

複雑な環境における自律駐車のための軌道計画：トンネルベースの最適制御アプローチ

Bai Li, Tankut Acarman, Qi Kong, and Youmin Zhang

概要-本論文では、自律駐車のための高速かつ高精度な軌道計画アルゴリズムを提案する。公称では、この方式を記述するために最適制御問題を定式化する必要があるが、最適制御問題の次元は通常大きく、車両は動的プロセス全体を通してすべての瞬間にすべての障害物との衝突を避ける必要があるからである。サンプルアンドサーチベースのプランナによって得られた初期推測は、数値最適化プロセスを容易にするが、リアルタイムというほど高速にはまだ遠い。この問題に対処するため、衝突回避制約をすべて一連のトンネル内条件に置き換える。具体的には、車両が障害物から自然に分離するトンネル内を移動するように制限されたトンネルベースの戦略を開発する。提案する軌道計画手法の統一性、効率性、ロバスト性はシミュレーションにより検証されている。

I. INTRODUCTION

A. Background

自律走行車技術は、都市交通に革命的な変化をもたらしている[1]。自律走行システムに必要なモジュールとして、軌道計画は、車両にとって運動学的に実現可能で、乗客にとって快適で、検出された障害物から衝突のない軌道を生成することである[2]。本論文では、自律駐車シナリオにおける軌道計画に焦点を当てる。

駐車場の軌道計画アルゴリズムは、路上走行のアルゴリズムよりも困難である。なぜなら、(i)ナビゲーションのための基準線がない、(ii)車両の運動学は後方走行をサポートすべきである、(iii)環境の複雑な障害物は問題定式化を複雑にしている、からである。これらの要因により、一般的な路上軌道プランナーは駐車タスクに直接適用できない。

B. 関連作品

大まかに言えば、自律駐車スキームを扱うことができる軌道プランナーは、サンプルアンドサーチベースの手法と最適化ベースの手法に分類できる。サンプルアンドサーチベースのプランナは、まず連続状態空間をグラフとして抽象化し、次にグラフから開始とゴールの構成をつなぐ満足のいくノードを検索する。

このカテゴリは、状態空間または入力空間をサンプリングすることで、さらに分割することができる。典型的な状態空間サンプラーには、状態格子プランナー[3]やRapidly-exploring Random Tree (RRT)ファミリー[4]がある。よく知られた入力空間サンプラーには、ハイブリッドA*アルゴリズム[5]や動的窓アプローチ(DWA)[6]がある。一方、最適化に基づく方法は、関係する軌道計画スキームを最適制御問題として記述し、それを数値的に解く。数値解は、最適制御問題を非線形計画法(NLP)問題に変換し、そのNLPを解くことによって導かれる。シーケンス二次計画法(SQP)[7,8]、内点法(IPM)[9]、凸実行可能集合アルゴリズム[10]、 g^2_0 [11]などのNLPソルバーが、駐車指向の軌道計画問題に適用されている。

サンプルアンドサーチベースのプランナと比較して、最適制御問題を定式化することは、(i)連続状態空間をプリミティブに離散化する必要がない、(ii)軌道速度分解を行わずに軌道を直接計画できる、などの点で有利である。しかし、最適化ベースのプランナの副作用は計算負荷が大きいことである。一般に、最適制御問題には多数の非凸衝突回避制約が含まれており、オンラインアプリケーションの妨げとなっている。この制限に対する一般的な解決策は、サンプルと探索に基づくプランナを利用して、粗い経路/軌道を素早く見つけ、その初期推測で数値最適化を実装することです[11-13]。素早く探索された初期推測は数値解法プロセスを容易にするが、定式化された最適制御問題には大規模な非凸制約がまだ存在し、初期推測が最適に近くないときはいつでも最適化を遅くする。したがって、初期化品質を維持することに加え、最適制御問題の定式化を簡略化するための多大な努力が必要である。

関連する最適制御問題における主な困難は、衝突回避制約にある[14]。車両は環境中の全ての障害物に衝突する機会を持たない可能性があるため、特に粗い経路/軌道が与えられた場合、衝突回避制約の一部を安全に除去することができる(図1(a))。この考え方に従えば、周囲の障害物すべてから車両を隔てるトンネルを舗装することを考えるのが自然である。このようなトンネルを手に入れば、トンネル内制約を使って衝突回避制約を完全に置き換えることができる。これにより、最適制御問題の次元性と非凸性は、最適制御問題の複雑さから独立することになる。

* Research supported by the National Key R&D Program of China under Grant 2018YFB1600804, and the Natural Sciences and Engineering Research Council of Canada.

Bai Li is with the JDX R&D Center of Automated Driving, JD Inc., Beijing, China (e-mail: libai@zju.edu.cn, libai1@jd.com).

Tankut Acarman is with the Computer Engineering Department, Galatasaray University, Istanbul, Turkey (e-mail: tacarman@gsu.edu.tr).

Qi Kong is with the JDX R&D Center of Automated Driving, JD Inc., Beijing, China (e-mail: Qi.Kong@jd.com).

Youmin Zhang is with the Department of Mechanical, Industrial and Aerospace Engineering, Concordia University, Montreal, Canada (e-mail: ymzhang@encs.concordia.ca).

Trajectory Planning for Autonomous Parking in Complex Environments: A Tunnel-based Optimal Control Approach *

Bai Li, Tankut Acarman, Qi Kong, and Youmin Zhang

Abstract—This paper proposes a fast and accurate trajectory planning algorithm for autonomous parking. Nominally, an optimal control problem should be formulated to describe this scheme, but the dimensionality of the optimal control problem is usually large, because the vehicle needs to avoid collision with every obstacle at every moment during the entire dynamic process. Although an initial guess obtained by a sample-and-search based planner facilitates the numerical optimization process, it is still far from being as fast as real-time. To address this issue, we replace all of the collision-avoidance constraints by series of within-tunnel conditions. Concretely, we develop a tunnel-based strategy such that the vehicle is restricted to move within the tunnels which naturally separate the vehicle from the obstacles. Unification, efficiency, and robustness of the proposed trajectory planning method have been verified by simulations.

I. INTRODUCTION

A. Background

Autonomous vehicle technologies are bringing about revolutionary changes to the urban transport [1]. As a necessary module in an autonomous driving system, trajectory planning is about generating a trajectory that is kinematically feasible for the vehicle, comfortable for the passengers, and collision-free from the detected obstacles [2]. This paper focuses on trajectory planning in the autonomous parking scenarios.

Trajectory planning algorithms for parking are more challenging than the ones for on-road driving, because (i) there is not a reference line for navigation; (ii) the vehicle kinematics should support driving backwards; and (iii) the intricate obstacles in the environment complicate the problem formulation. These factors make the prevalent on-road trajectory planners not directly applicable to the parking tasks.

B. Related Works

Broadly speaking, the trajectory planners that are capable of handling autonomous parking schemes can be classified as sample-and-search-based, and optimization-based methods. A sample-and-search-based planner first abstracts the

continuous state space as a graph, then searches in the graph for satisfactory nodes that link the starting and goal configurations. This category can be further divided by sampling the state space or the input space. Typical state-space samplers include the state lattice planner [3] and the Rapidly-exploring Random Tree (RRT) families [4]. Well-known input-space samplers include the hybrid A* algorithm [5] and Dynamic Window Approach (DWA) [6]. Optimization-based methods, on the other hand, describe the concerned trajectory planning scheme as an optimal control problem, and then numerically solve it. The numerical solution is derived by converting the optimal control problem into a nonlinear programming (NLP) problem, and then solving that NLP. NLP solvers such as sequence quadratic programming (SQP) [7,8], interior-point method (IPM) [9], convex feasible set algorithm [10], and g²o [11] have been applied to parking oriented trajectory planning problems.

Compared with the sample-and-search-based planners, formulating an optimal control problem is advantageous because (i) the continuous state space needs not discretized into primitives; and (ii) trajectories are directly planned without path velocity decomposition. But the side effect of an optimization-based planner is the heavy computational burden. Typically an optimal control problem contains large numbers of non-convex collision-avoidance constraints, which hinder the online applications. A common remedy for this limitation is utilizing a sample-and-search-based planner to quickly find a coarse path/trajectory, then implementing numerical optimization with that initial guess [11–13]. Although a quickly searched initial guess facilitates the numerical solution process, the large-scale non-convex constraints still exist in the formulated optimal control problem, which make the optimization slow *whenever* the initial guess is not close to the optimum. Therefore, in addition to maintaining the initialization quality, substantial efforts are needed to simplify the optimal control problem formulation.

The primary difficulties in the concerned optimal control problem lie in the collision-avoidance constraints [14]. Since the vehicle may not have chances to collide with every obstacle in the environment, some of the collision-avoidance constraints can be safely removed, especially when a coarse path/trajectory is given (Fig. 1(a)). Following this idea, it is natural to consider paving a tunnel that separates the vehicle from all of the surrounding obstacles. With such a tunnel at hand, one can use the within-tunnel constraints to completely replace the collision-avoidance constraints. Through this, the dimensionality and non-convexity of the optimal control problem becomes independent from the complexity of the

* Research supported by the National Key R&D Program of China under Grant 2018YFB1600804, and the Natural Sciences and Engineering Research Council of Canada.

Bai Li is with the JDX R&D Center of Automated Driving, JD Inc., Beijing, China (e-mail: libai@zju.edu.cn, libai1@jd.com).

Tankut Acarman is with the Computer Engineering Department, Galatasaray University, Istanbul, Turkey (e-mail: tacarman@gsu.edu.tr).

Qi Kong is with the JDX R&D Center of Automated Driving, JD Inc., Beijing, China (e-mail: Qi.Kong@jd.com).

Youmin Zhang is with the Department of Mechanical, Industrial and Aerospace Engineering, Concordia University, Montreal, Canada (e-mail: ymzhang@encs.concordia.ca).

