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。

* 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).

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]、g² o [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すれば、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。

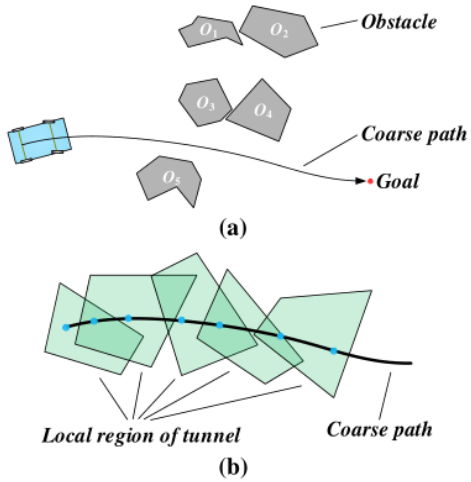


図1. (a)車両は1つのホモトピック粗経路を選択しているので、O1またはO2と衝突する可能性は低い。(b)車両が質量点であると仮定すると、従来の方法では、車両はすべての瞬間に局所領域の1つに留まるが、地上車両は質量点として扱うことができないため、地上車両が複数の局所領域に留まる可能性を記述するには、従来の方法は適用できない。

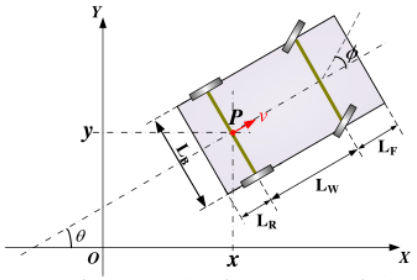


図2. 車両の運動学と幾何学に関する概略図。

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]:

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

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)は後輪軸の中点(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、*は操舵角を表し、*は対応する角速度を表し、*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:

$$\begin{aligned} &|a(t)| \le \mathrm{a_{max}} \\ &|v(t)| \le \mathrm{v_{max}} \\ &|\omega(t)| \le \Omega_{\max} \\ &|\phi(t)| \le \Phi_{\max} \end{aligned}, \; t\in[0,t_\mathrm{f}]. \qquad (2)$$

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:

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

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*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：

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

ここで、w_1, w_2 * 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：

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

B. 最適制御問題に対する数値解法

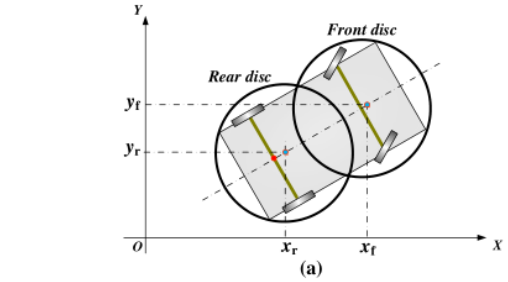
(5)の未知数は x (t), y (t), * (t), v (t), a (t), * (t), * (t), t_f である。(t), a (t), t_f が決まれば、残りのプロファイルは一意に固定される。* (t), a (t), t_f だけをシューティング法のように最適化するのではなく、状態プロファイルと制御プロファイルをすべて決定変数とみなすコロケーション法を採用する。コロケーション法は、シューティング法とは対照的に、一般的に高レベルの解の安定性を達成します[19]。

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), * (t), v (t), a (t), * (t), * (t), 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、駐車の計画軌道となる。

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*



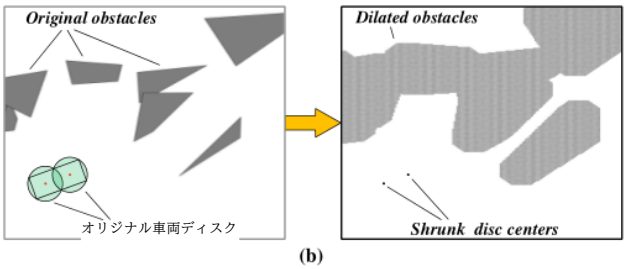


図3. 図3.車両形状の縮小と環境マップの拡張に関する模式図:(a)2枚の円盤で車両形状を提示する、(b)車両形状と環境障害物を同時に新しい形に変換する。

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、

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

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。

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

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。

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$t＊＊0$, $t_f＊＊$ 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$Traj_{Pf}$と$Traj_{Pr}$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$Traj_{Pf}$と$Traj_{Pr}$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、$Traj_{Pf}$と$Traj_{Pr}$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

f f N R $Traj_{Pf}$. これらのウェイポイントを代表ノードと呼ぶ(図4の例参照)。(N R ＊ 1)代表ノードが利用可能になると、次のステップは、代表ノードに関連する(N R ＊ 1)代表ボックスを指定することである。k番目の代表ボックスを指定する原理は以下のように導入される。

