

Spatio-temporal decomposition: a knowledge-based initialization strategy for parallel parking motion optimization



Bai Li^a, Youmin Zhang^{b,c}, Zhijiang Shao^{a,d,*}

^a College of Control Science and Engineering, Zhejiang University, Hangzhou, China

^b Department of Mechanical and Industrial Engineering, Concordia University, Montreal, Canada

^c Concordia Institute of Aerospace Design and Innovation, Concordia University, Montreal, Canada

^d State Key Laboratory of Industrial Control Technology, Hangzhou, China

ARTICLE INFO

Article history:

Received 29 January 2016

Revised 27 April 2016

Accepted 8 June 2016

Available online 8 June 2016

Keywords:

Nonlinear programming

Parallel parking

Initial guess

Objective computational intelligence

Knowledge based system

ABSTRACT

Motion planning methodologies for parallel parking have been well developed in the last decade. In contrast to the prevailing and emerging parking motion planners, this work provides a precise and objective description of the parking scenario and vehicle kinematics/dynamics. This is achieved by formulating a unified optimal control problem that is free of subjective knowledge (e.g., human experiences). The concerned optimal control problem, when parameterized into a large-scale nonlinear programming (NLP) problem, is extremely difficult to solve. This bottleneck has hindered many research efforts previously. Although the feasible regions of NLP problems are clearly defined, the majority of NLP-solving processes still require high-quality initial guesses, which accelerate the convergence process. In this work, we propose a spatio-temporal decomposition based initialization strategy to generate reliable initial guesses, so as to facilitate the NLP-solving process. In contrast to the typical facilitation strategies in robotic motion/path planning, our spatio-temporal decomposition strategy considers only objective knowledge, further breaking the limitation of subjective knowledge and making full use of a vehicle's maneuver potential. A series of comparative simulations verifies that the proposed initialization strategy is advantageous over its prevailing competitors, and that the proposed motion planner is promising for on-line planning missions. Theoretical analysis that supports our initialization strategy is given as well.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Parking refers to driving a vehicle into a parking spot. Among various parking types (e.g., parallel, perpendicular, and angle parking), parallel parking is the most frequently applied in urban life [1]. The incessant increase in the number of cars has led to a decrease in available parking spaces along streets [2], further aggravating the challenges to parallel parking [3,4]. In addition to promoting manual operation capabilities [5], parking assist systems and autonomous parking systems have been developed to address these modern parking challenges [6]. Specifically, parking assist systems provide broad perspectives during manual parking operations; autonomous parking systems can fully control the driving velocity and steering angle without manual assistance. Given the unreliability and uncertainty in manual operations [7–9], applying a fully automatic system for parking schemes would be nominally beneficial.

An autonomous parking system generally consists of three stages, namely, *perception*, *planning*, and *execution*. Perception contributes to establishing a parking scenario via vehicular sensors; planning provides a satisfactory path/trajectory towards the parking destination considering vehicle kinematics, maneuver capabilities, and traffic rules; execution is the implementation of the planned results [10]. Planning has typically been treated as a less important procedure than control execution [11–16], for the following reasons: (i) precise path/trajectory planning is computationally expensive, and (ii) fast path/trajectory planning is rough, but it can be compensated for at the closed-loop execution stage. However, the prevailing methodologies are not efficient in dealing with extremely difficult parking missions: relying on the control execution stage to always make perfect corrections is unrealistic. In other words, the intelligence of an autonomous parking system is reflected not by the merits of its control stability/robustness, but rather by its decision-making quality. Rapid developments in computing hardware have brought about underlying possibilities for robotics to solve accurate motion planning problems fast. This study focuses on precise motion planning in parallel parking missions because of its great potential.

* Corresponding author.

E-mail addresses: libai@zju.edu.cn, libaioutstanding@163.com (B. Li), youmin.zhang@concordia.ca (Y. Zhang), szj@zju.edu.cn (Z. Shao).

An optimal control problem is formulated to accurately describe the parallel parking motion planning problem; the optimal control problem consists of the following (i) differential equations that describe vehicle kinematics, (ii) algebraic equations/inequalities that describe the mechanical restrictions and parking scenario constraints, and (iii) a specified optimization objective [17]. Given the complexity and unification of this formulated optimal control problem, analytical solutions are generally unobtainable [18]. An alternative method to solve optimal control problems is by converting them into parameter optimization problems, that is, nonlinear programming (NLP) problems [19–21]. However, the corresponding NLP problems are NP-hard and thus have hindered a vast of research efforts in robotics [18]. A near-feasible initial guess is widely known to facilitate an NLP-solving process [22–26]. Herein, an initial guess refers to a guessed solution from which an optimization process starts. An initial guess choice can even alter the terminal convergence result in solving an optimization process, when gradient-based optimizers are used [27]. This study considers finding initial guesses that are generated fast, aiming to facilitate the formal NLP-solving process.

In order to generate preliminary paths/trajectories (i.e., initial guesses) fast, two typical branches have been developed in robotics. The first branch concerns about decoupling or simplifying the original planning scheme. Notomista and Botsch [12] assumed that a complete parallel parking event can be decoupled into three periods: going straight backward, going backward with the steering angle fixed to one bound, and going backward with the steering angle fixed to the other bound. Through this, parking motion planning is converted to determining the switching moments between adjacent periods. Similarly, Moon et al. [28] proposed a first-adjusting-then-entering strategy to decouple the original problem into two periods. Fiorini and Shiller [29] generated initial paths via tree search with simplified vehicle kinematics. Kant and Zucker [30] eased parking trajectory planning problems via a first-path-then-trajectory decoupling strategy. This pioneering work has inspired subsequent studies to do planning preliminarily in low dimensional solution spaces [31,32]. You et al. [33] introduced circumcircles to cover rectangular vehicles, aiming to ease the collision-avoidance descriptions. The second branch involves alternative or enhanced optimizers. Zhang and Huang [34] adopted a metaheuristic algorithm to initially seek for feasible solutions, followed by a formal gradient-based optimization process. Li and Shao [27] proposed an incremental strategy that gradually directs the optimization process towards global optimality. Existing facilitation strategies suffer from two common defects: (i) facilitations are usually made on the basis of subjective knowledge or experiences; (ii) decoupling the original problem into multiple phases would reduce solution optimality. Regarding defect (i), although some decoupling strategies appear to be smart and intuitively reasonable, the incorporated subjective knowledge can seldom lead to solutions out of the subjective expectations, thus is inefficient to guarantee solution unification. For example, the proposal in Ref. [12] cannot cope with extremely tiny parking spots, because far more periods are needed. In this sense, unified problem-solving capability originates from the utilization of purely objective knowledge, rather than biased experiences. Regarding the second defect, most previous decoupling-based studies would initially generate a rough solution, and then make post-processing efforts to achieve solution feasibility [35]. Although multi-phase strategies guarantee fast generation of paths/trajectories, the decoupled problems, when combined together, are not identical to original ones, thereby losing nominal solution optimality. Conversely, the availability of faster and cheaper computers as well as efficient initialization algorithms has largely eased the computational burden in solving a centralized (instead of decoupled) parking motion planning problem in robotics [36], even rendering

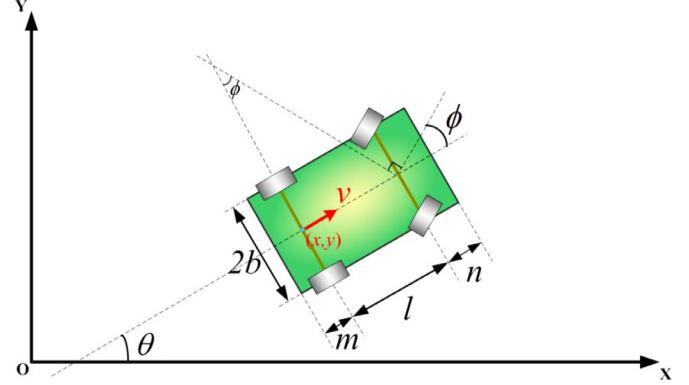


Fig. 1. Parametric notations related to vehicle size and kinematics.

real-time optimal control of complex nonlinear systems [37,38]. This work proposes an objective knowledge based initialization strategy to facilitate the parallel parking motion planning problem. Through this, our proposed methodology aims to tackle various parallel parking scenarios in a unified way.

The rest of this paper is organized into the following sections. Section 2 formulates the parallel parking motion planning scheme as an NLP problem. Principle of the NLP solver is presented in Section 3. Introduction of the initialization strategy is given in Section 4. Simulation results are provided in Section 5, together with in-depth discussions. Conclusions are drawn in Section 6.

2. Motion planning problem formulation

In this section, the original parallel parking motion planning mission is first formulated as an optimal control problem and then parameterized as an NLP problem.

2.1. Optimal control problem formulation

Kinematics of a front-steering automobile is described through differential equations [20]:

$$\left\{ \begin{array}{l} \frac{dx(t)}{dt} = v(t) \cdot \cos \theta(t) \\ \frac{dy(t)}{dt} = v(t) \cdot \sin \theta(t) \\ \frac{dv(t)}{dt} = a(t), \quad t \in [0, t_f], \\ \frac{d\theta(t)}{dt} = \frac{v(t) \cdot \tan \phi(t)}{l} \\ \frac{d\phi(t)}{dt} = \omega(t) \end{array} \right. \quad (1)$$

where t stands for time, t_f stands for the (unknown) completion time of parking mission, (x, y) denotes the mid-point along the rear wheel axis, θ denotes the orientation angle with respect to the x axis as defined in Fig. 1, v refers to the velocity of point (x, y) , a refers to the corresponding acceleration, ϕ refers to the steering angle of front wheels, ω refers to the corresponding angular velocity and l denotes the wheelbase length. Other geometric parameters regarding vehicle size include n (the front overhang length), m (the rear overhang length), and $2b$ (the car width). $\omega(t)$ and $a(t)$ are selected as control variables in this kinematic system.

In addition, several interior constraints should be imposed to guarantee physical realizability and tracking convenience [38]:

$$\left\{ \begin{array}{l} |a(t)| \leq a_{\max} \\ |v(t)| \leq v_{\max} \\ |\phi(t)| \leq \Phi_{\max}, \quad t \in [0, t_f], \\ |\omega(t)| \leq \Omega_{\max} \end{array} \right. \quad (2)$$

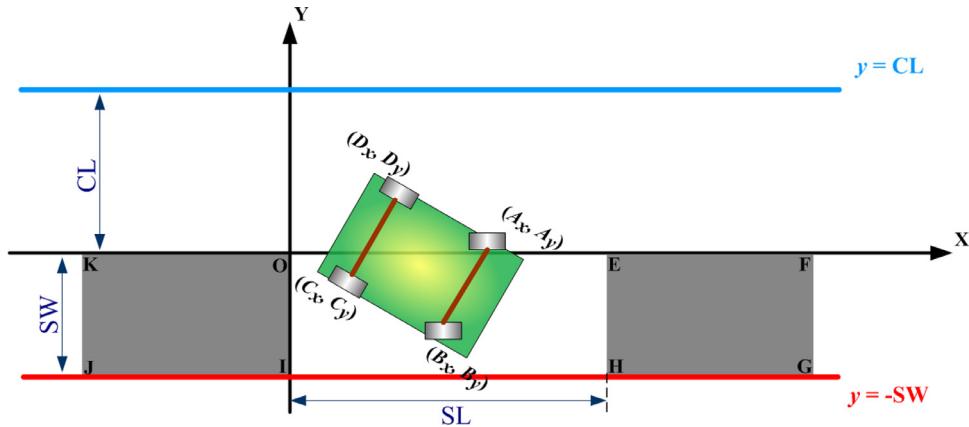


Fig. 2. Schematics on a parallel parking scenario. Note that $SL > 0$ represents the parking spot length, $SW > 0$ denotes the spot width, and $CL > 0$ denotes the lane width. Functions $y = CL$ and $y = -SW$ define the boundaries of admissible parking space.