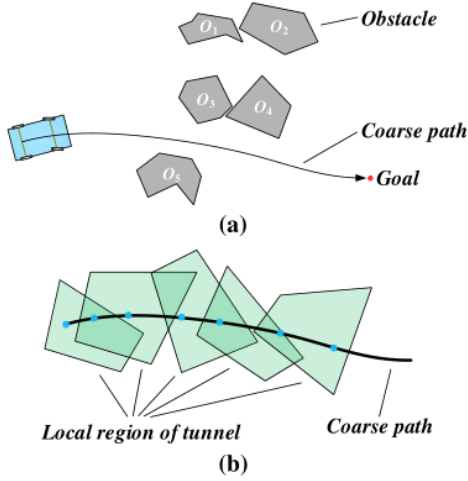


図1. (a)車両は1つのホモトピック粗経路を選択しているため、O1またはO2と衝突する可能性は低い。(b)車両が質量点であると仮定すると、従来の方法では、車両はすべての瞬間に局所領域の1つに留まるが、地上車両は質量点として扱うことができないため、地上車両が複数の局所領域に留まる可能性を記述するには、従来の方法は適用できない。

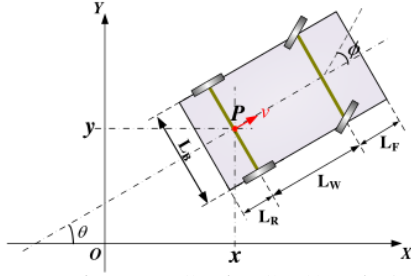


図2. 車両の運動学と幾何学に関する概略図。

環境である。トンネルベースの戦略は、航空機の軌道計画に適用された[15-17]。しかし、地上車両の形状を単純に質量点として扱うことはできないため、車体がトンネルの複数の局所領域をカバーしている可能性がある(図1(b))。この新しい課題により、これらの以前の方法論はここでは適用できない。

我々の知る限り、自律駐車問題に対するトンネルベースの戦略を検討した唯一の文献は[18]であり、そこでは、トンネル内条件が単純に車両の2次元位置と方位角に課される境界として定式化されるように、粗い経路/軌道の局所近傍が作成される。トンネル内条件を最も単純な線形制約として記述することができるが、この方法は、車両の形状に応じて近傍テンプレートを作成するために、非常に広範なオフラインの努力を必要とする。さらに、この方法は、近傍のサイズを決定する際に、あらかじめ定義された重み付けパラメータによって、角度スケールを盲目的に距離スケールに変換するため、解の実現可能性を失うリスクに悩まされる。

C. Contributions

本論文は、一般的な自律駐車問題に対して、高速で正確かつ最適に近い軌道を計画することを目的とする。この目的のために、トンネル構築戦略によって衝突回避制約を単純化する最適制御問題を定式化する。

これまでの研究と比較して、我々のトンネル建設戦略は、(i)オンライン使用前にオフラインの準備努力を必要とせず、(ii)車両の異なる部分がトンネルの異なるローカル領域に留まる可能性があるという問題に対処する。我々の新しいトンネルベースの戦略では、定式化された最適制御問題の規模は環境の複雑さから完全に独立しており、これは実際にはオンライン軌道計画において望ましい特性である。

D. Organization

本論文の残りの部分では、セクションIIで軌道計画指向の最適制御問題と数値解法原理を簡単に定義する。セクションIIIでは、トンネルベースの戦略によるトンネル内制約として、衝突回避制約を記述する方法を紹介する。シミュレーション結果と考察はセクションIVで行う。最後に、セクションVで結論を述べる。

II. 最適制御問題の定義と解決策

本節では、自律駐車軌道計画スキームを記述するための全体最適制御問題の定式化を提供し、この問題を数値的に解く方法を紹介する。

A. 最適制御問題の定式化

駐車場の軌道計画指向の最適制御問題は、コスト関数、車両運動学、境界条件、トンネル内制約から構成される。

車両は通常、駐車中に低速で走行するため、自転車モデルは車両の運動学を記述するのに十分である[9]：

$$\frac{d}{dt} \begin{bmatrix} x(t) \\ y(t) \\ v(t) \\ \phi(t) \\ \theta(t) \end{bmatrix} = \begin{bmatrix} v(t) \cdot \cos \theta(t) \\ v(t) \cdot \sin \theta(t) \\ a(t) \\ \omega(t) \\ v(t) \cdot \tan \phi(t) / L_w \end{bmatrix}, t \in [0, t_f]. \quad (1)$$

ここで、 t_f は駐車プロセス時間(固定されていない)を表し、 (x, y) は後輪軸の中心(図2の点P参照)を表し、 v はPの速度を表し、 a は対応する加速度プロファイルを表し、 ϕ は操舵角を表し、 ω は対応する角速度を表し、 θ は方位角を表し、 L_w はホイールベースを表す。 L_w に加えて、 L_f (前面オーバーハング長)、 L_r (背面オーバーハング長)、 L_b (幅)などの幾何学的パラメータも図2に描かれている。前述のプロファイルのいくつかに境界が課され、動作の機械的/物理的原理が記述されている：

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

境界条件は、初期モーメントと終端モーメントにおける車両の構成を指定する：

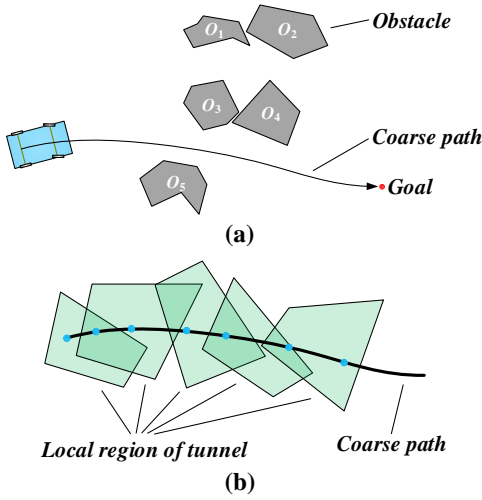


Fig. 1. Schematics on tunnel-based strategy: (a) since the vehicle has chosen one homotopic coarse path, it has slim chances to collide with O_1 or O_2 ; (b) assuming that the vehicle is a mass point, previous methods require that the vehicle stays in one of the local regions at every moment, but a ground vehicle cannot be treated as a mass point, thus previous methods are not applicable to describe the possibilities that a ground vehicle stays in more than one local regions.

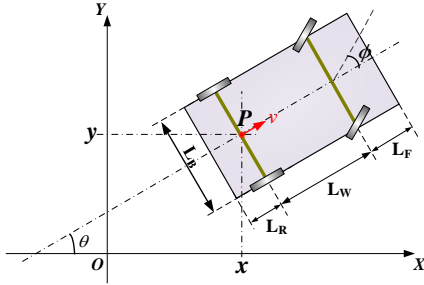


Fig. 2. Schematics on vehicle kinematics and geometries.

environment. The tunnel-based strategy was applied in the aerial vehicle trajectory planning [15–17]. However, since the shape of a ground vehicle cannot be simply treated as a mass point, it is possible that the vehicle body covers multiple local regions of the tunnel (Fig. 1(b)). This new challenge makes those previous methodologies not applicable here.

The only reference that considered the tunnel-based strategy for autonomous parking problems, to the best of our knowledge, is [18], wherein local neighborhoods of a coarse path/trajectory is created so that the within-tunnel conditions are simply formulated as bounds imposed on the vehicle's 2D location and the orientation angle. Although being able to describe the within-tunnel conditions as the simplest type of linear constraints, that method requires extremely extensive offline efforts to create the neighborhood templates according to the vehicle's shape. In addition, that method blindly converts the angular scale to a distance scale through a pre-defined weighting parameter when deciding the size of the neighborhood, thus it suffers from the risk to lose the solution feasibility.

C. Contributions

This paper aims to plan fast, accurate, and near-optimal trajectories for generic autonomous parking problems. To this end, an optimal control problem is formulated, wherein the collision-avoidance constraints are simplified through a tunnel

construction strategy. Compared with the previous works, our tunnel-construction strategy (i) does not require any offline preparation efforts before the online usage; and (ii) addresses the issue that different parts of the vehicle may stay in different local regions of the tunnel. With our novel tunnel-based strategy, the scale of the formulated optimal control problem is completely independent from the complexity of the environment, which is actually a desirable property in online trajectory planning.

D. Organization

In the rest of this paper, Section II briefly defines the trajectory planning oriented optimal control problem and the numerical solution principle. Section III introduces the way to describe the collision-avoidance restrictions as within-tunnel constraints through the tunnel-based strategy. Simulation results and discussions are provided in Section IV. Finally, the conclusions are drawn in Section V.

II. OPTIMAL CONTROL PROBLEM DEFINITION AND SOLUTION

This section provides the overall optimal control problem formulation for describing the autonomous parking trajectory planning scheme, and introduces how to solve the problem numerically.

A. Optimal Control Problem Formulation

The parking trajectory planning oriented optimal control problem consists of a cost function, vehicle kinematics, boundary conditions and the within-tunnel constraints.

Since a vehicle usually runs at low speeds during the parking process, the bicycle model is sufficient to describe the vehicle kinematics [9]:

$$\frac{d}{dt} \begin{bmatrix} x(t) \\ y(t) \\ v(t) \\ \phi(t) \\ \theta(t) \end{bmatrix} = \begin{bmatrix} v(t) \cdot \cos \theta(t) \\ v(t) \cdot \sin \theta(t) \\ a(t) \\ \omega(t) \\ v(t) \cdot \tan \phi(t) / L_W \end{bmatrix}, \quad t \in [0, t_f]. \quad (1)$$

Herein, t_f denotes the parking process duration (not fixed), (x, y) represents the mid-point of rear wheel axis (see point P in Fig. 2), v represents the velocity of P , a represents the corresponding acceleration profile, ϕ refers to the steering angle, ω represents the corresponding angular velocity, θ refers to the orientation angle, and L_W denotes the wheelbase. In addition to L_W , other geometric parameters such as L_F (front overhang length), L_R (rear overhang length), and L_B (width) are also depicted in Fig. 2. Boundaries are imposed on some of the aforementioned profiles to describe the mechanical/physical principles of the movement:

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

Boundary conditions specify the vehicle's configurations at the initial and terminal moments:

$$\begin{bmatrix} x(0) \\ y(0) \\ \theta(0) \\ v(0) \\ \phi(0) \\ a(0) \\ \omega(0) \end{bmatrix} = \begin{bmatrix} x_{init} \\ y_{init} \\ \theta_{init} \\ v_{init} \\ \phi_{init} \\ a_{init} \\ \omega_{init} \end{bmatrix}, \quad \begin{bmatrix} x(t_f) \\ y(t_f) \\ \theta(t_f) \\ v(t_f) \\ \phi(t_f) \\ a(t_f) \\ \omega(t_f) \end{bmatrix} = \begin{bmatrix} x_{final} \\ y_{final} \\ \theta_{final} \\ v_{final} \\ \phi_{final} \\ a_{final} \\ \omega_{final} \end{bmatrix}, \quad (3)$$

wherein x_{init} , y_{init} , θ_{init} , v_{init} , ϕ_{init} , a_{init} , ω_{init} , x_{final} , y_{final} , θ_{final} , v_{final} , ϕ_{final} , a_{final} , and ω_{final} are parameters which determine the starting and terminal configurations.

