

OCEAN:ADMMに基づく自律駐車のためのオープンスペース無衝突軌道プランナー

Dongxu Wang^{†1}, Yanbin Lu^{†1}, Weilong Liu¹, Hao Zuo¹, Jiade Xin¹, Xiang Long¹, Yuncheng Jiang²

概要- 本論文では、自律駐車のためのオープンスペース衝突-軌跡計画(OCEAN)を提案する。OCEANは、交互方向乗数法(ADMM)により加速された最適化ベースの軌道プランナーであり、計算効率とロバスト性が向上しており、動的障害物が少ない全てのシーンに適している。階層的最適化に基づく衝突回避の枠組みから出発して、軌道計画問題は、まず衝突のないハイブリッドA*軌道によってウォームスタートされ、次に衝突回避軌道計画問題は、滑らかで凸の双対形式として再定式化され、ADMMによって並列に解かれる。最適化変数は、ADMMの部分問題が2次計画法(QP)、逐次2次計画法(SQP)、2次コーン計画法(SOCP)の問題として定式化され、効率的かつロバストに解くことができるように、いくつかのグループに注意深く分割される。我々は、数百のシミュレーションシナリオと数百時間の公共駐車場の両方で、我々の方法を検証する。その結果、提案手法は他のベンチマークと比較して、より良いシステム性能を持つことがわかった。

I. INTRODUCTION

自律走行は長い間ホットな話題であり、自律駐車システムは消費者の間で関心を集めている。最近では、自動ヴァレ・パーキング(AVP)が産学双方から注目されている。より高い自律性を享受し、駐車場の入口から駐車場まで自律的に車を運転することができる。

しかし、AVPは自律駐車システムにもさらなる課題をもたらす。まず、古典的なカーブベースの駐車場プランナー[1]は、駐車場探索中の乱雑で密集した駐車場環境にもはや対処できない。第二に、非線形かつ非ホロノミックな車両ダイナミクスと自由空間の非凸性により、衝突のない軌道を計画することは非自明である。第三に、計算時間は障害物の数とともに急速に増大するため、実際のところ、ほとんどの正確な衝突評価アルゴリズムは実時間要件を満たすことが困難である。

自律駐車における衝突評価に特に注意を払う。衝突回避の評価を簡略化するには、点質量物体モデル[2]を考慮する方法があるが、このような簡略化には大きな近似誤差が必要である。無人航空機に広く使われている安全な通路は、

一連の連結した多角形(または多面体)からなる無衝突空間を生成することができる。しかし、このような表現は必然的に利用可能な運転エリアを犠牲にし、密集した乱雑な駐車環境では保守的すぎる。

駐車場計画における正確な衝突評価に取り組むためには、障害物と自車両を全次元で表現する必要があり、衝突回避問題は混合整数計画法(MIP)[3]として定式化される。MIP問題を解くための様々なアプローチがあるにもかかわらず、そのほとんどは同時要求を満たすことができない。そこで、[4]では、低次元線形計画法として定式化された全身評価を提案した。しかし、この方法は凸の障害物にしか適用できず、異なる障害物までの距離を選択的に計算することはできない。ユークリッド符号付き距離場(ESDF)も衝突評価に広く用いられている[5]が、実用上は地図解像度と計算時間のトレードオフが必要な場合が多く、ESDFはメモリや計算資源の消費量が多いため、自律駐車ではほとんど用いられていない。

ハイブリッド最適化ベースの衝突回避(Hybrid Optimization-based Collision Avoidance: HOBICA)は、追加の双対変数を導入することで、衝突回避制約を双対形式として再定式化するもので、[6]と[7]で提案されている。このアプローチは、元の非凸・非線形最適化問題を凸・非線形最適化問題に変換するが、問題の次元を劇的に増加させることは容認できない。OBICA[7]に基づく最近の改良は、Reformulated OBICA(TDR-OBICA)によるデュアルウォームスタートであり、実世界での実装において有望な結果を示している[8]。しかし、計算時間はまだ満足のいくものではありませんでした。OBICAをさらに高速化するために、[9]はADMMを採用することで、大きな集中問題をいくつかのサブ問題に分解するRegularized Dual Alternating Direction Method of Multiplier (RDA)を提案した。これらのサブ問題は分散最適化の枠組みで解くことができる。しかし、RDAは最適化変数を双対変数で分割しているため、計算効率の向上は限定的である。

要約すると、自律駐車計画の課題は、1)正確な衝突回避評価、2)アルゴリズムによるリアルタイム性能とロバスト性の要件から来る。OBICAに触発され、我々は、所定の課題に対処するために、OCEANと名付けられたロバストで加速された無衝突モーションプランナーを提示する。本手法は、衝突リスクを正確に評価することを目的としており、他のベンチマーク手法よりもリアルタイム性能とロバスト性に優れている。本論文の主な貢献は以下の通りである：

- 本手法は、モデル予測制御(MPC)フレームワークに基づき、

著者らは本論文に等しく貢献している。

^{†1}D. Wang, Y. Lu, W. Liu, H. Zuo, and J. Xin are with Mach Drive, LLC, Shanghai, China (Corresponding author: H. Zuo hao.zuo@mach-drive.com).

² Y. Jiang is with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong(email: yjiangbw@connect.ust.hk).