Besides the vehicle kinematics and physical/mechanical restrictions, parallel parking environment is established next. If a vacant parallel parking spot is formed by two parked cars on both sides (Fig. 2), the concerned parking vehicle is expected to terminally enter the parking spot. Specifically, the admissible parking space is defined by

$$\left\{ \begin{array}{l} -SW \leq A_y(t) \leq CL \\ -SW \leq B_y(t) \leq CL \\ -SW \leq C_y(t) \leq CL \\ -SW \leq D_y(t) \leq CL \end{array} , \text{ for } t \in [0, t_f]. \right. \quad (3)$$

where

$$\left\{ \begin{array}{l} A = (A_x, A_y) = (x + (l + n) \cdot \cos\theta - b \cdot \sin\theta, \\ \quad y + (l + n) \cdot \sin\theta + b \cdot \cos\theta) \\ B = (B_x, B_y) = (x + (l + n) \cdot \cos\theta + b \cdot \sin\theta, \\ \quad y + (l + n) \cdot \sin\theta - b \cdot \cos\theta) \\ C = (C_x, C_y) = (x - m \cdot \cos\theta + b \cdot \sin\theta, \\ \quad y - m \cdot \sin\theta - b \cdot \cos\theta) \\ D = (D_x, D_y) = (x - m \cdot \cos\theta - b \cdot \sin\theta, \\ \quad y - m \cdot \sin\theta + b \cdot \cos\theta) \end{array} \right. \quad (4)$$

together with collision-avoidance restrictions between the parking and either parked vehicle during the entire parking process. The collision-avoidance restrictions can be precisely described through the recently proposed “triangle area based criterion” [20]: when rectangle ABCD is required not to collide with rectangle EFGH (for example, Fig. 2), all the four corner points of ABCD must be outside region EFGH, and that all the four corner points of EFGH must be outside region ABCD:

$$\left\{ \begin{array}{l} S_{\Delta XAB} + S_{\Delta XBC} + S_{\Delta XCD} + S_{\Delta XDA} > S_{\square ABCD}, X \in \{E, F, G, H\} \\ S_{\Delta XEF} + S_{\Delta XFG} + S_{\Delta XGH} + S_{\Delta XHE} > S_{\square EFGH}, X \in \{A, B, C, D\} \end{array} \right. , \quad (5a)$$

where S_{Δ} denotes triangle area and S_{\square} denotes rectangle area. Collision-avoidance restrictions between rectangles ABCD and OIJK can be written in a similar way:

$$\left\{ \begin{array}{l} S_{\Delta XAB} + S_{\Delta XBC} + S_{\Delta XCD} + S_{\Delta XDA} > S_{\square ABCD}, X \in \{O, I, J, K\} \\ S_{\Delta XOI} + S_{\Delta XIJ} + S_{\Delta XJK} + S_{\Delta XKO} > S_{\square OIJK}, X \in \{A, B, C, D\} \end{array} \right. . \quad (5b)$$

A parallel parking process starts from a specified configuration with all the state variables explicitly given at $t = 0$. When $t = t_f$, the vehicle should stop within the parking spot, indicating that

$$v(t_f) = 0 \text{ and}$$

$$\left\{ \begin{array}{l} A_y(t_f) \leq 0 \\ B_y(t_f) \leq 0 \\ C_y(t_f) \leq 0 \\ D_y(t_f) \leq 0 \\ 0 \leq A_x(t_f) \leq SL \\ 0 \leq B_x(t_f) \leq SL \\ 0 \leq C_x(t_f) \leq SL \\ 0 \leq D_x(t_f) \leq SL \end{array} \right. . \quad (6)$$

The present study aims to find the motion that accomplishes the parking mission with minimum time, i.e., to solve a min-time optimal control problem.

2.2. NLP problem formulation

Given that the analytical solution to a general optimal control problem is unobtainable, NLP-based methodologies have been developed to form approximations to the original continuous problem. This study adopts an orthogonal collocation direct transcription method, which is featured by simultaneously discretizing not only controls but also state profiles in time. In contrast to the prevailing sequential strategies, the adopted direct transcription method possesses high order accuracy and excellent stability [39].

For the convenience of presentation, the formulated continuous optimal control problem is abstracted as

$$\min t_f,$$

$$\text{s.t.} \left\{ \begin{array}{l} \frac{dz(t)}{dt} = f(z(t), \xi(t), u(t)), \quad t \in [0, t_f] \\ g(z(t), \xi(t), u(t)) \leq 0 \end{array} \right. . \quad (7)$$

Herein, $z(t)$ denotes the differential state variables (i.e., $x(t)$, $y(t)$, $v(t)$, $\theta(t)$ and $\phi(t)$) in Eq. (1), $\xi(t)$ denotes the algebraic state variables ($A_x(t)$, $B_x(t)$, and so on), $u(t)$ denotes the control variables $\omega(t)$ and $a(t)$, and $g(z(t), \xi(t), u(t)) \leq 0$ represents all the aforementioned algebraic equations/inequalities. The decision variables in Eq. (7) include $z(t)$, $\xi(t)$, $u(t)$, and t_f . The concerned orthogonal collocation direct transcription method approximates $z(t)$, $\xi(t)$ and $u(t)$ via piecewise Lagrange polynomials. To begin with, the entire period $[0, t_f]$ is divided into N_{sp} intervals $\{[t_{i-1}, t_i]|i = 1, 2, \dots, N_{sp}\}$, wherein $t_{N_{sp}} = t_f$ and $t_0 = 0$. The present study considers equidistant division in $[0, t_f]$. Thus, each sub-period length h_i is equal to t_f/N_{sp} , $i = 1, 2, \dots, N_{sp}$. In each sub-period, algebraic state $\xi(t)$ is

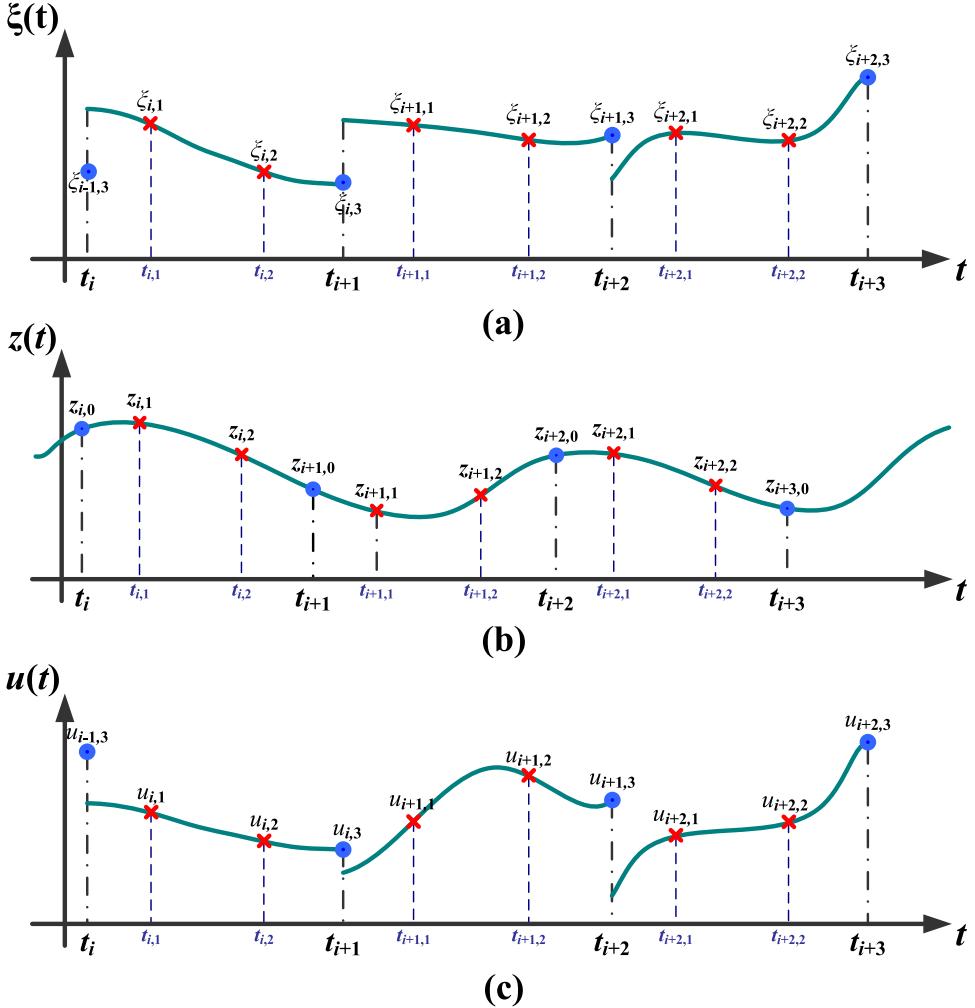


Fig. 3. Piecewise Lagrange polynomial approximations for (a) algebraic states, (b) differential states, and (c) controls.

approximated using K collocation points:

$$\xi(t_{i-1} + h_i \cdot \tau) = \sum_{j=1}^K \left(\xi_{ij} \cdot \prod_{k=0, k \neq j}^K \frac{(\tau - \gamma_k)}{(\gamma_j - \gamma_k)} \right), \quad (8)$$

where ξ_{ij} stands for the j th collocation point in the i th sub-period ($\tau \in [0, 1]$, $i = 1, 2, \dots, N_{sp}$, $j = 1, 2, \dots, K$). $\{\gamma_k\}_{0=0}^K = \gamma_0 < \gamma_1 < \dots < \gamma_K \leq 1\}$ represents Radau points and should be given offline by the Gaussian quadrature accuracy theorem [39]. Similarly, control variable $u(t)$ is approximated by K collocation points in each sub-period:

$$u(t_{i-1} + h_i \cdot \tau) = \sum_{j=1}^K \left(u_{ij} \cdot \prod_{k=0, k \neq j}^K \frac{(\tau - \gamma_k)}{(\gamma_j - \gamma_k)} \right). \quad (9)$$

Differential state variable $z(t)$ is presented in a slightly distinct way because $z(t)$ is necessarily continuous across sub-period boundaries:

$$z(t_{i-1} + h_i \cdot \tau) = \sum_{j=0}^K \left(z_{ij} \cdot \prod_{k=0, k \neq j}^K \frac{(\tau - \gamma_k)}{(\gamma_j - \gamma_k)} \right). \quad (10)$$

For $N_{sp} > 1$, continuity across sub-period boundaries is enforced as

$$z_{i+1,0} = z_{i,K} = \sum_{j=0}^K \left(z_{ij} \cdot \prod_{k=0, k \neq j}^K \frac{(1 - \gamma_k)}{(\gamma_j - \gamma_k)} \right), \quad i = 1, 2, \dots, N_{sp} - 1. \quad (11)$$

Fig. 3 sketchily depicts the collocation based piecewise Lagrange polynomials. Substituting Eqs. (8)–(11) into Eq. (7) forms

$$\begin{aligned} & \min_{z_{ij}, \xi_{ij}, u_{ij}, t_f} t_f, \\ & \text{s.t. } \begin{cases} \sum_{k=0}^K \left(\frac{d \left(\prod_{k=0, k \neq j}^K \frac{(\tau - \gamma_k)}{(\gamma_j - \gamma_k)} \right)}{d\tau} \Big|_{\tau=\gamma_k} \cdot z_{ik} \right) - h_i \cdot f(z_{ij}, \xi_{ij}, u_{ij}) = 0, \\ g(z_{ij}, \xi_{ij}, u_{ij}) \leq 0 \end{cases}, \\ & i = 1, 2, \dots, N_{sp}, j = 1, \dots, K. \end{aligned} \quad (12)$$

Through this, Eq. (12) defines a converted NLP problem whose solution is an approximation to the original optimal control problem in Eq. (7).

3. Interior-point method

The NLP problem formulated in Section 2.2 is solved by interior-point method (IPM) because IPM is efficient for large-scale problems, particularly when a number of active constraints exist [23]. IPM typically computes solutions for a sequence of barrier problems, wherein equality constraints are incorporated into the optimization objective via barrier weighted logarithmic terms. If inequality constraints exist, they are easily recast as equality constraints by introducing additional slack variables. The logarithmic terms ensure the solution-searching process is within the feasible region by penalizing solutions close to the boundaries. As the barrier weights decline to zero, the penalties gradually disappear; this

decline ultimately allows for solutions that converge to the constraint boundaries. A solution to each barrier problem is found by considering the Karush–Kuhn–Tucker (KKT) conditions. Interested readers may refer to Refs. [23,39] for detailed rationales. In the present work, IPM is adopted to determine the decision variables $z_{ik} \in \mathbb{R}^{N_{sp} \times (K+1)}$, $\xi_{ij} \in \mathbb{R}^{N_{sp} \times K}$, $u_{ij} \in \mathbb{R}^{N_{sp} \times K}$ and $t_f \in \mathbb{R}$ in the NLP problem Eq. (12) so as to minimize t_f .

4. Initialization strategy

This section focuses on the generation of initial guess to facilitate the NLP-solving process. The initialization strategy is formally introduced after the motivation. Supporting theoretical analyses are provided as well.

4.1. Motivation

Our initialization strategy is inspired by the objective phenomenon that, a vehicle would adjust its position locally (e.g., goes back and forth) for some time right before it fully stops in a parking spot. Restrictions based on this phenomenon do not appear in Section 2, because the formulated optimal control problem implicitly contains this phenomenon, if it is really true. Although being redundant, this phenomenon can be taken as objective knowledge to ease the NLP-solving burden. To illustrate the inspiration in more detail, we present an instance in Fig. 4, wherein a vehicle should move from point A to B and avoid the circular obstacles on the way. Commonly, the vehicle would check all the collision-avoidance restrictions at every moment before it reaches B. However, one may notice that no obstacle locates close to terminal B, thus, the vehicle will not hit in any obstacle during some time right before it reaches the destination. The collision-free restrictions can be discarded when the car is sufficiently close to spot B, provided that the definition of “sufficiently close” is correct. For example, if the vehicle is particularly required to remain in the dashed box region depicted in Fig. 4 during the second 50% period of the entire movement, then the collision-free conditions can be discarded during that period. In solving the formulated NLP problem, this type of additional requirements would largely simplify the constraints in the second 50% periods, thus making the NLP-solving process easier than the original one. This interesting phenomenon shows that spatio-temporal specifications can make the original problem easier instead of making it more difficult! Our spatio-temporal decomposition strategy is formally introduced in the next subsection.