トンネル内制約は、障害物との衝突リスクを回避するために利用される。詳細は次節で紹介する。

最適制御問題のコスト関数については、効率と快適性の両方が考慮される。具体的には、効率は、最小時間条件下で駐車場移動を達成する期待値によって反映され、快適性は、 v と a を変更するために最小エネルギーを費やす期待値によって反映される：

$$J_{cost} = t_f + w_1 \cdot \int_{t=0}^{t_f} a^2(t) \cdot dt + w_2 \cdot \int_{t=0}^{t_f} \omega^2(t) \cdot dt, \quad (4)$$

ここで、 w_1 , $w_2 \geq 0$ は重み付けパラメータである。要約すると、自律駐車軌道計画タスクは、以下の最適制御問題として記述される：

$$\begin{aligned} & \text{Minimize (4),} \\ & \text{s.t. Kinematic constraints (1), (2);} \\ & \text{Within-tunnel constraints;} \\ & \text{Boundary conditions (3).} \end{aligned} \quad (5)$$

B. 最適制御問題に対する数値解法

(5)の未知数は $x(t)$, $y(t)$, $\theta(t)$, $v(t)$, $a(t)$, $\omega(t)$, t_f である。 $x(t)$, $y(t)$, $\theta(t)$, $v(t)$, $a(t)$, $\omega(t)$ が決まれば、残りのプロファイルは一意的に固定される。 $\theta(t)$, $a(t)$, $\omega(t)$ だけをシューティング法のように最適化するのではなく、状態プロファイルと制御プロファイルをすべて決定変数とみなすコロケーション法を採用する。コロケーション法は、シューティング法とは対照的に、一般的に高レベルの解の安定性を達成します[19]。

一次陽解法ルンゲクッタ法を適用して、7つのプロファイル $x(t)$, $y(t)$, $\theta(t)$, $v(t)$, $a(t)$, $\omega(t)$, t_f およびコスト関数/制約を離散化する。この離散化により、NLPが構築される。SQPがNLPソルバーとして選ばれたのは、バリア機能法(IPMなど)よりもウォームスタートフレンドリーだからである。最後に、SQPの出力は、駐車計画軌道となる。

III. 内部制約の定式化

本節では、最適制御問題(5)におけるトンネル内制約を定式化する。

A. Step 0. Dilating Obstacles and Shrinking Vehicle Body

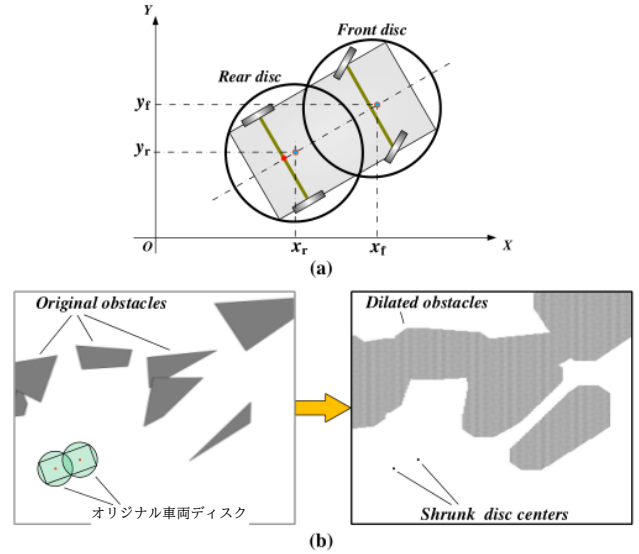


図3. 図3. 車両形状の縮小と環境マップの拡張に関する模式図: (a)2枚の円盤で車両形状を提示する、(b)車両形状と環境障害物を同時に新しい形に変換する。

一般的な手法として、ディスクは長方形の車体を覆うために使用することができる[20]。図3(a)に示すように、本研究では車体をカバーするために2枚のディスクを採用している。 $P_f = (x_f, y_f)$ と $P_r = (x_r, y_r)$ とすると、2つの円盤の中心は車両の縦軸に沿った四分位点、すなわち、

$$\begin{aligned} x_f(t) &= x(t) + \frac{1}{4}(3L_W + 3L_F - L_R) \cdot \cos \theta(t), \\ y_f(t) &= y(t) + \frac{1}{4}(3L_W + 3L_F - L_R) \cdot \sin \theta(t), \\ x_r(t) &= x(t) + \frac{1}{4}(L_W + L_F - 3L_R) \cdot \cos \theta(t), \\ y_r(t) &= y(t) + \frac{1}{4}(L_W + L_F - 3L_R) \cdot \sin \theta(t). \end{aligned} \quad (6a)$$

各ディスクの半径(R_C とする)は、以下の方法で決定される。

$$R_C = \frac{1}{2} \sqrt{\left(\frac{L_R + L_W + L_F}{2}\right)^2 + (L_B)^2}. \quad (6b)$$

一般的に、衝突回避制約は、各ディスクが障害物を排除することを要求するもので、これは各ディスクの中心が障害物から少なくとも R_C の距離を保つべきであるという制約と同じである。つまり、2つの円盤を同時に中心として縮小し(すなわち P_f と P_r)、障害物を R_C で拡張する等価変換を行うことができる(図3(b))。この等価変換だけでは、プランニングタスクに本質的な変更を加えることはできないが、次のいくつかのサブセクションで紹介する、我々のトンネル内制約の定式化に貢献するものである。

この小節のまとめとして、元の軌道計画スキームを新しい形に変換する: 車体は2つの質量点に縮小され、環境障害物は R_C によって拡張される。本稿の残りの部分では、拡張障害物を含む環境マップを拡張マップと呼ぶ。

$$\begin{bmatrix} x(0) \\ y(0) \\ \theta(0) \\ v(0) \\ \phi(0) \\ a(0) \\ \omega(0) \end{bmatrix} = \begin{bmatrix} x_{\text{init}} \\ y_{\text{init}} \\ \theta_{\text{init}} \\ v_{\text{init}} \\ \phi_{\text{init}} \\ a_{\text{init}} \\ \omega_{\text{init}} \end{bmatrix}, \quad \begin{bmatrix} x(t_f) \\ y(t_f) \\ \theta(t_f) \\ v(t_f) \\ \phi(t_f) \\ a(t_f) \\ \omega(t_f) \end{bmatrix} = \begin{bmatrix} x_{\text{final}} \\ y_{\text{final}} \\ \theta_{\text{final}} \\ v_{\text{final}} \\ \phi_{\text{final}} \\ a_{\text{final}} \\ \omega_{\text{final}} \end{bmatrix}, \quad (3)$$

wherein x_{init} , y_{init} , θ_{init} , v_{init} , ϕ_{init} , a_{init} , ω_{init} , x_{final} , y_{final} , θ_{final} , v_{final} , ϕ_{final} , a_{final} , and ω_{final} are parameters which determine the starting and terminal configurations.

The within-tunnel constraints are utilized to avoid the collision risks with the obstacles. The details are introduced in the next section.

Regarding the cost function of the optimal control problem, both efficiency and comfort are considered. Concretely, efficiency is reflected by the expectation to accomplish the parking movement subject to minimum time, and comfort is reflected by the expectation to spend minimum energy to change v and ϕ :

$$J_{\text{cost}} = t_f + w_1 \cdot \int_{t=0}^{t_f} a^2(t) \cdot dt + w_2 \cdot \int_{t=0}^{t_f} \omega^2(t) \cdot dt, \quad (4)$$

wherein $w_1, w_2 \geq 0$ are weighting parameters. As a summary, the autonomous parking trajectory planning task is described as the following optimal control problem:

$$\begin{aligned} & \text{Minimize (4),} \\ & \text{s.t. Kinematic constraints (1), (2);} \\ & \quad \text{Within-tunnel constraints;} \\ & \quad \text{Boundary conditions (3).} \end{aligned} \quad (5)$$

B. Numerical Solution to Optimal Control Problem

The unknowns in (5) include $x(t)$, $y(t)$, $\theta(t)$, $v(t)$, $a(t)$, $\omega(t)$, $\phi(t)$, and t_f . If $\omega(t)$, $a(t)$, and t_f are determined, the rest profiles are uniquely fixed. Instead of optimizing only $\omega(t)$, $a(t)$, and t_f like a shooting method, this work adopts the collocation method which regards all of the state and control profiles as decision variables. A collocation method typically achieves high-level solution stability in contrast to a shooting method [19].

The first-order explicit Runge-Kutta method is applied to discretize the 7 profiles $x(t)$, $y(t)$, $\theta(t)$, $v(t)$, $a(t)$, $\omega(t)$, $\phi(t)$, as well as the cost function/constraints. Through this discretization, an NLP is built. SQP is chosen as the NLP solver because it is more warm-starting friendly than a barrier-function method (such as IPM). Finally, the output of SQP is the planned trajectory for parking.

III. WITHIN-TUNNEL CONSTRAINT FORMULATION

This section formulates the within-tunnel constraints in the optimal control problem (5).

A. Step 0. Dilating Obstacles and Shrinking Vehicle Body

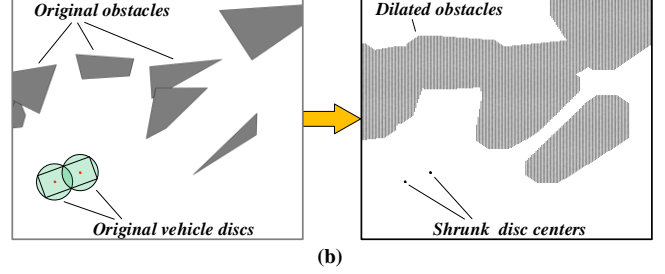
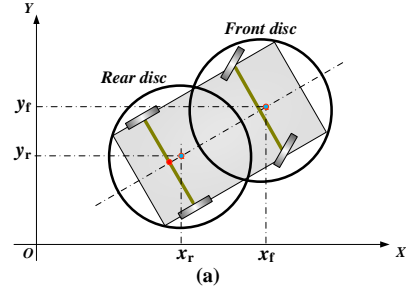


Fig. 3. Schematics on vehicle shape shrink and environmental map dilatation: (a) presenting vehicle shape with two discs; (b) simultaneously converting the vehicle shape and environmental obstacles into new forms.