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 ＊$_f$（ t_k ）, y_f（ t_k ）＊ を $Traj_{Pf}$ に沿って指定する。

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

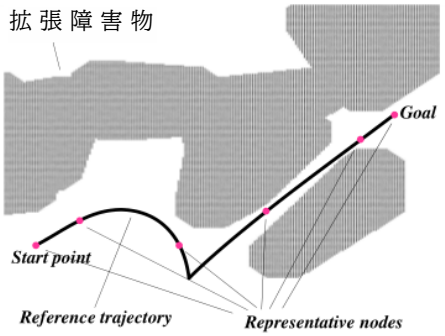


Fig. 4. Schematics on representative nodes ($N_R + 1 = 6$).

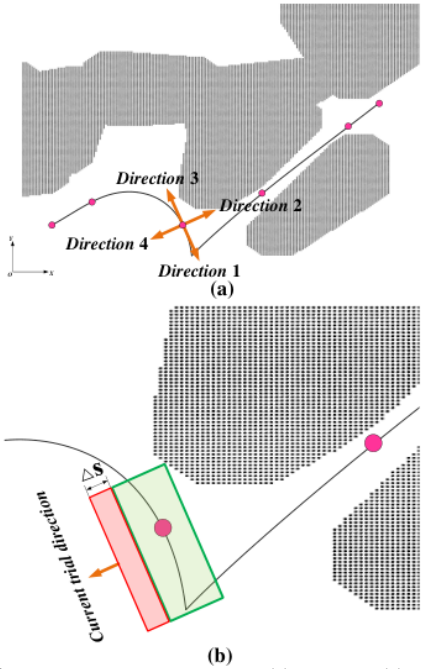


図5. 各代表ボックスを指定する原理の模式図：(a)膨張方向、(b)一方向の膨張試行。

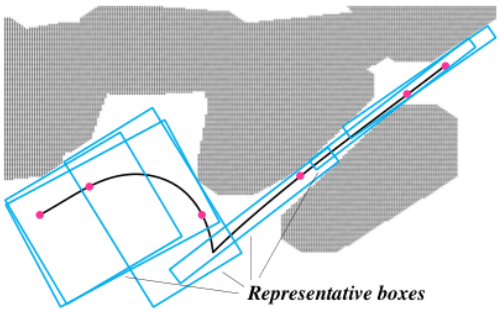


図6. 代表的な箱で形成されたトンネルの模式図。理想的には、これらの代表的なボックスは、基準軌道を完全にカバーする必要がある。

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

代表ノードをゼロ幅-ゼロ長の長方形(コア)とみなし、4方向それぞれでコアを一定のステップ*sずつ漸進的に展開する。一方向の展開試行が行われるたびに、その試行が拡張マップの障害物と衝突するかどうかをチェックする。衝突が発生しない場合、その試行は承認され、そうでない場合は拒否され、その方向への将来の展開試行はもはや考慮されない。

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:拡張写像、＊ x_f（ t_k ）, y_f（ t_k ）, ＊$_f$（ t_k ）＊, パラメータ L_{limit} ;

出力: 代表ボックスkの頂点位置。

に従って4つの展開方向を定義する ＊$_f$（ t_k ）　　）;
インデックス集合 ＊＊{1,2,3,4} を初期化する;
ベクトルを長さ ＊[0, 0, 0, 0] で初期化する;
承認領域 ＊ を質量点 ＊ x_f（ t_k ）, y_f（ t_k ）＊ として初期化する;
While $\Omega \neq \varnothing$, **do**
  For each $i \in \Omega$, **do**
    Expand Υ in direction i by Δs to form a trial box region Υ_{trial} ;
    If Υ_{trial} does not overlap with obstacles of dilated map, **then**
      Update $\textbf{length}_i \leftarrow \textbf{length}_i + \Delta s$;
      Merge Υ_{trial} into Υ ;
      If $\textbf{length}_i > L_{\text{limit}}$, then
        Remove i from Ω ;
      End if
    Else
      Remove i from Ω ;
    End if
  End for
End while
4つの頂点の位置を*に従って指定する;
Output vertex locations;
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_r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。