4.2. Spatio-temporal decomposition strategy

This subsection presents the principle of spatio-temporal decomposition strategy with the backdrop of parallel parking motion

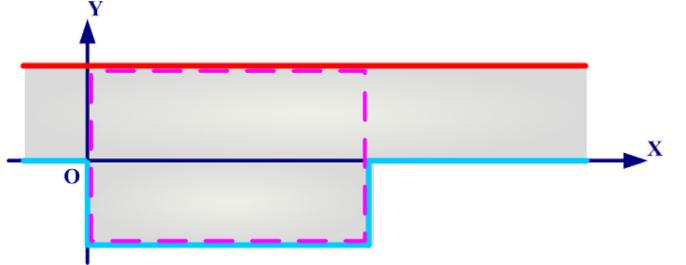


Fig. 5. Schematic on the defined critical region in the spatio-temporal decomposition strategy.

planning. First, a critical region is defined as the dashed box region in Fig. 5. From a specified moment until $t = t_f$, the vehicle should remain in the critical region. The “specified moment” is parameterized through an integer $N_\chi \in [1, N_{sp}]$, yielding the critical region requirement as

$$\begin{cases} -SW \leq A_y(t) \leq CL \\ -SW \leq B_y(t) \leq CL \\ -SW \leq C_y(t) \leq CL \\ -SW \leq D_y(t) \leq CL \\ 0 \leq A_x(t) \leq SL \\ 0 \leq B_x(t) \leq SL \\ 0 \leq C_x(t) \leq SL \\ 0 \leq D_x(t) \leq SL \end{cases}, \quad t \in [h_i \cdot N_\chi, t_f]. \quad (13)$$

Staying in the critical region indicates that the vehicle will certainly not collide with either parked car during the specific period $t \in [h_i \cdot N_\chi, t_f]$. That means, the complicated collision-avoidance restrictions (Eq. (5)) can be replaced by Eq. (13) when $t \in [h_i \cdot N_\chi, t_f]$. Readers may wonder why replacing the eight inequalities in Eq. (5) with the eight ones in Eq. (13) would simplify the NLP formulation. That is because the triangle areas in Eq. (5) (e.g., $S_{\Delta XAB} = \frac{1}{2}|\vec{XA} \times \vec{XB}|$) involve complicated nonlinearity and absolute value operations; conversely, Eq. (13) consists of simple bounded constraints. Suppose that $N_{sp} = 100$, Eq. (13) takes effect during the latter 90% period during $[0, t_f]$ if N_χ is set to 11. As an obvious trend, the corresponding NLP problem is the simplest if N_χ is set to the smallest (i.e., $N_\chi = 1$); conversely, if $N_\chi = N_{sp}$, the concerned NLP is identical to the original one. Our initialization strategy primarily utilizes the inside-box objective knowledge as well as the aforementioned trend. Specifically, an NLP-solving process with $N_\chi = 1$ and without initial guess is carried out at the beginning. The successfully optimized solution is taken as the initial guess in the subsequent NLP-solving procedure ($N_\chi = 2$). This iterative process continues until $N_\chi = N_{sp}$, when the original NLP problem is handled. The complete principle of spatio-temporal

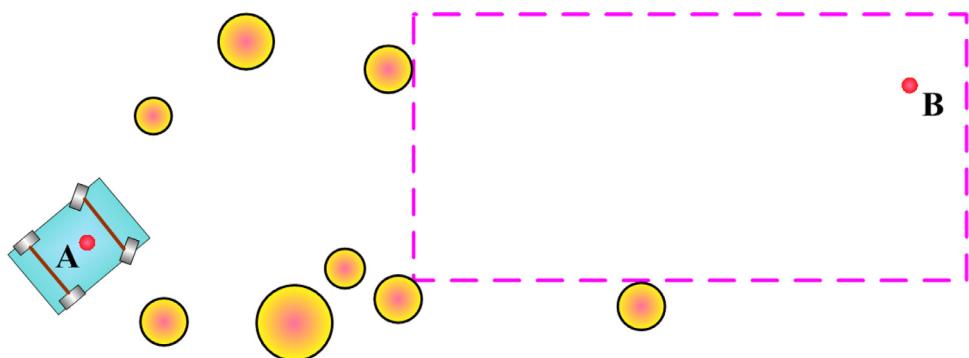


Fig. 4. Schematic on the effect of additional spatio-temporal specifications that simplify the NLP formulation, thereby facilitating the NLP-solving process.

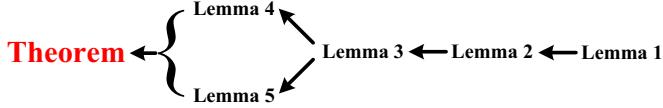


Fig. 6. Pathway of rationality analyses of our spatio-temporal decomposition based initialization method.

Algorithm 1 Spatio-temporal decomposition based initialization strategy.

1. Set $N_X \leftarrow 0$;
2. Set $\text{Rec_t}_f > 0$ sufficiently large;
3. **While** $N_X < N_{\text{sp}}$, **do**
4. Set $N_X \leftarrow N_X + 1$;
5. Solving the N_X based NLP problem using IPM with the initial guess (if there exists);
6. **If** optimal solution is found, **then**
7. **If** the current t_f is smaller than Rec_t_f , **then**
8. Update the initial guess via the optimized solution;
9. Update Rec_t_f by t_f ;
10. **End if**
11. **End if**
12. **End while**
13. Exit

decomposition based initialization strategy is presented in **Algorithm 1**. Interesting theoretical analyses that justify this strategy are given in the next subsection.

4.3. Theoretical analysis

The rationality of the aforementioned initialization methodology is analyzed in this subsection. An overview of the proof procedure is shown in **Fig. 6**, where the establishment of the Theorem is our ultimate goal. For convenience's sake, all of the proofs are presented in the [Appendix](#).

Lemma 1. Consider a path planning scenario depicted in **Fig. 7(a)** for a car-like vehicle system described by [Eq. \(1\)](#) with interior restrictions described by [Eq. \(2\)](#), a circular shape path can be

Table 1
User-specific parametric settings.

Parameter	Description	Setting
ϵ_{tol}	Convergence tolerance in IPM	10^{-6}
K_c	Minimum absolute distance from the initial point to bound in IPM	10^{-4}
K	Number of collocation points in orthogonal collocation direct transcription method	3
N_{sp}	Number of sub-periods in orthogonal collocation direct transcription method	50/100
N_X	Proportion parameter in our initialization methodology	-
SL	Parking slot length	variable
SW	Parking slot width	2.5 m
CL	Lane width	4.0 m
n	Front overhang length	0.96 m
l	Distance between the front and back wheel axes	2.8 m
m	Rear overhang length	0.929 m
b	Vehicle half width	0.971 m
Φ_{max}	Bound of steering angle	0.576 rad
a_{max}	Bound of acceleration	0.75 m/s^2
v_{max}	Bound of velocity	1.8 m/s
Ω_{max}	Bound of steering angular velocity	1.2 rad/s

Table 2
Cases 1–8 with various parking spot lengths.

Case no.	Parking slot length	Special requirement
1	SL = 7.00m	$N_{\text{sp}} = 50$
2	SL = 6.00m	$N_{\text{sp}} = 50$
3	SL = 5.75m	$N_{\text{sp}} = 50$
4	SL = 5.50m	$N_{\text{sp}} = 50$
5	SL = 5.25m	$N_{\text{sp}} = 50$
6	SL = 5.20m	$N_{\text{sp}} = 100$
7	SL = 5.15m	$N_{\text{sp}} = 100$
8	SL = 5.10m	$N_{\text{sp}} = 100$

Table 3
Comparison regarding minimum spot–vehicle length ratio.

Reference no.	[4]	[28]	[38]	[41]	[42]	[43]	[44]
Minimum ratio	1.163	1.389	1.214	1.080	1.440	1.275	1.199

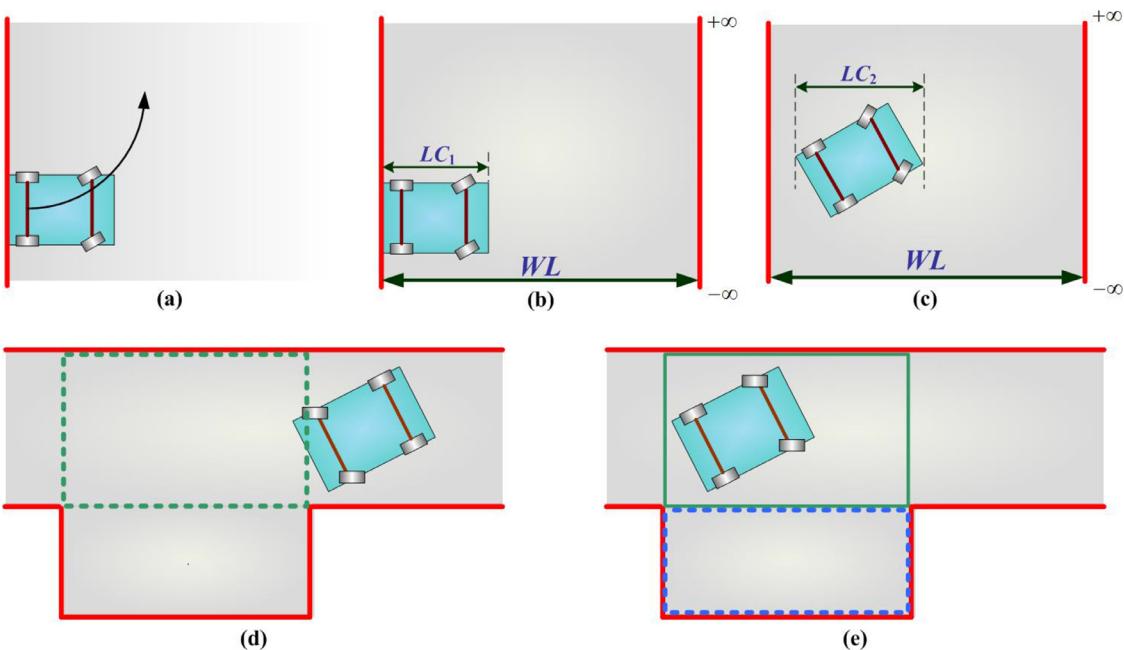


Fig. 7. Schematics on the supporting lemmas.

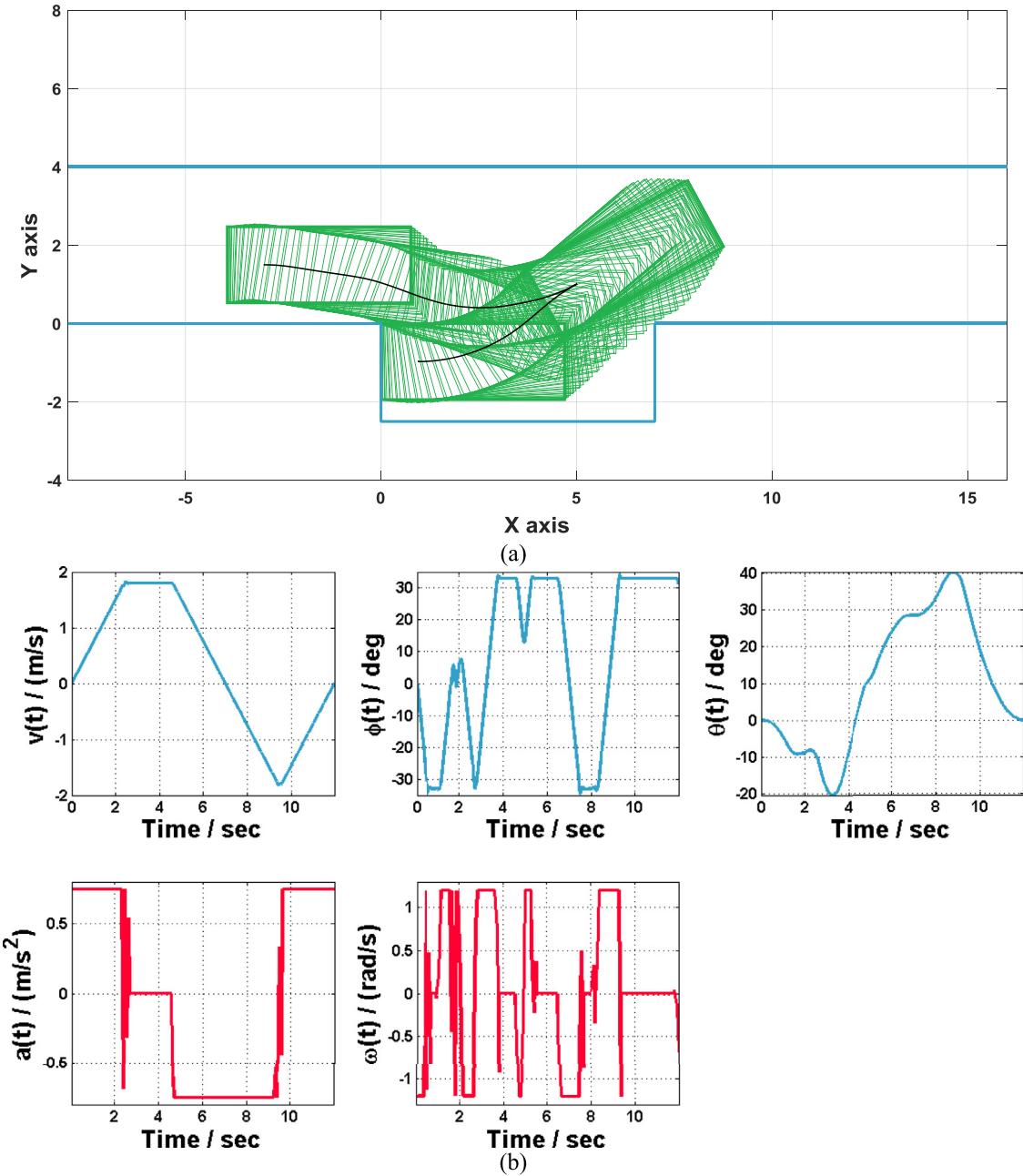


Fig. 8. Optimization results of Case 1 ($t_f = 11.97\text{s}$): (a) optimized motions; and (b) optimized control/state profiles.

generated when ϕ is fixed to a nonzero constant. Under the aforementioned circumstance, the vehicle does not collide with the obstacle barrier when it goes forward.