As a common practice, discs can be used to cover the rectangular vehicle body [20]. As depicted in Fig. 3(a), this work adopts 2 discs to cover the vehicle body. Denoted as $P_f = (x_f, y_f)$ and $P_r = (x_r, y_r)$, the centers of the two discs are quartile points along the vehicle's longitudinal axis, i.e.,

$$\begin{aligned} x_f(t) &= x(t) + \frac{1}{4}(3L_w + 3L_f - L_r) \cdot \cos \theta(t), \\ y_f(t) &= y(t) + \frac{1}{4}(3L_w + 3L_f - L_r) \cdot \sin \theta(t), \\ x_r(t) &= x(t) + \frac{1}{4}(L_w + L_f - 3L_r) \cdot \cos \theta(t), \\ y_r(t) &= y(t) + \frac{1}{4}(L_w + L_f - 3L_r) \cdot \sin \theta(t). \end{aligned} \quad (6a)$$

The radius of each disc, denoted as R_c , is determined via

$$R_c = \frac{1}{2} \sqrt{\left(\frac{L_r + L_w + L_f}{2}\right)^2 + (L_b)^2}. \quad (6b)$$

Nominally the collision-avoidance constraints require that each disc keeps clear of the obstacles, which is identical to the restriction that each disc center should keep at least a distance of R_c from the obstacles. This means we can make an equivalent conversion to simultaneously shrink the two discs as their centers (i.e. P_f and P_r) and dilate the obstacles by R_c (Fig. 3(b)). Although this equivalent conversion alone does not make any inherent change to the planning task, it contributes to the formulation of our within-tunnel constraints, which will be introduced in the next few subsections.

As a summary of this subsection, the original trajectory planning scheme is converted into a new form: the vehicle body is shrunk to two mass points, and the environmental obstacles are dilated by R_c . In the rest of the paper, we refer to the environmental map with dilated obstacle as *dilated map*.

B. ステップ1. 参照軌道の生成

本節では、 P_f と P_r の参照軌道を生成し、今後の利用を目指す。このため、まずハイブリッドA*アルゴリズムを採用して点 $P^*(x, y)$ の経路を求め、その経路に沿って時間経過を付加して軌跡 $Traj_p$ を形成し、 $Traj_p$ に従って P_f と P_r の軌跡を一意に決定する。そのほか、 $Traj_p$ もNLP解法の初期推測として用いられる。

ハイブリッドA*アルゴリズムを選択した理由は、運動学的モデルを尊重し、後方操作をサポートするためである。この予備的な研究では、駐車シナリオに予測可能で扱いやすい移動障害物がないと仮定すると、ハイブリッドA*によって導出された経路に沿って時間経過を付加する手順は、1次元最小時間最適制御問題となり、これはPontryaginの最大原理によって解析的に解くことができる。 $Traj_p$ が決まれば、 P_f と P_r の軌道は(6a)に従って一意に決まる。 $Traj_p$ の時間領域を $t \in [0, t_f]$ とし、 P_f の軌跡と

P_r as $Traj_{P_f}$ and $Traj_{P_r}$, respectively.

名目上、 P_f と P_r は、拡張マップ内の拡張障害物との衝突を避けるべきである。ここで、基準軌道 $Traj_{P_f}$ と $Traj_{P_r}$ を手に、衝突回避制約を定式化する必要がなくなった。代わりに、 P_f と P_r がそれぞれ $Traj_{P_f}$ と $Traj_{P_r}$ と同相の2つのトンネルに留まることだけを要求する。トンネルは非凸であるため、各トンネルをカバーするために局所的な凸ボックスを使用する。これらのローカルボックスは、本論文の残りの部分では代表ボックスと呼ばれる。

C. ステップ2. 代表ボックスの指定

本節では、 $Traj_{P_f}$ と $Traj_{P_r}$ をそれぞれカバーする2つの代表的なボックスを生成する。まず、 $Traj$ のボックス生成スキームに注目しよう。

To begin with, we sample $(N_R + 1)$ waypoints along $Traj_{P_f}$ evenly in the time horizon. Concretely, the waypoints

$\tilde{t}_i \in [0, t_f]$ for $i = 0, 1, \dots, N_R$ are sampled along $Traj_{P_f}$. These waypoints are called representative nodes (see Fig. 4 for reference). $(N_R + 1)$ representative nodes are available, and the next step is, for each representative node, to specify $(N_R + 1)$ representative boxes. The principle of specifying the representative boxes is as follows.

まず、有限差分法を用いて、現在関係する代表ノード $* x_f(t_k)$ における方位角 $* y_f(t_k)$ を $Traj_{P_f}$ に沿って指定する。

$t_k = \frac{\tilde{t}_i}{N_R} \cdot k$. Secondly, we define four direction, namely $\theta_i(t_k)$,

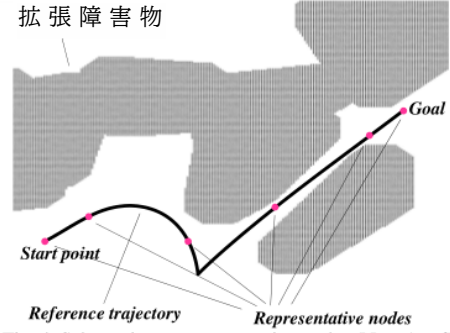


Fig. 4. Schematics on representative nodes ($N_R + 1 = 6$).

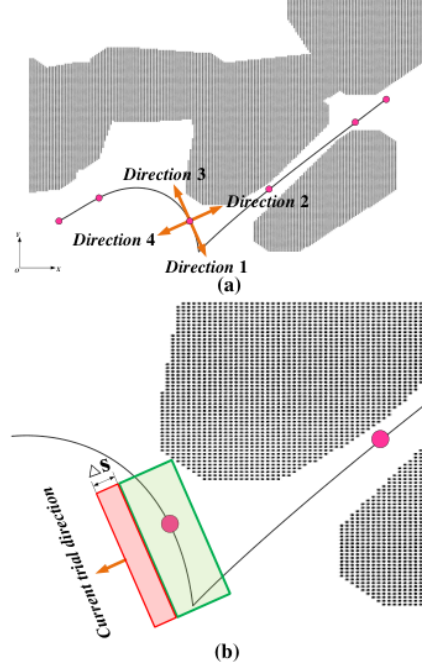


図5. 各代表ボックスを指定する原理の模式図: (a)膨張方向、(b)一方向の膨張試行。

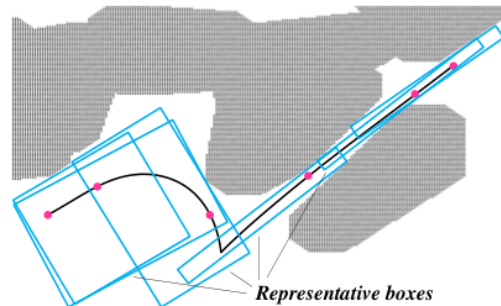


図6. 代表的な箱で形成されたトンネルの模式図。理想的には、これらの代表的なボックスは、基準軌道を完全にカバーする必要がある。

$\theta_i(t_k) + \frac{\pi}{2}$, $\theta_i(t_k) + \pi$, and $\theta_i(t_k) + \frac{3\pi}{2}$ (see Fig. 5(a)). We

representative nodes as zero-width-zero-length rectangles (cores) and expand them in four directions with a fixed step size s gradually. During the expansion trial, we check whether the trial path collides with obstacles in the expanded map. If no collision occurs, the trial is accepted; otherwise, it is rejected, and the next trial in that direction is not considered.

B. Step 1. Generating Reference Trajectories

This subsection aims to generate the reference trajectories of P_f and P_r for future usages. To this end, we first adopt the hybrid A* algorithm to find a path for point $P = (x, y)$, then attach a time course along that path to form a trajectory $Traj_p$, and then uniquely determine the trajectories of P_f and P_r according to $Traj_p$. Besides that, $Traj_p$ is also used as the initial guess in the NLP solution process.

The reason of choosing the hybrid A* algorithm is it respects the kinematic model and supports backward maneuvers. In this preliminary work, we assume that there are no predictable or tractable moving obstacles in the parking scenario, then the procedure to attach a time course along the path derived by hybrid A* becomes a one-dimensional minimum-time optimal control problem, which can be analytically solved via Pontryagin's maximum principle. Once $Traj_p$ is determined, the trajectories of P_f and P_r are uniquely determined according to (6a). Let us denote the time domain of $Traj_p$ as $t \in [0, \tilde{t}_f]$, and the trajectories of P_f and P_r as $Traj_{P_f}$ and $Traj_{P_r}$, respectively.

Nominally, P_f and P_r should avoid collisions with any of the dilated obstacles in the dilated map. Now with the reference trajectories $Traj_{P_f}$ and $Traj_{P_r}$ at hand, we no longer need to formulate the collision-avoidance constraints. Instead, we only need to require that P_f and P_r stay in two tunnels that are homotopic with $Traj_{P_f}$ and $Traj_{P_r}$, respectively. Since the tunnels are non-convex, we use local convex boxes to cover each tunnel. Those local boxes are referred to as *representative boxes* in the rest of this paper.

C. Step 2. Specifying Representative Boxes

In this subsection, two series of representative boxes are generated to cover $Traj_{P_f}$ and $Traj_{P_r}$, respectively. Let us focus on the box generation scheme for $Traj_{P_f}$ at first.

To begin with, we sample $(N_R + 1)$ waypoints along $Traj_{P_f}$ evenly in the time horizon. Concretely, the waypoints $(x_f(t), y_f(t))$ at $t = \frac{\tilde{t}_f}{N_R} \cdot i$ ($i = 0, 1, \dots, N_R$) are extracted from $Traj_{P_f}$. These waypoints are referred to as *representative nodes* (see the example illustrated in Fig. 4). Once the $(N_R + 1)$ representative nodes are available, the next step is to specify $(N_R + 1)$ representative boxes in association with the representative nodes. The principle to specify the k th representative box is introduced as follows.