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

P_f（ 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。

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

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。

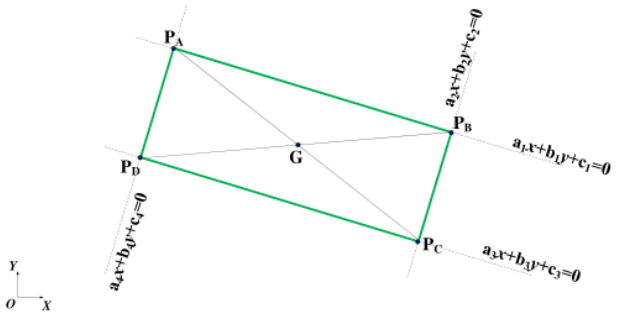


図7. 点-矩形内制約の定式化に関する模式図。

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。

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

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。

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 I 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/brQo9lPw9cw 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*（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。

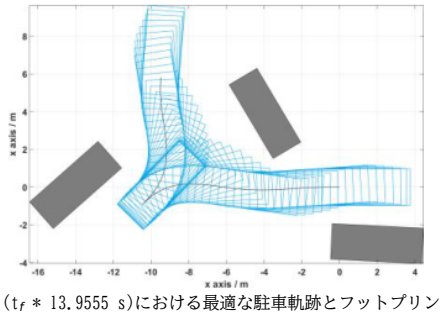


図8. ケース1(t_f * 13.9555 s)における最適な駐車軌跡とフットプリント。

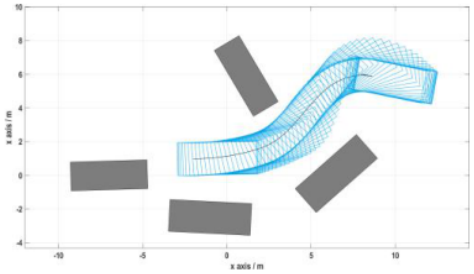


図9. ケース2(t_f * 10.3821 s)における最適な駐車軌跡とフットプリント。

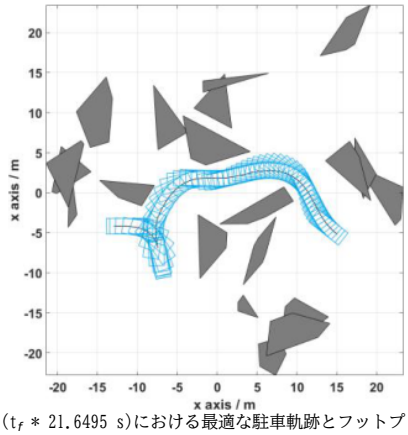


図10. ケース3(t_f * 21.6495 s)における最適な駐車軌跡とフットプリント.

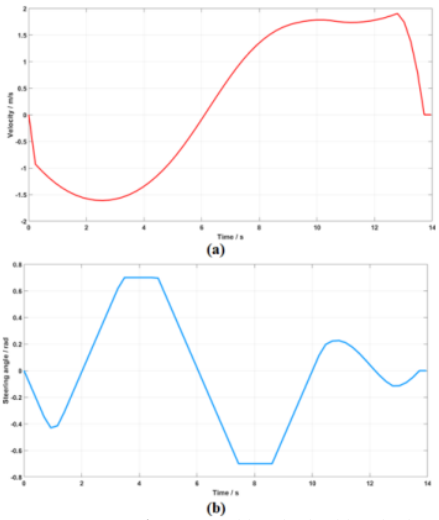


図11. ケース1の最適化プロファイル:(a) v（ｔ）, (b) *（ｔ）.

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。

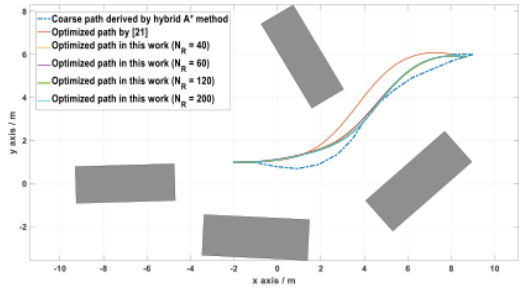


図12. N_Rを様々に設定した場合の最適化された軌道(ケース2)。

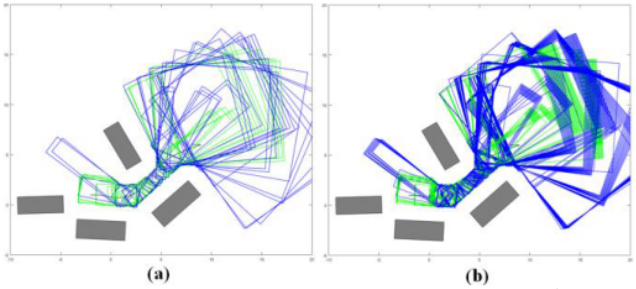


図13. ケース2のN_Rを様々に設定した場合の舗装トンネル。青枠はP_f、緑枠はP_rを表す。

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理計画法の定式化を、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.