Lemma 2. Consider a path planning scenario depicted in Fig. 7(b), a vehicle that is adhered to the left barrier of a tube region at the beginning can arrive at any specified height provided that $LC_1 < WL$.

Lemma 3. Consider a path planning scenario depicted in Fig. 7(c), a vehicle that irregularly stops in a tube region at the beginning can arrive at any specified height provided that $LC_2 < WL$.

Lemma 4. Consider a path planning scenario depicted in Fig. 7(d), a vehicle that irregularly stops above the x axis at the beginning can be parked within the dashed box region.

Lemma 5. Consider a path planning scenario depicted in Fig. 7(e), a vehicle that irregularly stops in the solid box region at the beginning can be parked within the dashed box region.

Theorem. The spatio-temporal decomposition strategy is always active in helping find feasible parking trajectories.

5. Simulation results and discussions

Simulations were conducted in AMPL environment and executed on an Intel Core i7-4710MQ CPU that runs at 2.50 GHz. The 3.12.1 version of IPOPT (a software package of IPM) [40] was adopted. Concerned parameters and their settings are listed in Table 1. Radau collocation point settings include $\gamma_0 = 0$, $\gamma_1 =$

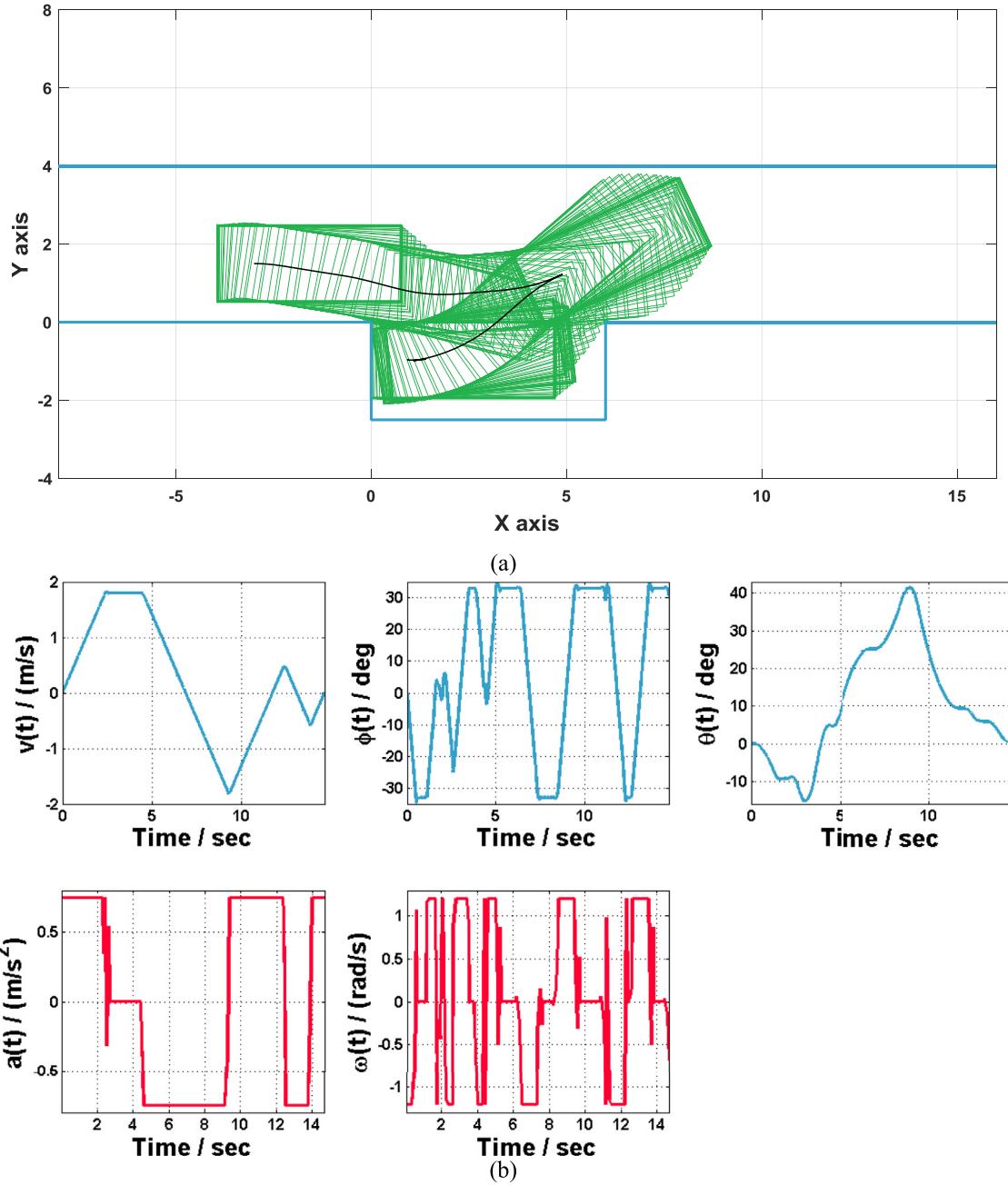


Fig. 9. Optimization results of Case 2 ($t_f = 14.66$ s): (a) optimized motions; and (b) optimized control/state profiles.

0.1551 , $\gamma_2 = 0.6450$ and $\gamma_3 = 1$ when K is set to 3 in this work.

A sequence of cases with increasing difficulty has been designed. Specifically, parking spot length changes among the cases while other conditions stay the same. The initial conditions include $x(0) = -3$, $y(0) = 1.5$, $v(0) = 0$, $\theta(0) = 0$, $\phi(0) = 0$, $a(0) = 0$, and $\omega(0) = 0$, whereas the endpoint conditions consist of $v(t_f) = 0$ and Eq. (6). Case details are listed in Table 2. Optimization results are depicted in Figs. 8–15.

5.1. Optimization results

Among the eight parking cases, the changing parameter SL decreases from 7.0 m to 5.1 m with the spot-vehicle length ratio

(defined as $SL/(m+l+n)$) decreasing from 1.493 to 1.088. From Figs. 8–15, as the spot-vehicle length ratio decreases, the parking vehicle adjusts itself locally with more attempts before it fully stops. Fig. 16 summarizes this trend: the optimization objective t_f increases more rapidly as the spot-vehicle length ratio decreases. Fig. 16 also reveals a positive correlation between t_f and the number of maneuvers. In particular the spot length in Case 8 is merely 8.8% longer than the parking vehicle; this condition is more difficult than most of the simulations reported in previous studies. Table 3 briefly reports the concerned comparison, which indicates that the present study's methodology is capable of handling relatively challenging cases. However, parking quality comparisons cannot be fairly made because different methods consider different kinematic models and/or accuracy levels.

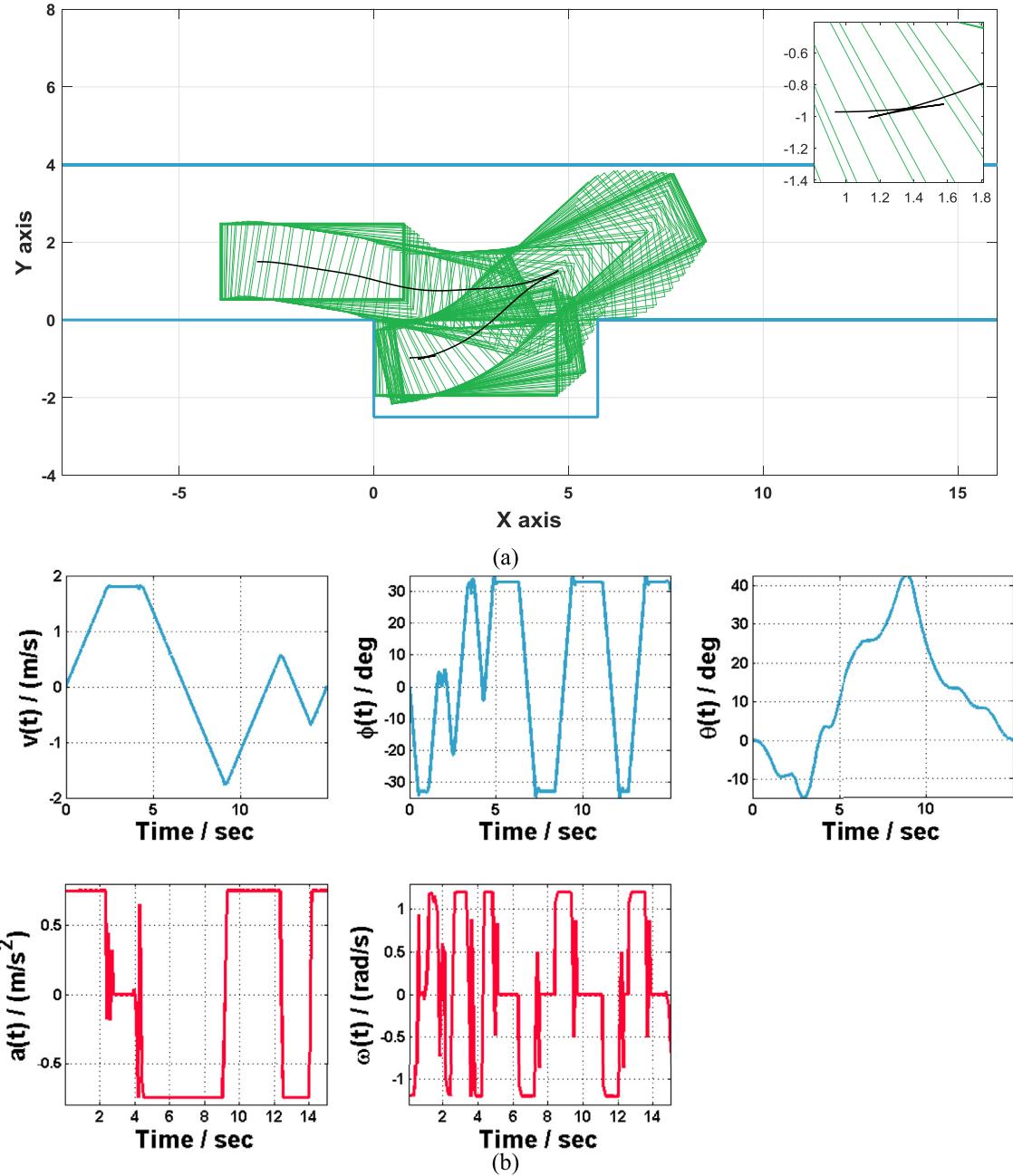


Fig. 10. Optimization results of Case 3 ($t_f = 14.99s$): (a) optimized motions; and (b) optimized control/state profiles. Terminal maneuvers are zoomed in at the upper-right corner.

5.2. Unification of the proposed methodology

Formulating a unified optimal control problem is beneficial in handling various demands without making significant changes. This subsection presents the unification of the proposed motion planner via comparative simulations. The first round of simulations concerns distinct minimization objectives. Specifically, the optimization criterion is slightly changed by adding an item $\int_0^{t_f} \phi^2(\tau)d\tau$, which accumulates the vehicle steering energies. Given that an integral cannot be directly added as part of an optimization objective, a state variable $energy(t)$ with the following equations is added to the optimal control problem:

$$\frac{denergy(t)}{dt} = \phi^2(t), \quad (14)$$

where $energy(0) = 0$. The optimization objective is set to $t_f + \xi \cdot energy(t_f)$, where ξ serves as a trade-off weight between time and steering energy. The effect of ξ on the optimization result in Case 1 is studied via a series of simulations depicted in Fig. 17. When ξ is large, the optimizer prioritizes the steering efforts to generate smoother motions at the expense of completion time. The second round of simulations involves distinct state constraints. The steering bound Φ_{max} was gradually enlarged from 0.576 to 0.800 in resolving Case 8. A series of optimization results is illustrated in Fig. 18. The obtained decreasing trend in completion time is intuitively understandable: a large Φ_{max} yields agile parking motions, which can save time. Both rounds of comparative simulations indicate that the proposed method is a unified solver, rather than one that can only tackle special cases.