Firstly, we use the finite difference method to specify the orientation angle $\theta_f(t_k)$ at the currently concerned representative node $(x_f(t_k), y_f(t_k))$ along $Traj_{P_f}$, where $t_k = \frac{\tilde{t}_f}{N_R} \cdot k$. Secondly, we define four direction, namely $\theta_f(t_k)$,

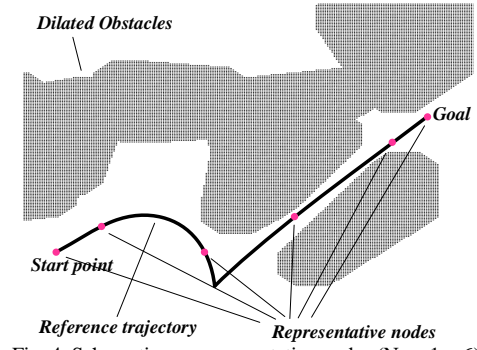


Fig. 4. Schematics on representative nodes ($N_R + 1 = 6$).

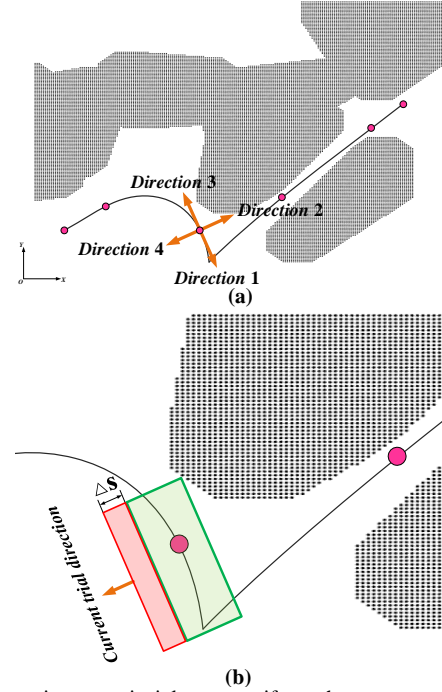


Fig. 5. Schematics on principle to specify each representative box: (a) expansion directions; (b) expansion trial in one direction.

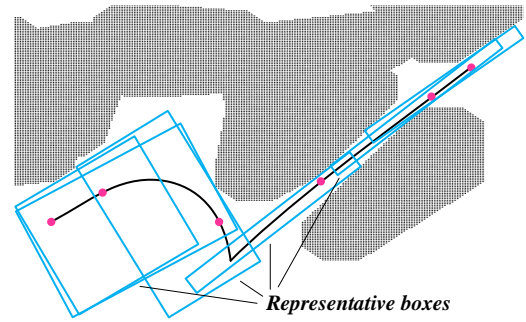


Fig. 6. Schematics on tunnel formed by representative boxes. Ideally these representative boxes should fully cover the reference trajectory.

$\theta_f(t_k) + \frac{\pi}{2}$, $\theta_f(t_k) + \pi$, and $\theta_f(t_k) + \frac{3\pi}{2}$ (see Fig. 5(a)). We regard the representative node as a zero-width-zero-length rectangle (i.e. a core), and incrementally expand the core in each of the four directions by a constant step Δs each time. Whenever an expansion trial in one direction is made, we check if that trial causes collisions with the obstacles in the dilated map. If no collision occurs, then that trial is approved; otherwise it is rejected and future expansion trials in that

図5(b)を例にとると、緑色のボックスが現在承認されている領域を示し、現在懸念されている方向への拡大試行が赤色のボックスをレンダリングするとする。赤枠は拡張マップの障害物と重ならないため、赤枠が承認され、承認された領域は緑枠と赤枠の両方で構成されるようになった。さらに、各方向の過剰な膨張を避けるために、最大膨張長パラメータ L_{limit} を定義する。k番目の代表ボックスの幾何学的サイズを指定する原理は、以下の擬似コードにまとめられる。

アルゴリズム 1. 代表的なボックス指定アプローチ。

入力: 入力: 拡張画像、* $x_f(t_k)$, $y_f(t_k)$, * (t_k) *, パラメータ L_{limit} ;

出力: 代表ボックスkの頂点位置。

1. 従って4つの展開方向を定義する * $f(t_k)$) ;
2. インデックス集合 * $\{1, 2, 3, 4\}$ を初期化する;
3. ベクトルを長さ * $[0, 0, 0, 0]$ で初期化する;
4. 承認領域 * を質量点 * $x_f(t_k)$, $y_f(t_k)$ * として初期化する;
5. **While** $\Omega \neq \emptyset$, **do**
6. **For each** $i \in \Omega$, **do**
7. Expand Υ in direction i by Δs to form a trial box region Υ_{trial} ;
8. **If** Υ_{trial} does not overlap with obstacles of dilated map, **then**
9. Update $length_i \leftarrow length_i + \Delta s$;
10. Merge Υ_{trial} into Υ ;
11. **If** $length_i > L_{limit}$, **then**
12. Remove i from Ω ;
13. **End if**
14. **Else**
15. Remove i from Ω ;
16. **End if**
17. **End for**
18. **End while**
19. 4つの頂点の位置を*に従って指定する;
20. Output vertex locations;
21. Return.

すべての代表箱を同じように指定することで、 $(N_R * 1)$ 個の代表箱からなる点 P_f のトンネルを開くことができます(図6)。 P_f のトンネルも同様に舗装できる。混乱を避けるため、 P_f のトンネルをフロントトンネル、 P_r のトンネルをリアトンネルと表記する。

D. ステップ3. トンネル内制約の定式化

過去数節の準備に基づいて、この節ではトンネル内制約を正式に構築する。具体的には、以下のことが必要である。

$P_f(t)$ は前方トンネルからk番目の代表箱に留まる。

$$\text{when } t \in \left[\frac{t_f}{N_R} \cdot k, \frac{t_f}{N_R} \cdot (k+1) \right], k = 0, \dots, N_R - 1; \quad (7a)$$

$P_r(t_f)$ は前方トンネルから N_R 番目の代表箱に留まる。

同様に、 P_r についても以下の制約を課す:

$P_r(t)$ は後方トンネルからk番目の代表箱に留まる。

$$\text{when } t \in \left[\frac{t_r}{N_R} \cdot k, \frac{t_r}{N_R} \cdot (k+1) \right], k = 0, \dots, N_R - 1; \quad (7b)$$

$P_r(t_f)$ は後方トンネルから N_R 番目の代表箱に留まる。

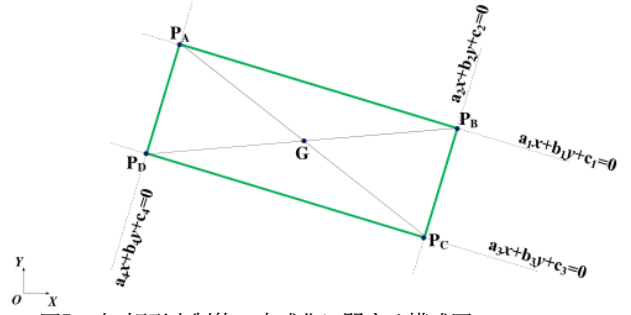


図7. 点-矩形内制約の定式化に関する模式図。

次に、このような制約を代数的不等式として記述する方法を簡単に紹介する。

点Sが不規則に配置された矩形に留まるように制限することは、点Sが同時に矩形の各辺の片側に留まるという条件と同じである。図7を例に、矩形の4つの頂点を P_A 、 P_B 、 P_C 、 P_D とし、点Sを (x_s, y_s) とする。ここで、矩形内制約とは、直線 $P_A P_B$ 、 $P_B P_C$ 、 $P_C P_D$ 、 $P_D P_A$ で囲まれた領域内に点Sが位置する制約と記述する。それぞれの直線は等式で表すことができる。例えば、線分 $P_A P_B$ は、 $a_1 * x * b_1 * y * c_1 * 0$ と記述される。

$$\begin{aligned} a_1 &= y_2 - y_1, \\ b_1 &= x_1 - x_2, \\ c_1 &= x_2 \cdot y_1 - x_1 \cdot y_2, \\ P_A &= (x_1, y_1), P_B = (x_2, y_2). \end{aligned} \quad (8)$$

Sが直線 $P_A P_B$ の片側にあることを要求することは、 $a_1 * x_s * b_1 * y_s * c_1 * 0$ または $a_1 * x_s * b_1 * y_s * c_1 * 0$ と記述できる。2つの不等式からどちらかを選ぶ方法について、点Sと矩形の幾何学的中心(図7ではG * (x_{center}, y_{center}) と表記)が直線 $P_A P_B$ の同じ側に留まることに気づく。したがって、 $a_1 * x_{center} * b_1 * y_{center} * c_1 * 0$ ならば、 $a_1 * x_s * b_1 * y_s * c_1 * 0$ を選ぶ。残りの3本の直線に関連する制約も同様に指定する。結論として、4つの線形不等式が点-矩形内制約を構成する。

我々が関心を持つ軌道計画スキームに関して、 $8 * (N_R * 1)$ 種類の単純な不等式が、完全にトンネル内制約を構成する。明らかに、制約の規模は環境中の障害物の数とは無関係である。非微分で非凸である公称衝突回避制約(例えば[21])とは対照的に、我々のトンネル内制約はほぼ線形であり、それによってNLPソルバーで簡単に処理することができる。

本節の最後に、トンネル内を移動するのに必要なのは2点(すなわち P_f と P_r) だけであるが、トンネルは元のマップではなく、拡張されたマップに従って生成されることを強調したい。したがって、 P_f と P_r がそれぞれのトンネル内に留まれば、車体全体が障害物を排除することは安全に保証される。

direction are not considered any longer. Let us take Fig. 5(b) as an example, suppose the green box denotes the currently approved region, and the expansion trial in the currently concerned direction renders the red box. Since the red box does not overlap with the obstacles in the dilated map, the red box is approved, which means the approved region now consists of both the green and red boxes. Besides that, we define a maximum expansion length parameter L_{limit} to avoid excessive expansions in each direction. The principle to specify the geometric size of the k th representative box is summarized in the following pseudo code.

