pp. 1155–1160.
[25] W. Yang, Optimal Approach for Autonomous Parallel Parking of Nonholonomic Car-Like Vehicle, M.S. Thesis, Department of Mechanical & Aerospace Engineering, State University of New York at Buffalo, New York, 2006.
[26] W.N. Patten, H.C. Wu, W. Cai, Perfect parallel parking via Pontryagin's principle, J. Dyn. Syst. Measure. Control 116 (4) (1994) 723–728.
[27] K. Kondak, G. Hommel, Computation of time optimal movements for autonomous parking of non-holonomic mobile platforms, in: Proceedings of 2001 IEEE International Conference on Robotics and Automation, ICRA 2001, vol. 3, 2001, pp. 2698–2703.
[28] I.E. Paromtchik, C. Laugier, Autonomous parallel parking of a nonholonomic vehicle, in: Proceedings of 1996 IEEE Intelligent Vehicles Symposium, 1996, pp. 13–18.
[29] I.E. Paromtchik, C. Laugier, Motion generation and control for parking an autonomous vehicle, in: Proceedings of 1996 IEEE International Conference on Robotics and Automation, ICRA 1996, vol. 4, pp. 3117–3122.
[30] I.E. Paromtchik, C. Laugier, Automatic parallel parking and returning to traffic manoeuvres, in: Proceedings of 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 1997, vol. 3, 1997, pp. 21–23.
[31] S. Kameswaran, L.T. Biegler, Simultaneous dynamic optimization strategies: recent advances and challenges, Comput. Chem. Eng. 30 (10) (2006) 1560–1575.
[32] A. Wächter, L.T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Math. Program. 106 (1) (2006) 25–57.
[33] L.T. Biegler (Ed.), Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes, SIAM, 2010.
[34] R. Fourer, D.M. Gay, B.W. Kernighan, AMPL: A Modeling Language for Mathematical Programming, second ed., Brooks/Cole-Thomson Learning, Pacific Grove, 2003.
[35] W. Liu, Z. Zheng, K.Y. Cai, Bi-level programming based real-time path planning for unmanned aerial vehicles, Knowl.-Based Syst. 44 (2013) 34–47.