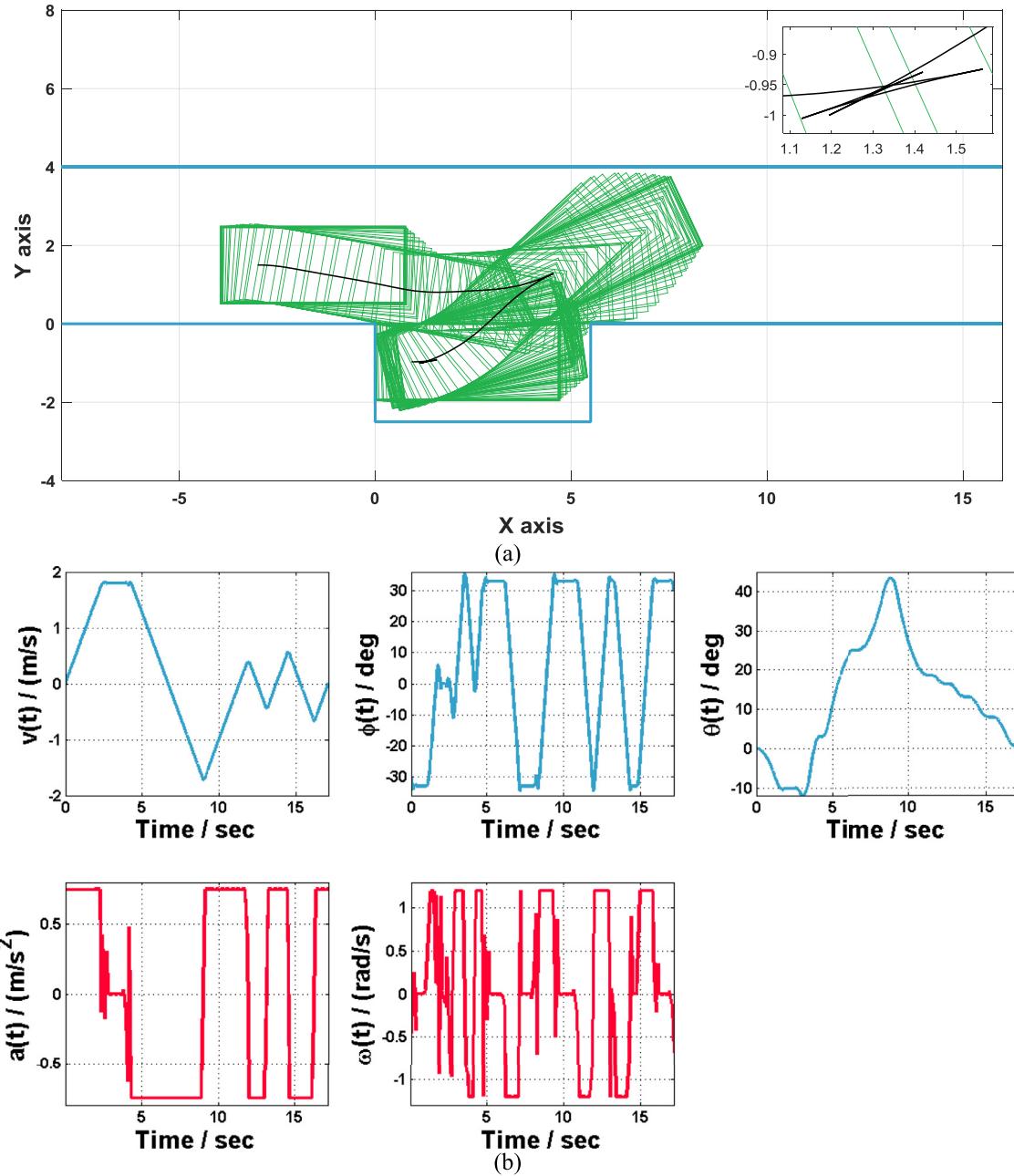


Fig. 11. Optimization results of Case 4 ($t_f = 17.13$ s): (a) optimized motions; and (b) optimized control/state profiles. Terminal maneuvers are zoomed in at the upper-right corner.

5.3. Efficiency of the proposed initialization strategy

This subsection focuses on the performance of the proposed initialization strategy when facilitating the computational difficulties. Comparative NLP-solving processes are conducted using spatio-temporal decomposition strategy and its competitors, namely, a homotopy-based backtracking method (HBM) [26] and an incremental strategy [27]. Experimental results are listed in Table 4. Herein, an NLP-solving process is considered a failure, if the optimization converges to infeasibility or does not converge within 500 iterations. According to the results shown in Table 4, the following conclusions can be made: (i) facilitating the NLP-solving process via initialization strategies is sensible, and (ii) the proposed spatio-temporal decomposition strategy is able to cope with all the

eight simulation cases, whereas HBM and the incremental strategy fail to solve relatively difficult parking problems. Both HBM and the incremental strategy are about solving simplified problems first and then gradually shifting toward the original problem. However, significant gaps may always exist between a simplified problem and the original problem; these gaps make both strategies inefficient to deal with extremely tiny parking spots. In contrast, spatio-temporal decomposition strategy simplifies the original problem by imposing additional constraints, and every pre-solved problem (with $N_X < N_{sp}$) can provide solutions that are exactly feasible to the original NLP problem! As a key advantage of spatio-temporal decomposition strategy, the iterative pre-solving process remains in the feasible region of high-dimensional solution space.

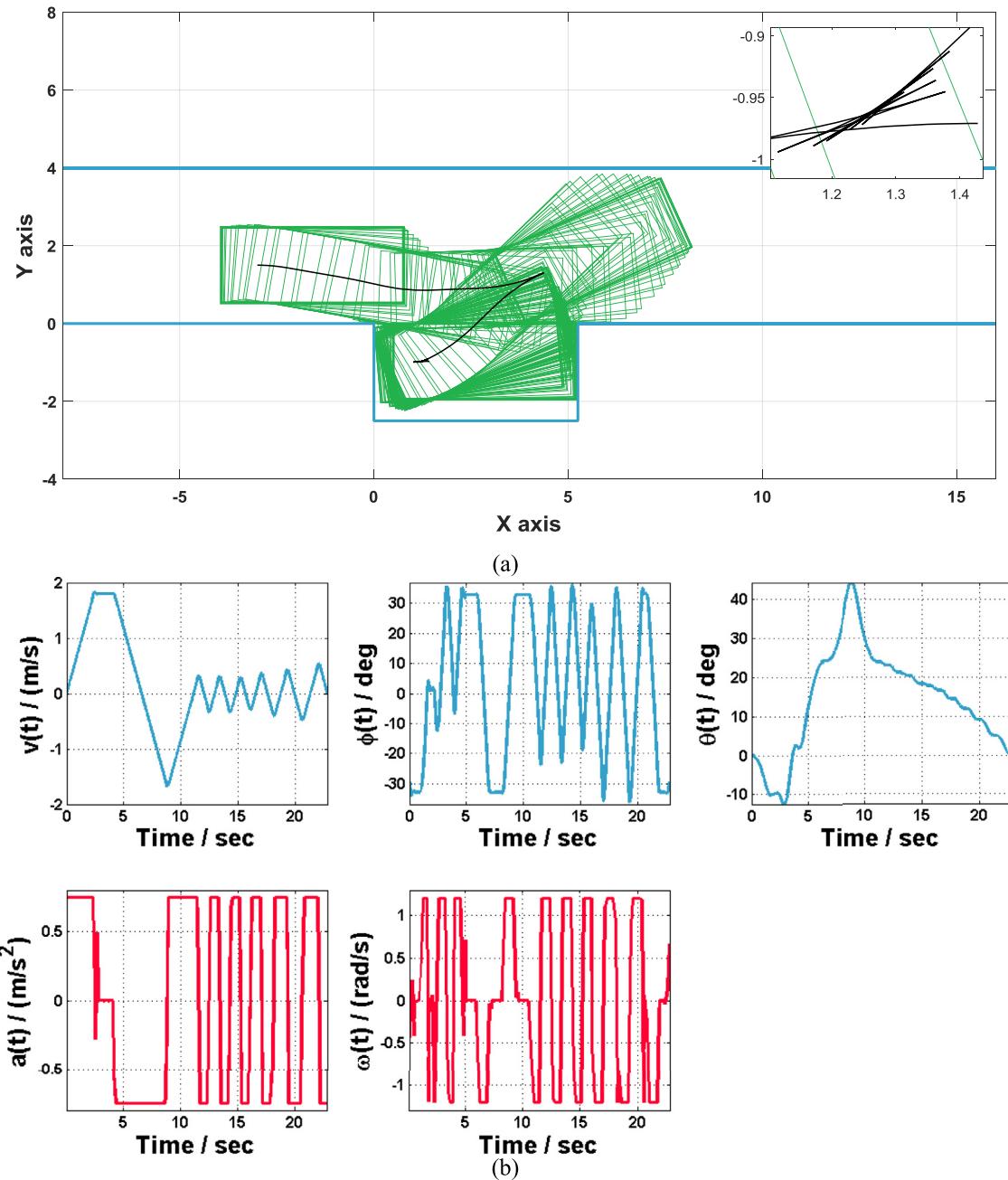


Fig. 12. Optimization results of Case 5 ($t_f = 22.82\text{s}$): (a) optimized motions; and (b) optimized control/state profiles. Terminal maneuvers are zoomed in at the upper-right corner.

Table 4
Comparisons on the performances of distinct initialization strategies.

Case No.	No initialization		HBM [26]		Incremental strategy [27]		The present study	
	CPU time	t_f	CPU time	t_f	CPU time	t_f	CPU time	t_f
1	8.25	fail	66.01	fail	45.48	11.97	39.82	11.97
2	8.47	fail	22.95	18.53	81.99	27.62	55.75	14.66
3	4.49	fail	1.22	15.96	28.22	fail	46.09	14.99
4	9.27	fail	10.82	20.63	283.48	44.18	45.76	17.13
5	3.71	fail	18.77	fail	27.47	fail	46.61	22.82
6	11.42	fail	33.95	fail	40.74	fail	71.50	27.41
7	2.43	fail	21.96	fail	91.69	fail	106.94	32.41
8	3.48	fail	27.92	fail	17.66	fail	133.95	44.66

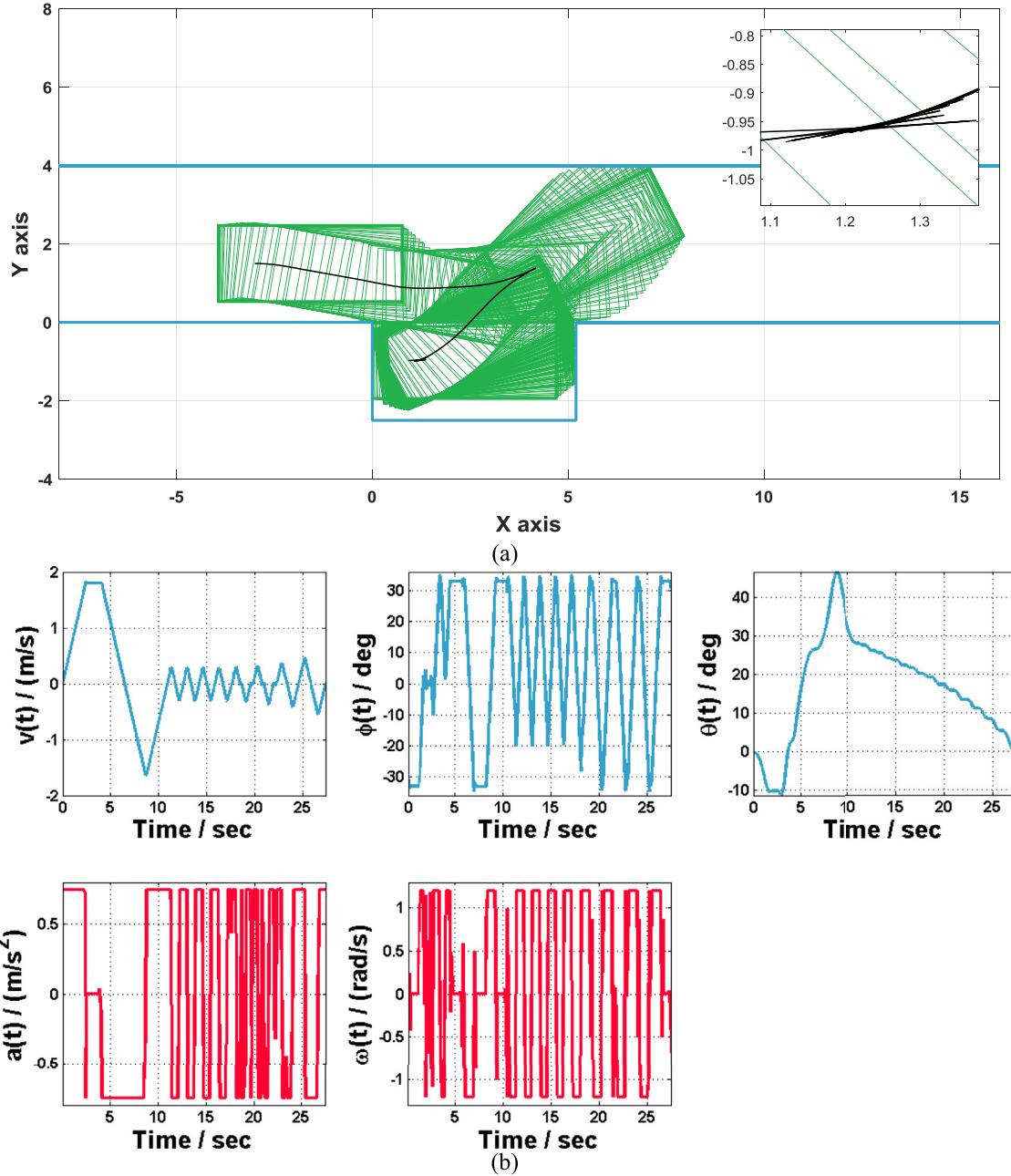


Fig. 13. Optimization results of Case 6 ($t_f = 27.41$ s): (a) optimized motions; and (b) optimized control/state profiles. Terminal maneuvers are zoomed in at the upper-right corner.