Algorithm 1. Representative box specification approach.

Input: Dilated map, $[x_f(t_k), y_f(t_k), \theta_f(t_k)]$, and parameter L_{limit} ;

Output: Vertex locations of representative box k .

1. Define four expansion directions according to $\theta_f(t_k)$;
2. Initialize an index set $\Omega = \{1, 2, 3, 4\}$;
3. Initialize a vector as **length** = $[0, 0, 0, 0]$;
4. Initialize approved region Y as the mass point $(x_f(t_k), y_f(t_k))$;
5. **While** $\Omega \neq \emptyset$, **do**
6. **For each** $i \in \Omega$, **do**
7. Expand Y in direction i by Δs to form a trial box region Y_{trial} ;
8. **If** Y_{trial} does not overlap with obstacles of dilated map, **then**
9. Update **length** $_i \leftarrow \text{length}_i + \Delta s$;
10. Merge Y_{trial} into Y ;
11. **If** **length** $_i > L_{\text{limit}}$, **then**
12. Remove i from Ω ;
13. **End if**
14. **Else**
15. Remove i from Ω ;
16. **End if**
17. **End for**
18. **End while**
19. Specify four vertexes' locations according to Y ;
20. Output vertex locations;
21. Return.

Through specifying all of the representative boxes in the same way, we can pave a tunnel for point P_f , which consists of $(N_R + 1)$ representative boxes (Fig. 6). The tunnel for P_r can be paved in the same way. To avoid confusion, the tunnel for P_f is denoted as the *front tunnel*, while the one for P_r is denoted as the *rear tunnel*.

D. Step 3. Formulating Within-Tunnel Constraints

Based on the preparations in the past few subsections, this subsection formally builds the within-tunnel constraints. Concretely, we require that

$P_f(t)$ stays in k th representative box from front tunnel

$$\text{when } t \in \left[\frac{t_f}{N_R} \cdot k, \frac{t_f}{N_R} \cdot (k+1) \right], k = 0, \dots, N_R - 1; \quad (7a)$$

$P_f(t_f)$ stays in N_R th representative box from front tunnel.

Similarly, we impose the following constraints for P_r :

$P_r(t)$ stays in k th representative box from rear tunnel

$$\text{when } t \in \left[\frac{t_f}{N_R} \cdot k, \frac{t_f}{N_R} \cdot (k+1) \right], k = 0, \dots, N_R - 1; \quad (7b)$$

$P_r(t_f)$ stays in N_R th representative box from rear tunnel.

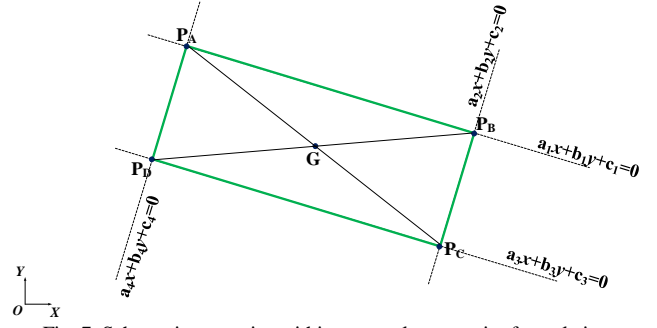


Fig. 7. Schematics on point-within-rectangle constraint formulation.

Next, we will briefly introduce how to describe such restrictions as algebraic inequalities.

Restricting a point S to stay in an irregularly placed rectangle is identical to the condition that the point S simultaneously stays on some one side of the rectangle's each edge. Taking Fig. 7 as an example, we denote the four vertexes of the rectangle as P_A , P_B , P_C , and P_D , and denote point S as (x_s, y_s) . Now the within-rectangle constraint is described as the restriction that point S should locate within the region surrounded by straight lines P_AP_B , P_BP_C , P_CP_D , and P_DP_A . Each straight line can be presented in the form of an equality. For example, line P_AP_B is described as $a_1 \cdot x + b_1 \cdot y + c_1 = 0$ with

$$\begin{aligned} a_1 &= y_2 - y_1, \\ b_1 &= x_1 - x_2, \\ c_1 &= x_2 \cdot y_1 - x_1 \cdot y_2, \\ P_A &= (x_1, y_1), P_B = (x_2, y_2). \end{aligned} \quad (8)$$

Requiring that S stays on one side of line P_AP_B can be described as $a_1 \cdot x_s + b_1 \cdot y_s + c_1 > 0$ or $a_1 \cdot x_s + b_1 \cdot y_s + c_1 < 0$. Regarding how to choose one between the two inequalities, we notice that point S and the rectangle's geometric center (denoted as $G = (x_{\text{center}}, y_{\text{center}})$ in Fig. 7) stay on the same side of line P_AP_B . Therefore, if $a_1 \cdot x_{\text{center}} + b_1 \cdot y_{\text{center}} + c_1 > 0$, then we choose $a_1 \cdot x_s + b_1 \cdot y_s + c_1 > 0$, and vice versa. The constraints in association with the rest three straight lines are specified in the same way. As a conclusion, four linear inequalities constitute the point-within-rectangle constraint.

Regarding the trajectory planning scheme we concern, $8 \cdot (N_R + 1)$ types of simple inequalities totally constitute the within-tunnel constraints. Apparently, the scale of the constraints is irrelevant to the number of obstacles in the environment. In contrast with the nominal collision-avoidance constraints (e.g. in [21]) which are non-differentiable and non-convex, our within-tunnel constraints are nearly linear, thereby being easy to be handled by the NLP solver.

Before the end of this section, we would like to emphasize that although we only require the two points (namely P_f and P_r) to travel in the tunnels, the tunnels are generated according to the dilated map rather than the original map, thus it is safely guaranteed that the whole vehicle body keeps clear of the obstacles provided that P_f and P_r stay in their own tunnels.

IV. シミュレーション結果と考察

シミュレーションはC++で行い、3.60×2GHzで動作するi7-7700 CPUと8GB RAMで実行した。SNOPTでは、SQPの商用ソフトウェアパッケージをデフォルトのオプションでAMPLに利用した。MATLAB 2019aを用いてシミュレーション結果を実証した。基本的なパラメトリック設定を表1に示す。主要なシミュレーション結果を含む動画は<https://youtu.be/brQo91Pw9cw>に掲載されている。

表1. モデルとアプローチに関するパラメトリックな設定.

Parameter	Description	Setting
L_F	車両のフロントハング長	0.96 m
L_W	Wheelbase of vehicle.	2.80 m
L_R	Rear hang length of vehicle.	0.929 m
L_B	Width of vehicle.	1.942 m
a_{\max}	Upper bound of $ a(t) $.	4.0 m/s ²
v_{\max}	Upper bound of $ v(t) $.	3.0 m/s
Φ_{\max}	Upper bound of $ \phi(t) $.	0.70 rad
Ω_{\max}	Upper bound of $ \omega(t) $.	0.5 rad/s
w_1, w_2	Weights in cost function (4).	0.1, 0.01
$N_R + 1$	各トンネルの代表的なボックスの数。	61
Δs	アルゴリズム1の単位ステップ長。	0.1 m
L_{limit}	アルゴリズム1の最大ステップ長。	8.0 m
N_{fe}	Number of finite elements in Runge-Kutta method.	60

A. 軌道プランナーの有効性について

シミュレーションの第一ラウンドは、計画された軌道の有効性に焦点を当てる。3つの駐車ケースの最適化された軌道を図に示す。それぞれ8-10である。ケース1とケース2は、自車両の近くに不規則に駐車している車があるシナリオを表し、ケース3は乱雑な環境を表している。図中のフットプリントによると8-10において、エゴ・ビークルは障害物との衝突を回避することに成功し、提案プランナの有効性を示している。特に、ケース1の最適化プロファイル $v(t)$ と $\phi(t)$ は、車両運動学的制約(1)と(2)の満足度を反映した図11のようになる。

B. トンネルベース戦略の有効性について

第2ラウンドのシミュレーションでは、提案するトンネルベースの戦略の有効性を調査する。ケース2を例にとってみよう。図12は、ハイブリッドA*アルゴリズムによって導出された粗い経路、 N_R の様々な設定の下でトンネルベース戦略によって最適化された軌道、および完全な衝突回避制約を持つ数値最適制御アプローチ[21]によって導出された局所最適をプロットしたものである。ハイブリッドA*アルゴリズムで得られた粗い経路と比較して、最適化された軌道はより滑らかである。[21]で導かれた局所最適とは対照的に、トンネルベースの戦略で得られた軌道は最適ではなく、 N_R が大きくなるにつれて解が局所最適に収束する傾向はない。図13は、舗装されたトンネルを端的に印象づけるために、 $N_R * 40$ と $N_R * 200$ のトンネルを示している。

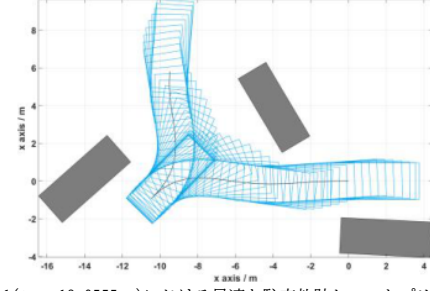


図8. ケース1($t_f * 13.9555$ s)における最適な駐車軌跡とフットプリント。

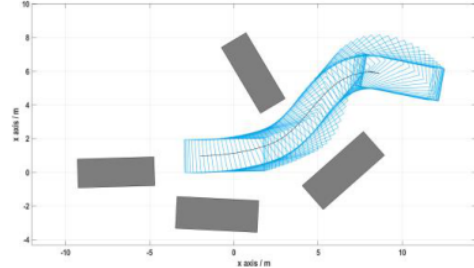


図9. ケース2($t_f * 10.3821$ s)における最適な駐車軌跡とフットプリント。

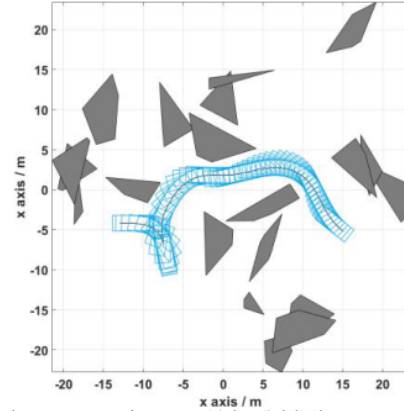


図10. ケース3($t_f * 21.6495$ s)における最適な駐車軌跡とフットプリント。

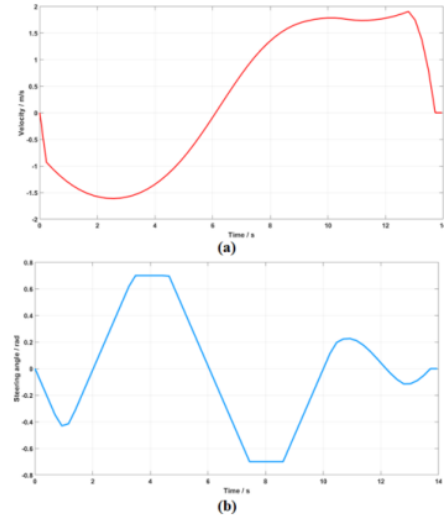


図11. ケース1の最適化プロファイル:(a) $v(t)$, (b) $\phi(t)$.