5.4. On-site motion planning performance

According to Table 4, our proposed planner spends tens of seconds to generate a trajectory each time, indicating that the planner is not suitable for on-site/on-line planning. This subsection preliminarily discusses how to improve the real-time performance of the proposed motion planner. In handling on-site planning cases, a lookup table that stores a finite number of parallel parking scenarios together with the optimized solutions is utilized [45]. When an on-site parking scheme is dispatched, the most similar scenario stored in the lookup table needs to be found, and then the corresponding solution is extracted as an initial guess when optimizing the dispatched parking mission. This study preliminarily investigates the potential of establishing a lookup table

by shifting the vehicle starting position in horizontal or vertical directions using the nominal initial guess. Simulation results are plotted in Fig. 19 on the basis of Case 4. As depicted in Fig. 19, the shifted NLP-solving processes are accomplished in less than 2 sec in general. Given that the concerned NLP problem (originates from Case 4) contains 2987 decision variables and 6568 constraints, the fact that optimums can be found in approximately 1.5 sec when tested on a personal computer is satisfactory. Particularly, when the position shifts are close to the nominal one (i.e., around the coordinate origin in Fig. 19), the CPU time lengths are relatively low. This indicates that the proposed motion planner can react to small execution noises, thereby being promising to contribute in on-line replanning frameworks (e.g., nonlinear model predictive control based methods).

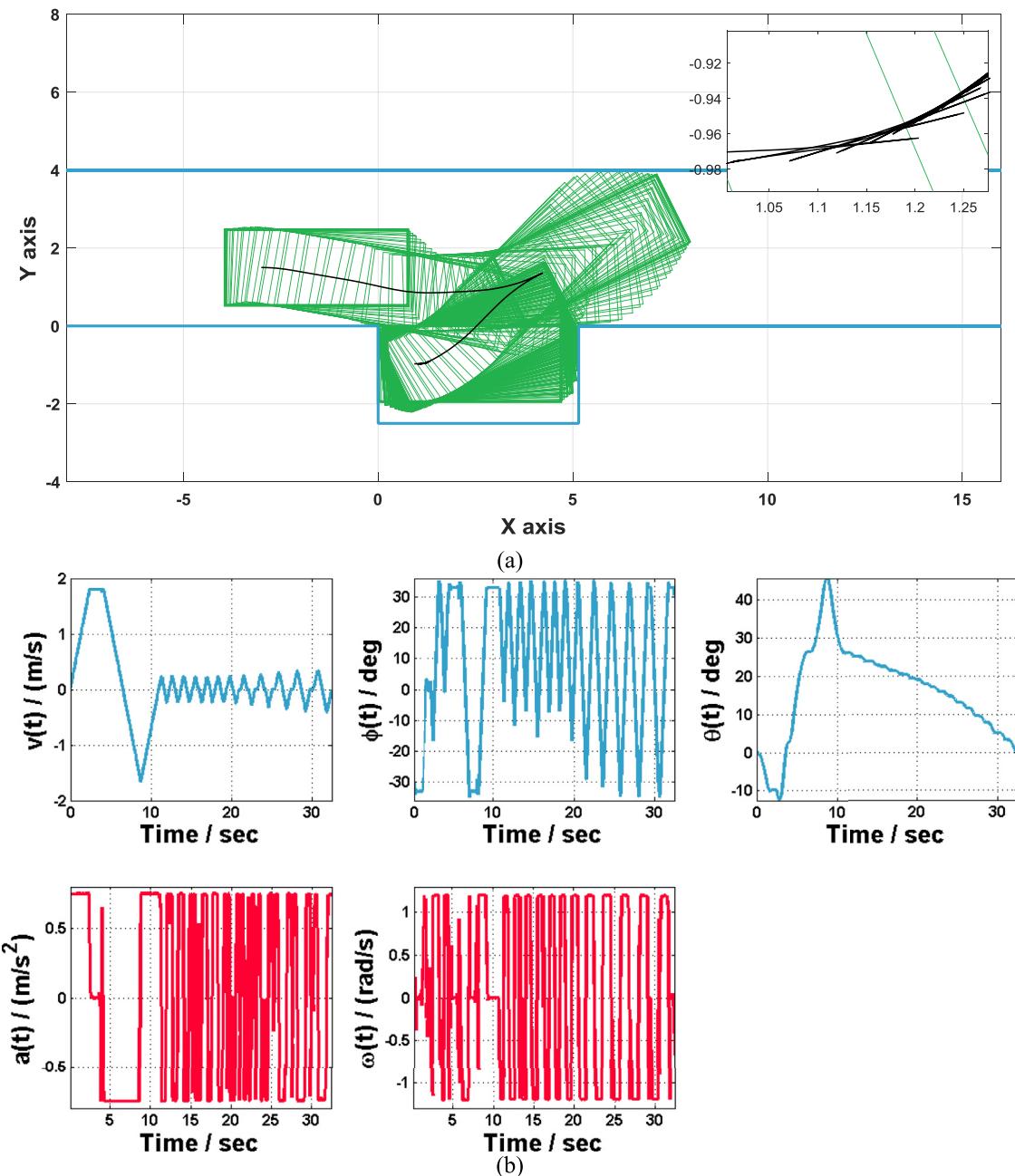


Fig. 14. Optimization results of Case 7 ($t_f = 32.41$ s): (a) optimized motions; and (b) optimized control/state profiles. Terminal maneuvers are zoomed in at the upper-right corner.

6. Conclusions

This work describes parallel parking motion planning problem in an accurate way using optimal control problem formulation. Such accurate descriptions arouse difficulties in solving the converted NLP problem. A spatio-temporal decomposition based initialization method is proposed to facilitate the NLP-solving process. Comparative simulations verify the following: (i) the proposed unified motion planner can handle a vast number of scenarios, instead of only some special cases; (ii) reasonable parking motions can be generated, even though the present study's formulation utilizes purely objective knowledge; (iii) the proposed initialization

strategy is efficient and advantageous compared with several typical competitors; and (iv) the proposed motion planner is promising in on-site planning missions. Point (ii) in particular encourages the formulation of an objective knowledge based system, which aims to fully understand and exploit vehicle maneuver potentials in parking decision making. In this sense, subjective knowledge (e.g., human experiences) should not be the terminal of computational intelligence.

Future efforts are needed to implement the planned motions on a real-world automobile platform on the basis of an on-line receding horizon control framework.

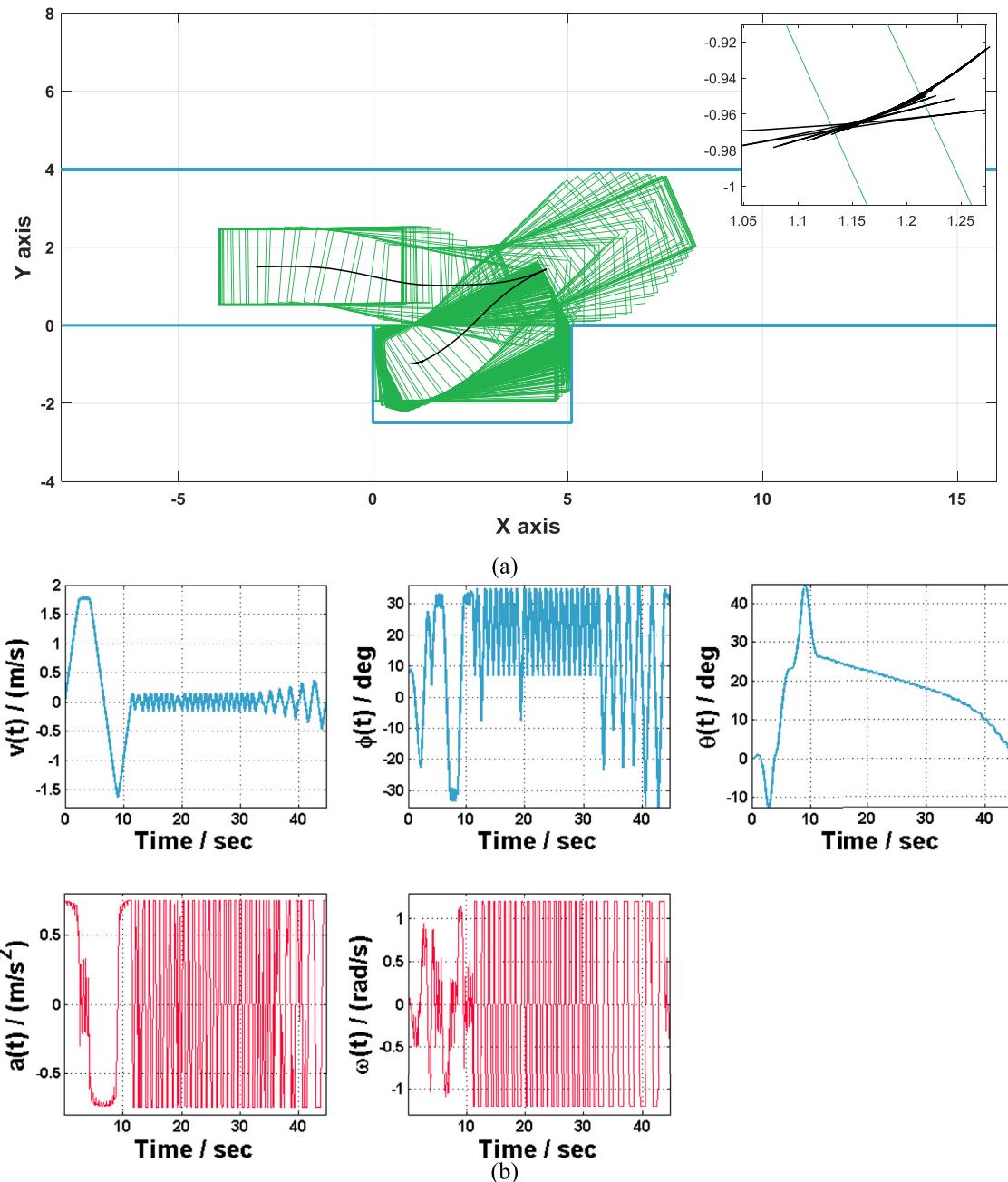


Fig. 15. Optimization results of Case 8 ($t_f = 44.66\text{s}$): (a) optimized motions; and (b) optimized control/state profiles. Terminal maneuvers are zoomed in at the upper-right corner.

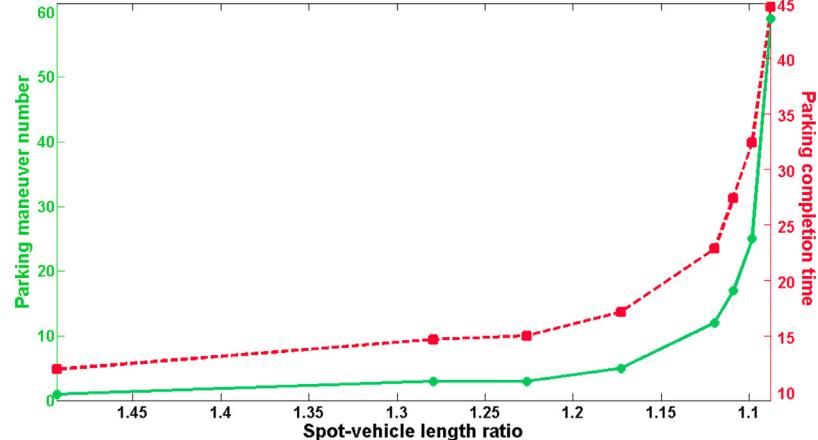


Fig. 16. Visualization of serial optimization results. Note that the x axis represents the spot-vehicle length ratio in descending order. The corresponding number of maneuvers and completion time lengths represent the y axes on both sides.

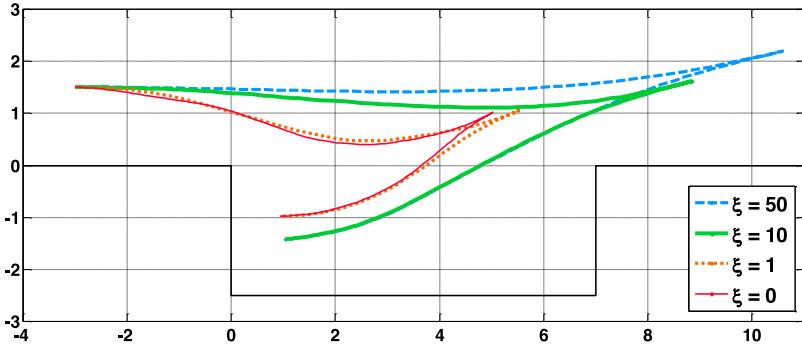


Fig. 17. Comparative simulations with various weights ξ in the minimization objective $t_f + \xi \cdot \text{energy}(t_f)$ in Case 1. Note that smooth motions are obtained when ξ is large, and long completion time is required as a consequence. Specifically, $t_{f(\xi=0)} = 11.97\text{s}$, $t_{f(\xi=1)} = 12.47\text{s}$, $t_{f(\xi=10)} = 16.09\text{s}$, and $t_{f(\xi=50)} = 18.62\text{s}$.

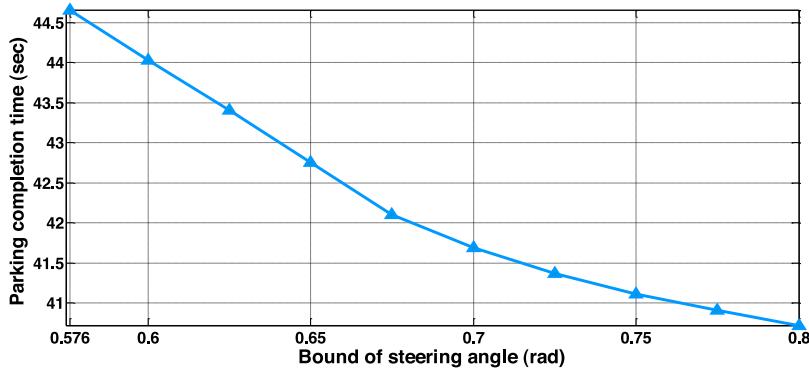


Fig. 18. Comparative simulations with various steering bounds Φ_{\max} in Case 8. Note that a large Φ_{\max} results in a short parking completion time.

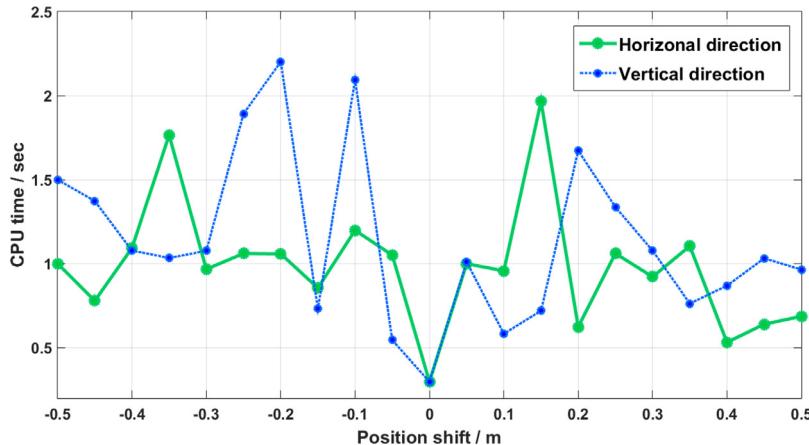


Fig. 19. Comparative simulations with shifted vehicle starting locations in Case 4.

Acknowledgements

The authors are sincerely grateful to the five anonymous reviewers, Prof. Lorenz T. Biegler, and Prof. Pu Li for their valuable comments and suggestions, as well as to Prof. Hamido Fujita for the efficient handling of this manuscript. Bai Li thanks Ms. Yanfen Zeng for the inspiration during this research study. This work is supported by the 973 Program of China under Grant 2012CB720503, National Nature Science Foundation under Grant 61374167, and College Students' Innovation & Technical Training Program of Zhejiang Province (Xinmiao Talent Program) under Grant 2016R401239.

To assist understanding, a video that contains the simulation results of Cases 1–8 is available at http://www.tudou.com/programs/view/hRE1Z_oOHVA/. Source codes of this study are available at <https://www.researchgate.net/publication/303913791>.

Appendix

Proof of **Lemma 1**. Let us first consider an automobile that goes forward with $\phi(t) \equiv -\phi_0 < 0$. Eq. (1) yields the following equations when paths (rather than trajectories) are investigated.

$$\begin{cases} \frac{dx}{ds} = \frac{dx}{dt} \frac{dt}{ds} = v \cdot \cos \theta \cdot \frac{1}{v} = \cos \theta \\ \frac{dy}{ds} = \frac{dy}{dt} \frac{dt}{ds} = v \cdot \sin \theta \cdot \frac{1}{v} = \sin \theta \\ \frac{d\theta}{ds} = \frac{d\theta}{dt} \frac{dt}{ds} = \frac{v \cdot \tan \phi}{l} \cdot \frac{1}{v} = \frac{-\tan \phi_0}{l} \end{cases} \quad (\text{A1})$$

Note that $\theta(s)|_{s=0} = 0$, thus $\theta(s) = -\frac{\tan \phi_0}{l} \cdot s$. Substituting $\theta(s)$ into the first equation regarding $\frac{dx}{ds}$ in Eq. (A1) and calculating the

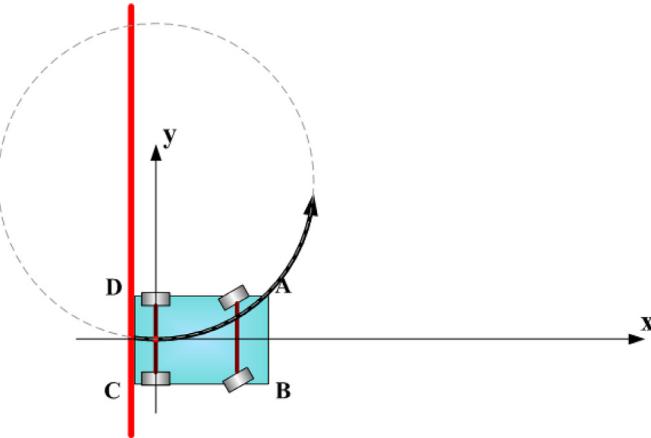


Fig. A1. Schematic on Lemma 1.

integral, we have

$$x(s) = x(s)|_{s=0} + \frac{l}{\tan \phi_0} \cdot \sin \left(\frac{\tan \phi_0}{l} \cdot s \right). \quad (\text{A2})$$

Similarly,

$$y(s) = y(s)|_{s=0} + \frac{l}{\tan \phi_0} \cdot \left[1 - \cos \left(\frac{\tan \phi_0}{l} \cdot s \right) \right]. \quad (\text{A3})$$

According to Eqs. (A2) and (A3), $x(s)$ and $y(s)$ contribute to a circular arc because $[x(s) - x(s)|_{s=0}]^2 + [y(s) - y(s)|_{s=0} - \frac{l}{\tan \phi_0}]^2 = (\frac{l}{\tan \phi_0})^2$. We refer to the constant circular arc radius as $R = \frac{l}{\tan \phi_0} > 0$.

For the convenience, we establish a coordinate system like the one depicted in Fig. A1. This setting is beneficial in that $x(s)|_{s=0} = y(s)|_{s=0} = 0$, which simplifies Eqs. (A2) and (A3) as

$$\begin{cases} x(s) = R \cdot \sin \left(\frac{s}{R} \right) \\ y(s) = R - R \cdot \cos \left(\frac{s}{R} \right) \end{cases}. \quad (\text{A4})$$

Let us focus on the process that $s \in [0, \alpha \cdot R]$ where $0 < \alpha \leq \frac{\pi}{2}$. The path of the corner point D on the automobile (see Fig. A1) can be described as

$$\begin{cases} D_x(s) = (R - b) \cdot \sin \left(\frac{s}{R} \right) - m \cdot \cos \left(\frac{s}{R} \right) \\ D_y(s) = R - (R - b) \cdot \cos \left(\frac{s}{R} \right) - m \cdot \sin \left(\frac{s}{R} \right) \end{cases}. \quad (\text{A5})$$

To avoid collision, $D_x(s) \geq -m$ should hold true constantly when $s \in [0, \alpha \cdot R]$. $D_x(s)$ can be further simplified via a trigono-

metric formula:

$$D_x(\tau) = \sqrt{(R - b)^2 + m^2} \cdot \sin(\tau - \beta), \quad (\text{A6})$$

where $\beta = \arctan(\frac{m}{R-b}) \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ and $\tau \in [0, \alpha]$. This indicates that $D_x(\tau) \geq -m$ when $\tau \in [0, \alpha]$, and $\alpha \in (0, \frac{\pi}{2}]$. Since $D_x(s) \geq -m$, we conclude that the corner point D does not collide with the left barrier when it follows a circular arc path. The other three corner points (i.e., A, B and C) as well as points along the vehicle margins can be analyzed in a similar way. Thus the concerned vehicle would not collide with the barrier. \square

Proof of Lemma 2. Let us consider a specific kind of motions with fixed steering angles. According to Lemma 1, this kind of motions would result in circular arc-shaped paths. In more detail, we require that $\phi \equiv -\phi_0$ from the beginning until $\theta = \arcsin(\frac{WL-LC_1}{2R})$; thereafter, ϕ is fixed to ϕ_0 until the car stops with $\theta = 0$ again. Given that Lemma 1 indicates no collisions occur during such motions, the vehicle would successfully reach a higher location that is parallel to the original one (see Fig. A2(a)). By simply reversing the motions, the vehicle arrives at location N from M as illustrated in Fig. A2(b). Location N is higher than the original L and the increment in height Δh is $4R - 4\sqrt{R^2 - (\frac{WL-LC_1}{2})^2}$, which is rewritten as

$$\Delta h = \frac{4(WL - LC_1)^2}{R + \sqrt{R^2 - (\frac{WL-LC_1}{2})^2}}. \quad (\text{A7})$$

When $|\phi_0| \rightarrow 0_+$, $R \rightarrow +\infty$, thus $\Delta h \rightarrow 0_+$. This indicates that this specific type of motions would lead the vehicle to any (small) increment along the Y axis. Conversely, since $|\phi_0| \leq \Phi_{\max}$, then $R \geq \frac{l}{\tan \Phi_{\max}}$, thereby Δh has the supremum $\frac{4 \cdot (WL - LC_1)^2}{\frac{l}{\tan \Phi_{\max}} + \sqrt{\frac{l^2}{\tan^2 \Phi_{\max}} - (\frac{WL-LC_1}{2})^2}}$. Repeating this kind of motion would lead the vehicle to any (large) increment along the Y axis.

To conclude, any user-specified height in the tube region depicted in Fig. 7(b) can be achieved. \square

Proof of Lemma 3. The scenario in Lemma 2 is a special case of that in Lemma 3. By going backward, the vehicle can reach the location with (at least) one corner point adhered to the left barrier (see Fig. A3(a)). If the vehicle moves within the shaded region in Fig. A3(b), the scheme becomes that analyzed in Lemma 2. Through the aforementioned motions with fixed steering angles, a location depicted in Fig. A3(c) is reached. Through going forward further, the vehicle can reach the location with (at least) one corner point adhered to the right barrier. By simply reversing these motions, the vehicle can arrive at location N from M as illustrated

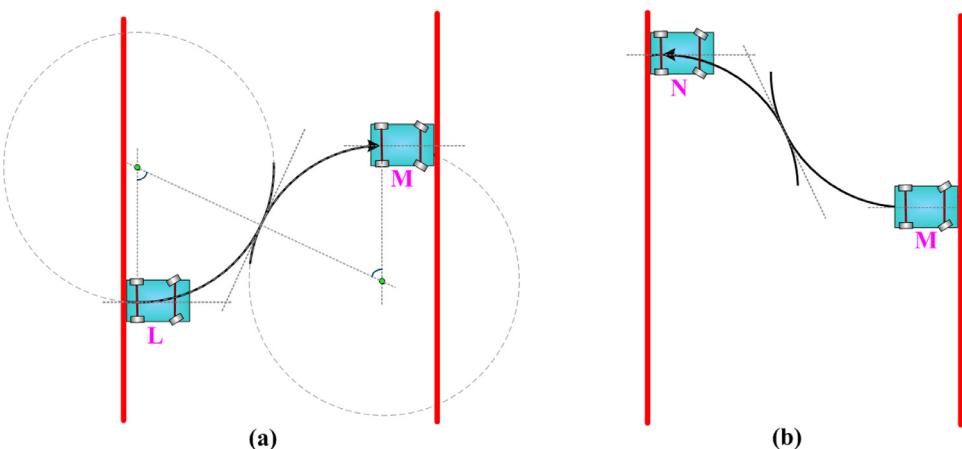


Fig. A2. Schematic on Lemma 2.