この図では、走行可能領域の一部が代表的なボックスで覆われていないことがわかる。この現象は、この研究の限界とみなすことができ、他のタイプの代表的なポリゴンを使用することで、状況は改善されるが、アルゴリズム1はより複雑になる。

IV. SIMULATION RESULTS AND DISCUSSIONS

Simulations were performed in C++ and executed on an i7-7700 CPU with 8 GB RAM that runs at 3.60×2 GHz. SNOPT, a commercial software package of SQP was utilized in AMPL with default options. MATLAB 2019a was used to demonstrate the simulation results. Basic parametric settings are listed in Table I. A video with the primary simulation results is provided at <https://youtu.be/brQo9lPw9cw>.

TABLE I. PARAMETRIC SETTINGS REGARDING MODEL AND APPROACH.

Parameter	Description	Setting
L_F	Front hang length of vehicle.	0.96 m
L_W	Wheelbase of vehicle.	2.80 m
L_R	Rear hang length of vehicle.	0.929 m
L_B	Width of vehicle.	1.942 m
a_{\max}	Upper bound of $ a(t) $.	4.0 m/s ²
v_{\max}	Upper bound of $ v(t) $.	3.0 m/s
Φ_{\max}	Upper bound of $ \phi(t) $.	0.70 rad
Ω_{\max}	Upper bound of $ \omega(t) $.	0.5 rad/s
w_1, w_2	Weights in cost function (4).	0.1, 0.01
$N_R + 1$	Number of representative boxes in each tunnel.	61
Δs	Unit step length in Algorithm 1.	0.1 m
L_{limit}	Maximum step length in Algorithm 1.	8.0 m
N_{fe}	Number of finite elements in Runge-Kutta method.	60

A. On the Efficacy of Trajectory Planner

The first round of simulations focuses on the efficacy of the planned trajectories. The optimized trajectories of three parking cases are depicted in Figs. 8–10, respectively. Cases 1 and 2 represent the scenarios with irregularly parked cars near our ego-vehicle, while Case 3 represents a cluttered environment. According to the footprints in Figs. 8–10, the ego-vehicle manages to avoid collisions with the obstacles, which show the efficacy of the proposed planner. Particularly, the optimized profiles $v(t)$ and $\phi(t)$ in Case 1 are shown in Fig. 11, which reflect satisfactions to the vehicle kinematic restrictions (1) and (2).

B. On the Efficacy of Tunnel-based Strategy

The second round of simulations investigates the efficacy of the proposed tunnel-based strategy. Let us take Case 2 as an example. Fig. 12 plots the coarse path derived by the hybrid A* algorithm, the optimized trajectories with the tunnel-based strategy under various settings of N_R , and the local optimum derived by the numerical optimal control approach [21] with complete collision-avoidance constraints. Compared with the coarse path obtained by the hybrid A* algorithm, the optimized trajectories are smoother. In contrast with the local optimum derived by [21], the trajectories obtained with the tunnel-based strategy are not optimal, and there is not a trend that the solution converges to the local optimum as N_R grows. In order to have a straightforward impression of the paved tunnels, Fig. 13 illustrates the tunnels with $N_R = 40$ and $N_R = 200$. In that figure, it is obvious that part of the drivable

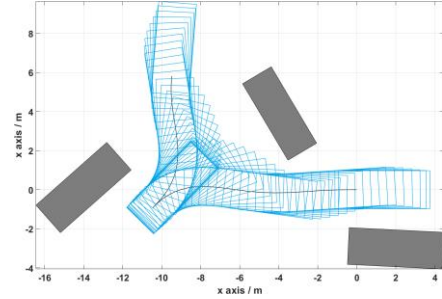


Fig. 8. Optimized parking trajectory and footprints in Case 1 ($t_f = 13.9555$ s).

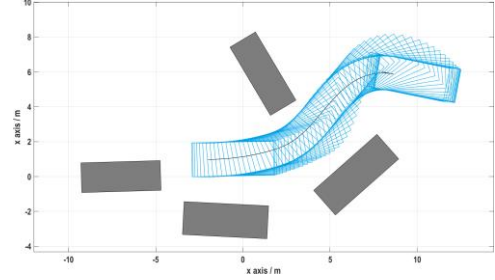


Fig. 9. Optimized parking trajectory and footprints in Case 2 ($t_f = 10.3821$ s).

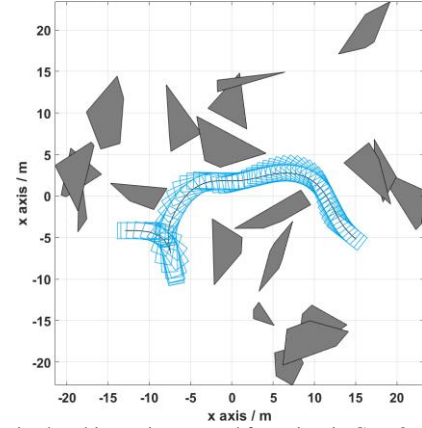


Fig. 10. Optimized parking trajectory and footprints in Case 3 ($t_f = 21.6495$ s).

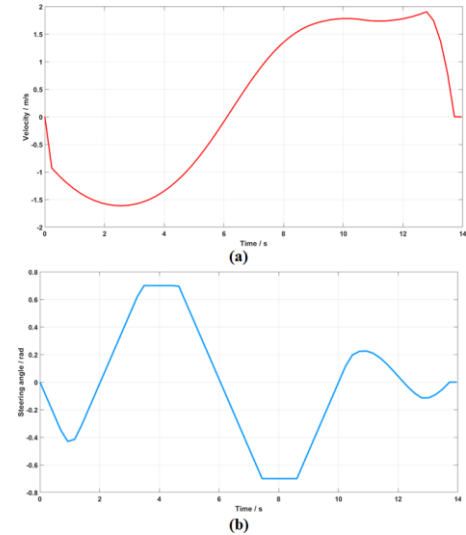


Fig. 11. Optimized profiles in Case 1: (a) $v(t)$, and (b) $\phi(t)$.

area are not covered by the representative boxes. This phenomenon may be regarded as a limitation of this work, and using other types of representative polygons would improve the situation but Algorithm 1 becomes more complicated then.

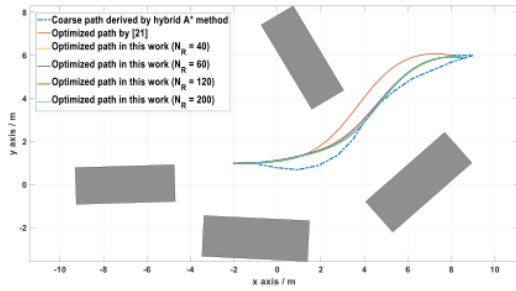


図12. N_R を様々に設定した場合の最適化された軌道(ケース2)。

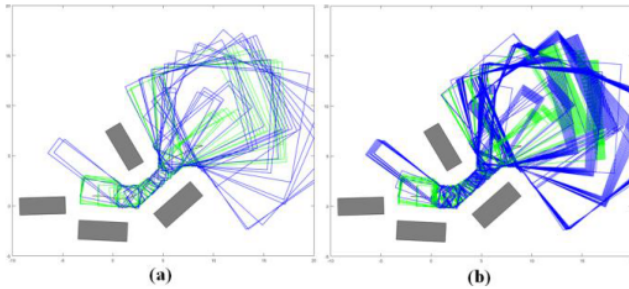


図13. ケース2の N_R を様々に設定した場合の舗装トンネル。青枠は P_R 、緑枠は P_T を表す。

最適化された軌道と我々のトンネルベースの戦略を比較すると、 N_R の変化は解にあまり変化していないことがわかり、これは提案するトンネルベースの戦略の頑健性を反映している。

V. CONCLUSIONS

本論文では、一般的な自律駐車スキームのための高速軌道プランナを提案した。大規模で複雑な衝突回避制約を定式化する一般的なプランナと比較して、車両本体と障害物を自然に分離する舗装トンネルを考慮する。提案するトンネルベースの戦略は、最適制御問題の規模を環境の複雑さに影響されないようにする。

今後の課題として、(i)移動障害物のある駐車ケースを検討する。(ii)トンネルをカバーするために、矩形ではなく、他のタイプの凸多角形を採用する可能性がある。(iii)図11(a)の最適化されたプロファイルは、ジャークに境界を課す必要性を示している。また、[15–17]で紹介された混合整数数値計画法の定式化を、解の最適性を促進する可能性について試してみる。