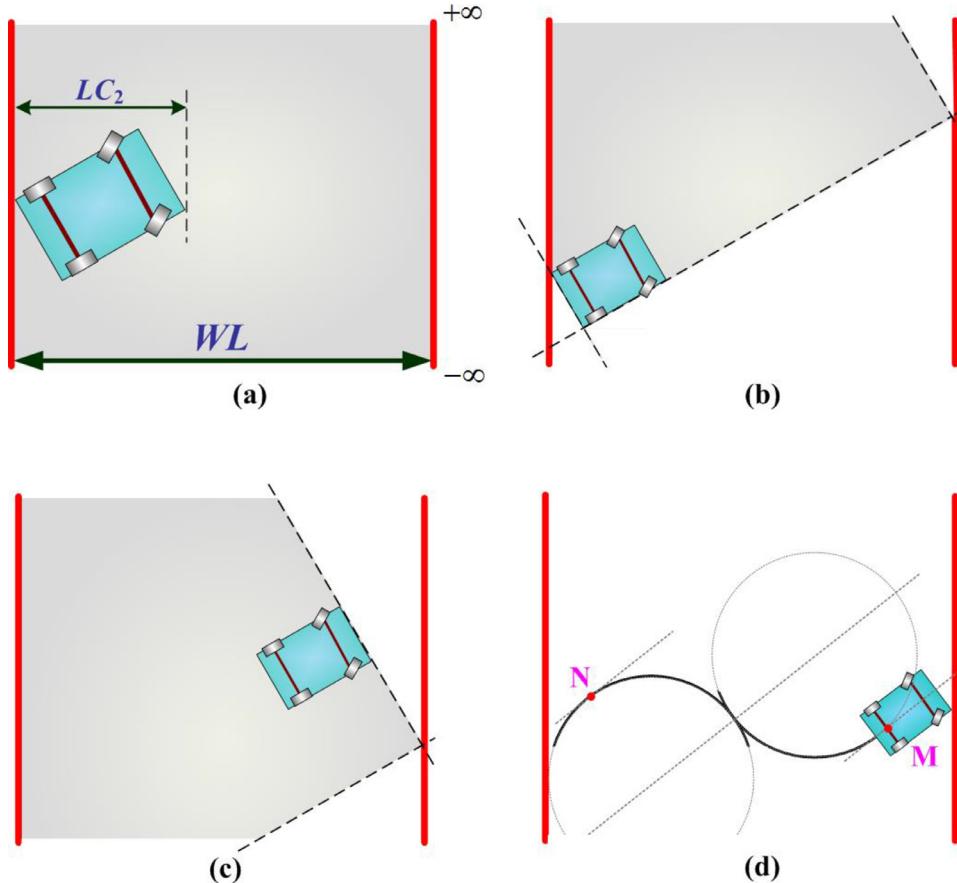


Fig. A3. Schematic on Lemma 3.

in Fig. A3(d). Location N is higher than the original one. Repeating these motions would lead the vehicle to any user-specified height. \square

Proof of Lemma 4. Implementing a rotation of ninety degrees on the concerned scenario yields the scenario that has been analyzed in Lemma 3. Given that the vehicle can reach any specified height (according to Lemma 3), it can finally be parked into the dashed box region in Fig. 7(d). \square

Proof of Lemma 5. This is evident following the analysis for Lemma 4. \square

Proof of Theorem. Combining Lemma 4 and Lemma 5 renders a conclusion that there always exist paths that can guide an vehicle into the parking slot regardless of where it is originally located. Since the vehicle can never enter the parking spot without involving the dashed box region in Fig. 7(d), there exists a moment $0 \leq t_{\text{std}} < t_f$ such that the vehicle remains in the critical region (defined in Eq. (6)) from the moment t_{std} to t_f . Thus, there always exists one feasible path that can lead the vehicle into the parking slot when spatio-temporal decomposition strategy is utilized.

On the other hand, the motions associated with our concerned paths involve temporarily fixed steering angles. This causes the velocity $v(s)$ to not influence the generation of a path at all (as can be noticed in Eqs. (A2) and (A3)). Thus attaching an arbitrary time course along the concerned path would result in one feasible trajectory. Given that Eq. (1) composes of elementary functions, the particular trajectory we obtain (via fixed steering angles) is not the only one that can be found using spatio-temporal decomposition strategy. In other words, a number of feasible trajectories can be obtained via spatio-temporal decomposition strategy without particularly fixing the steering angles at all. Through this, we conclude that the spatio-temporal decomposition strategy is always active to help find feasible parallel parking trajectories. \square

References

- [1] J.F. Pereira, M.S. Thesis, Autonomous parking using 3d perception, University of Aveiro, Aveiro, Portugal, 2012.
- [2] I. Bogoslavskyi, L. Spinello, W. Burgard, C. Stachniss, Where to park? minimizing the expected time to find a parking space, in: Proceedings of 2015 IEEE International Conference on Robotics and Automation, ICRA 2015, 2015, pp. 2147–2152.
- [3] D.G. Kidd, A.T. McCartt, Differences in glance behavior between drivers using a rearview camera, parking sensor system, both technologies, or no technology during low-speed parking maneuvers, Accid. Anal. Prev. 87 (2016) 92–101.
- [4] H. Vorobieva, S. Glaser, N. Minoiu-Enache, S. Mammar, Automatic parallel parking in tiny spots: path planning and control, IEEE Trans. Intell. Transp. Syst. 16 (1) (2015) 396–410.
- [5] B. Reimer, B. Mehler, J.F. Coughlin, Reductions in self-reported stress and anticipatory heart rate with the use of a semi-automated parallel parking system, Appl. Ergonom. 52 (2016) 120–127.
- [6] W. Wang, Y. Song, J. Zhang, H. Deng, Automatic parking of vehicles: A review of literatures, Int. J. Automot. Technol. 15 (6) (2014) 967–978.
- [7] C. Chuang, C. Huang, L. Ko, C. Lin, An EEG-based perceptual function integration network for application to drowsy driving, Knowl.-Based Syst. 80 (2015) 143–152.
- [8] M.M. Waldrop, Autonomous vehicles: no drivers required. Available online: <http://www.nature.com/news/autonomous-vehicles-no-drivers-required-1.16832>. Feb. 2015.
- [9] W.Y. Kwon, I.H. Suh, Planning of proactive behaviors for human-robot cooperative tasks under uncertainty, Knowl.-Based Syst. 72 (2014) 81–95.
- [10] C. Katrakazas, M. Quddus, W. Chen, L. Deka, Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions, Transp. Res. Part C 60 (2015) 416–442.
- [11] C. Siedentop, V. Laukart, B. Krastev, D. Kasper, A. Wedel, G. Breuel, C. Stachniss, in: Autonomous parking using previous paths, In Advanced Microsystems for Automotive Applications, Springer International Publishing, 2015, pp. 3–14.
- [12] G. Notomista, M. Botsch, Maneuver segmentation for autonomous parking based on ensemble learning, in: Proceedings of 2015 International Joint Conference on Neural Networks, IJCNN 2015, 2015, pp. 1–8.
- [13] X. Du, K.K. Tan, Autonomous reverse parking system based on robust path generation and improved sliding mode control, IEEE Trans. Intell. Transp. Syst. 16 (3) (2015) 1225–1237.

- [14] A.N. Chand, M. Kawanishi, T. Narikiyo, Fast parallel parking for autonomous vehicles using Gompertz curves, in: Proceedings of 2014 11th International Conference on Ubiquitous Robots and Ambient Intelligence, URAI 2014, 2014, pp. 572–578.
- [15] C. Chen, M. Rickert, A. Knoll, Path planning with orientation-aware space exploration guided heuristic search for autonomous parking and maneuvering, in: Proceedings of 2015 IEEE Intelligent Vehicles Symposium, IV 2015, 2015, pp. 1148–1153.
- [16] C. Sprunk, B. Lau, P. Pfaff, W. Burgard, An accurate and efficient navigation system for omnidirectional robots in industrial environments, *Autonom. Robots* (2016) 1–21.
- [17] B. Li, Z. Shao, Precise trajectory optimization for articulated wheeled vehicles in cluttered environments, *Adv. Eng. Softw.* 92 (2016) 40–47.
- [18] Z. Shiller, Offline and on-line trajectory planning, in: *Motion and Operation Planning of Robotic Systems*, Springer International Publishing, 2015, pp. 29–62.
- [19] D.G. Hull, Conversion of optimal control problems into parameter optimization problems, *J. Guidance, Control Dyn.* 20 (1) (1997) 57–60.
- [20] B. Li, Z. Shao, A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles, *Knowl.-Based Syst.* 86 (2015) 11–20.
- [21] H.R. Marzban, S.M. Hoseini, An efficient discretization scheme for solving nonlinear optimal control problems with multiple time delays, *Optimal Control Appl. Methods* (2015) 1–26.
- [22] B. Tian, W. Fan, Q. Zong, Integrated guidance and control for reusable launch vehicle in reentry phase, *Nonlinear Dyn.* 80 (1–2) (2015) 397–412.
- [23] R.B. Patel, P.J. Goulart, Trajectory generation for aircraft avoidance maneuvers using online optimization, *J. Guidance Control Dy.* 34 (1) (2011) 218–230.
- [24] B.H. Li, C.T. Chang, A simple and efficient initialization strategy for optimizing water-using network designs, *Ind. Eng. Chem. Res.* 46 (25) (2007) 8781–8786.
- [25] S.M. Safrannejad, J.D. Hedengren, N.R. Lewis, E.L. Haseltine, Initialization strategies for optimization of dynamic systems, *Comput. Chem. Eng.* 78 (2015) 39–50.
- [26] L. Zhu, Z. Chen, X. Chen, Z. Shao, J. Qian, Simulation and optimization of cryogenic air separation units using a homotopy-based backtracking method, *Separation Purification Technol.* 67 (3) (2009) 262–270.
- [27] B. Li, Z. Shao, An incremental strategy for tractor-trailer vehicle global trajectory optimization in the presence of obstacles, in: Proceedings of IEEE International Conference on Robotics and Biomimetics, ROBIO 2015, 2015, pp. 1447–1452.
- [28] J. Moon, I.B. Jaeyoung, J. Cha, S. Kim, A trajectory planning method based on forward path generation and backward tracking algorithm for automatic parking systems, in: Proceedings of 2014 IEEE 17th International Conference on Intelligent Transportation Systems, ITSC 2014, 2014, pp. 719–724.
- [29] P. Fiorini, Z. Shiller, Time optimal trajectory planning in dynamic environments, in: Proceedings of 1996 IEEE International Conference on Robotics and Automation, ICRA 1996, vol. 2, 1996, pp. 1553–1558.
- [30] K. Kant, S.W. Zucker, Toward efficient trajectory planning: the path-velocity decomposition, *Int. J. Robot. Res.* 5 (3) (1986) 72–89.
- [31] G. Wagner, H. Choset, Subdimensional expansion for multirobot path planning, *Artif. Intell.* 219 (2015) 1–24.
- [32] C. Ferner, G. Wagner, H. Choset, ODRM*: optimal multirobot path planning in low dimensional search spaces, in: Proceedings of 2013 IEEE International Conference on Robotics and Automation, ICRA 2013, 2013, pp. 3854–3859.
- [33] F. You, R. Zhang, G. Lie, H. Wang, H. Wen, J. Xu, Trajectory planning and tracking control for autonomous lane change maneuver based on the cooperative vehicle infrastructure system, *Exp. Syst. Appl.* 42 (14) (2015) 5932–5946.
- [34] Y. Zhuang, H. Huang, Time-optimal trajectory planning for underactuated spacecraft using a hybrid particle swarm optimization algorithm, *Acta Astronautica* 94 (2) (2014) 690–698.
- [35] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, P. Abbeel, Motion planning with sequential convex optimization and convex collision checking, *Int. J. Robot. Res.* 33 (9) (2014) 1251–1270.
- [36] N. Dadkhah, B. Mettler, Survey of motion planning literature in the presence of uncertainty: considerations for UAV guidance, *J. Intell. Robot. Syst.* 65 (1–4) (2012) 233–246.
- [37] L.T. Biegler, X. Yang, G. Fischer, Advances in sensitivity-based nonlinear model predictive control and dynamic real-time optimization, *J. Process Control* 30 (2015) 104–116.
- [38] 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.
- [39] L.T. Biegler, S. Campbell, V. Mehrmann, Control and optimization with differential-algebraic constraints, Philadelphia (2012).
- [40] A. Wächter, L.T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Math. Programm.* 106 (1) (2006) 25–57.
- [41] P. Zips, M. Böck, A. Kugi, Optimisation based path planning for car parking in narrow environments, *Robot. Autonom. Syst.* 79 (2016) 1–11.
- [42] 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.
- [43] Z. Lv, L. Zhao, Z. Liu, A Path-Planning Algorithm for Automatic Parallel Parking, in: Proceedings of 2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control, IMCCC 2013, 2013, pp. 474–478.
- [44] K. Cheng, Y. Zhang, H. Chen, Planning and control for a fully-automatic parallel parking assist system in narrow parking spaces, in: Proceedings of 2013 IEEE Intelligent Vehicles Symposium, IV 2013, 2013, pp. 1440–1445.
- [45] B. Li, Z. Shao, Time-optimal maneuver planning in automatic parallel parking using a simultaneous dynamic optimization approach, *IEEE Trans. Intell. Transp. Syst.* (2016) in press, doi:10.1109/TITS.2016.2546386.