REFERENCES

- [1] C. Włodzimierz, and G. Iwona, “Autonomous vehicles in urban agglomerations,” *Transportation Research Procedia*, vol. 40, pp. 655–662, 2019.
- [2] B. Li and Y. Zhang, “Fast trajectory planning for off-road autonomous driving with a spatiotemporal tunnel and numerical optimal control approach,” In *Proc. 2019 IEEE International Conference on Advanced Robotics and Mechatronics (ICARM)*, pp. 924–929, 2019.
- [3] M. Pivtoraiko, and A. Kelly, “Generating near minimal spanning control sets for constrained motion planning in discrete state spaces,” In *Proc. 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3231–3237, 2005.
- [4] S. Karaman, and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
- [5] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, “Path planning for autonomous vehicles in unknown semi-structured environments,” *International Journal of Robotics Research*, vol. 29, no. 5, pp. 485–501, 2010.
- [6] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
- [7] K. Kondak and G. Hommel, “Computation of time optimal movements for autonomous parking of non-holonomic mobile platforms,” In *Proc. 2001 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, pp. 2698–2703, 2001.
- [8] K. Bergman, and D. Axehill, “Combining homotopy methods and numerical optimal control to solve motion planning problems,” In *Proc. 2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 347–354, 2018.
- [9] B. Li, and Z. Shao, “Time-optimal maneuver planning in automatic parallel parking using a simultaneous dynamic optimization approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3263–3274, 2016.
- [10] C. Liu, and M. Tomizuka, “Real time trajectory optimization for nonlinear robotic systems: Relaxation and convexification,” *Systems & Control Letters*, vol. 108, pp. 56–63, 2017.
- [11] C. Rösmann, F. Hoffmann, and T. Bertram, “Integrated online trajectory planning and optimization in distinctive topologies,” *Robotics and Autonomous Systems*, vol. 88, pp. 142–153, 2017.
- [12] X. Zhang, A. Liniger, A. Sakai, and F. Borrelli, “Autonomous parking using optimization-based collision avoidance,” In *Proc. 2018 IEEE Conference on Decision and Control (CDC)*, pp. 4327–4332, 2018.
- [13] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, and Y. Xia, “Two-stage trajectory optimization for autonomous ground vehicles parking maneuver,” *IEEE Transactions on Industrial Informatics*, vol. 15, pp. 3899–3909, 2019.
- [14] M. Babu, Y. Oza, A. K. Singh, K. M. Krishna, and S. Medasani, “Model predictive control for autonomous driving based on time scaled collision cone,” In *Proc. 2018 European Control Conference (ECC)*, pp. 641–648, 2018.
- [15] M. Vitus, V. Pradeep, G. Hoffmann, S. Waslander, and C. Tomlin, “Tunnel-MILP: Path planning with sequential convex polytopes,” In *Proc. AIAA Guidance, Navigation and Control Conference and Exhibit*, no. 7132, 2008.
- [16] J. Tordesillas, B. T. Lopez, and J. P. How, “FaSTraP: Fast and safe trajectory planner for flights in unknown environments,” In *Proc. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, accepted.
- [17] S. Liu, M. Watterson, K. Mohta, et al., “Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3D complex environments,” *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, 2017.
- [18] H. Andreasson, J. Saarinen, M. Cirillo, T. Stoyanov, and A. J. Lilienthal, “Fast, continuous state path smoothing to improve navigation accuracy,” In *Proc. 2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 662–669, 2015.
- [19] L. T. Biegler, A. M. Cervantes, and A. Wächter, “Advances in simultaneous strategies for dynamic process optimization,” *Chemical Engineering Science*, vol. 57, no. 4, pp. 575–593, 2002.
- [20] J. Ziegler, and C. Stiller, “Fast collision checking for intelligent vehicle motion planning,” In *Proc. 2010 IEEE Intelligent Vehicles Symposium (IV)*, pp. 518–522, 2010.
- [21] B. Li, and Z. Shao, “A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles,” *Knowledge-Based Systems*, vol. 86, pp. 11–20, 2015.
- [22] B. Li, Y. Zhang, T. Acarman, Q. Kong, and Y. Zhang, “Trajectory planning for a tractor with multiple trailers in extremely narrow environments: A unified approach,” In *Proc. 2019 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8557–8562, 2019.
- [23] B. Li, Y. Zhang, Y. Ge, Z. Shao, and P. Li, “Optimal control-based online maneuver planning for cooperative lane change of connected and automated vehicles,” In *Proc. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3689–3694, 2017.

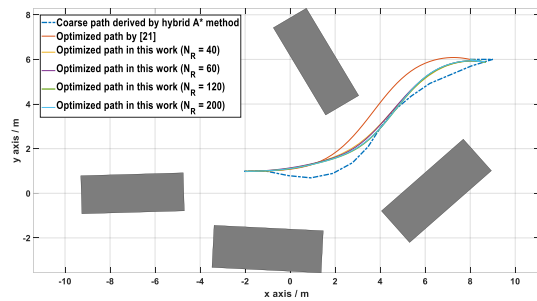


Fig. 12. Optimized trajectories with various settings of N_R (Case 2).

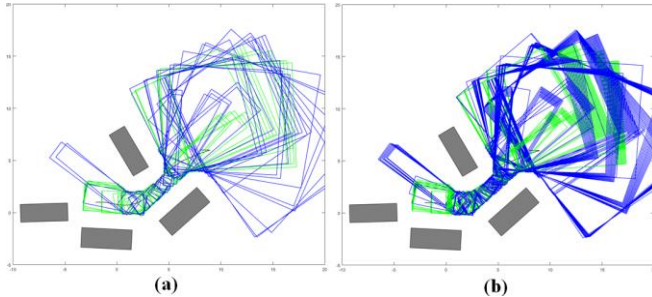


Fig. 13. Paved tunnels with various settings of N_R in Case 2. Node that the blue boxes denote the representative boxes for P_f , while green for P_r .

Through comparing among the optimized trajectories with our tunnel-based strategy, we notice that the changes in N_R do not alter much in the solutions, which reflects the robustness of the proposed tunnel-based strategy.

V. CONCLUSIONS

This paper has proposed a fast trajectory planner for generic autonomous parking schemes. Compared with the prevalent planners that formulate the large-scale and complicated collision-avoidance constraints, we consider paving tunnels which naturally separate the vehicle body from the obstacles. The proposed tunnel-based strategy makes the scale of the optimal control problem insensitive to the complexity of the environment.

As our future work, (i) the parking cases with moving obstacles will be considered; (ii) other types of convex polygons rather than rectangles may be adopted for covering the tunnels; (iii) the optimized profile in Fig. 11(a) indicates a need to impose bounds on jerk. We will also try the mixed integer mathematical programming formulations introduced in [15–17] for potential chances of promoting the solution optimality.

REFERENCES

- [1] C. Włodzimierz, and G. Iwona, “Autonomous vehicles in urban agglomerations,” *Transportation Research Procedia*, vol. 40, pp. 655–662, 2019.
- [2] B. Li and Y. Zhang, “Fast trajectory planning for off-road autonomous driving with a spatiotemporal tunnel and numerical optimal control approach,” In *Proc. 2019 IEEE International Conference on Advanced Robotics and Mechatronics (ICARM)*, pp. 924–929, 2019.
- [3] M. Pivtoraiko, and A. Kelly, “Generating near minimal spanning control sets for constrained motion planning in discrete state spaces,” In *Proc. 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3231–3237, 2005.
- [4] S. Karaman, and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

- [5] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, “Path planning for autonomous vehicles in unknown semi-structured environments,” *International Journal of Robotics Research*, vol. 29, no. 5, pp. 485–501, 2010.
- [6] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
- [7] K. Kondak and G. Hommel, “Computation of time optimal movements for autonomous parking of non-holonomic mobile platforms,” In *Proc. 2001 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, pp. 2698–2703, 2001.
- [8] K. Bergman, and D. Axehill, “Combining homotopy methods and numerical optimal control to solve motion planning problems,” In *Proc. 2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 347–354, 2018.
- [9] B. Li, and Z. Shao, “Time-optimal maneuver planning in automatic parallel parking using a simultaneous dynamic optimization approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3263–3274, 2016.
- [10] C. Liu, and M. Tomizuka, “Real time trajectory optimization for nonlinear robotic systems: Relaxation and convexification,” *Systems & Control Letters*, vol. 108, pp. 56–63, 2017.
- [11] C. Rösmann, F. Hoffmann, and T. Bertram, “Integrated online trajectory planning and optimization in distinctive topologies,” *Robotics and Autonomous Systems*, vol. 88, pp. 142–153, 2017.
- [12] X. Zhang, A. Liniger, A. Sakai, and F. Borrelli, “Autonomous parking using optimization-based collision avoidance,” In *Proc. 2018 IEEE Conference on Decision and Control (CDC)*, pp. 4327–4332, 2018.
- [13] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, and Y. Xia, “Two-stage trajectory optimization for autonomous ground vehicles parking maneuver,” *IEEE Transactions on Industrial Informatics*, vol. 15, pp. 3899–3909, 2019.
- [14] M. Babu, Y. Oza, A. K. Singh, K. M. Krishna, and S. Medasani, “Model predictive control for autonomous driving based on time scaled collision cone,” In *Proc. 2018 European Control Conference (ECC)*, pp. 641–648, 2018.
- [15] M. Vitus, V. Pradeep, G. Hoffmann, S. Waslander, and C. Tomlin, “Tunnel-MILP: Path planning with sequential convex polytopes,” In *Proc. AIAA Guidance, Navigation and Control Conference and Exhibit*, no. 7132, 2008.
- [16] J. Tordesillas, B. T. Lopez, and J. P. How, “FaSTraP: Fast and safe trajectory planner for flights in unknown environments,” In *Proc. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, accepted.
- [17] S. Liu, M. Watterson, K. Mohta, et al., “Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3D complex environments,” *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, 2017.
- [18] H. Andreasson, J. Saarinen, M. Cirillo, T. Stoyanov, and A. J. Lilienthal, “Fast, continuous state path smoothing to improve navigation accuracy,” In *Proc. 2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 662–669, 2015.
- [19] L. T. Biegler, A. M. Cervantes, and A. Wächter, “Advances in simultaneous strategies for dynamic process optimization,” *Chemical Engineering Science*, vol. 57, no. 4, pp. 575–593, 2002.
- [20] J. Ziegler, and C. Stiller, “Fast collision checking for intelligent vehicle motion planning,” In *Proc. 2010 IEEE Intelligent Vehicles Symposium (IV)*, pp. 518–522, 2010.
- [21] B. Li, and Z. Shao, “A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles,” *Knowledge-Based Systems*, vol. 86, pp. 11–20, 2015.
- [22] B. Li, Y. Zhang, T. Acarman, Q. Kong, and Y. Zhang, “Trajectory planning for a tractor with multiple trailers in extremely narrow environments: A unified approach,” In *Proc. 2019 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8557–8562, 2019.
- [23] B. Li, Y. Zhang, Y. Ge, Z. Shao, and P. Li, “Optimal control-based online maneuver planning for cooperative lane change of connected and automated vehicles,” In *Proc. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3689–3694, 2017.