非ホロノミックな車両ダイナミクスと制御限界を問題定式化に埋め込むものである。二重凸の再定式化を活用することで、非凸の衝突回避制約を凸制約として再定式化する。

- ADMMは、元の非線形計画問題を一連の修正QP、SQP、SOCP問題に分解するために採用される。これらの問題は、完全に分散化された並列アーキテクチャで解くことができるため、計算負荷が大幅に軽減され、リアルタイム性能が保証される。
- H-OBICA、TDR-OBICA、RDAがウォームスタートの結果に大きく依存するのとは異なり、本手法は双対変数衝突回避問題をSOCP問題として再定式化し、ADMM構造で反復的に解く。本手法ではウォームスタートは必要ない。
- 状態遷移における非線形性に対処するために、それぞれの部分問題における非線形制約を線形化するための追加の最適化変数と制約を導入する。
- 我々は、数百のシミュレーションシナリオと数百時間の公共駐車場の両方で、アルゴリズムの効率と頑健性を検証する。本論文が発表されるとすぐに、提案する駐車アルゴリズムのソースコードが、本論文が発表されるとすぐに公開される。

本稿の構成は以下の通りである。セクションIIでは、関連する研究をレビューする。セクションIIIでは、軌道最適化問題と、衝突のない制約の二重再定式化について述べる。OCEANの主なアルゴリズムはセクションIVに示す。セクションVではシミュレーションと実環境でのテスト結果を示し、セクションVIで結論を述べる。

II. 関連研究

自律走行計画アルゴリズムの分野では、MPCに基づく軌道計画法が広く推進され、適用されている[10]。しかし、多障害物駐車環境では、非凸の衝突回避制約が軌道最適化アルゴリズムに課題をもたらす。現在、軌道最適化に適用可能な障害物モデリング手法がいくつかある：

A. 楕円体または球近似

最初のアプローチは、2Dでは楕円体、3Dでは球として障害物をモデル化する[11]。この方法は、障害物距離の直接計算と障害物から離れる勾配方向の決定が可能であるため、直感的でシンプルである。しかし、車両と障害物の間の距離を正確に表現することはできない。

B. Safe Corridor

第二のアプローチは、無人航空機で一般的に採用されている安全な通路[12][13]の概念を利用し、障害物空間を凸多角形制約空間として再定義する。このアプローチは、凸空間の範囲内で軌道の安全性を保証する。しかし、これらのコリドーの構築にかかるコストは、障害物の離散化の解像度と、コリドー拡張時に使用されるステップサイズに依存する。

その結果、これらの要因により、実際の障害物空間を保守的に推定することができる。さらに、車両の軌跡が障害物に近接している場合、高品質な通路の構築にはコストがかかる。

C. ダブルサイン距離障害物表現

本研究の第3のアプローチは、二重符号付き距離障害物表現と二重凸再構成を採用している[14]。このアプローチでは、障害物は凸多面体形状としてモデル化され、車両と障害物間の距離は符号付き距離で表される。この表現を用いることで、各障害物までの最小距離 d_{min} を直接取り込むことができる。次に、二重凸再定式化を活用することで、衝突回避制約を非線形だが凸制約に変換する。車両モデルは、点質量または凸多面体のいずれかである。しかし、二重符号付き距離障害物表現は、問題にかなりの数の等式制約と不等式制約を導入し、それによってその次元を増加させる。N個のサンプリング点とM個の障害物を持つ点群システムシナリオでは、 N^M 個の双対変数を導入する。制御されたシステムの形状を考慮すると、問題の次元は劇的に増加し、その結果、さらに $2N^M$ の線形不等式制約が生じる。

D. 線形計画法

第4の方法は、全身衝突における車両平面と障害物平面間の距離の最適化問題を線形計画法(LP)問題として定式化する方法である[4]。これにより、障害物の数に関係なく、軌道の安全性を確保するために1つの変数のみを使用することで、計算効率を大幅に向上させることができる。しかし、この方法は凸の障害物にしか適用できず、異なる障害物までの距離を選択的に計算することはできない。また、主に多項式表現に基づく軌道最適化アルゴリズムと組み合わせている。

E. ユークリッド符号付き距離場

第五の方法は、ESDF [5]やリスクフィールド [15]などのポテンシャルフィールドアプローチに基づくものである。しかし、実用化には解像度と計算時間のトレードオフが必要になることが多い。近年、SDF Any-Shape法[16]が提案され、時空間連続性を利用することで、ロボットとその軌跡が形成する走査体積の符号付き距離を、遅延かつ効率的な方法で暗黙的に計算する。この方法は、主に多項式表現に基づく軌道最適化問題と組み合わされている。しかし、時間に対して微分可能な軌跡であることが必要である。

F. 最適化アルゴリズムの高速化

既存のアプローチは、主にアルゴリズム設計の側面から計算時間の短縮に焦点を当てている。これらのアプローチは、ヒューリスティック、近似、並列計算、学習、エッジコンピューティングなどの様々な技法を取り入れている[17]。

本論文は、ウォームスタート問題を効果的に定式化し、ADM [18]によって集中最適化問題を分解することで、問題解決を加速することを目的としている。ADMMは、簡単で強力なアルゴリズムであり、多くの分野で広く使われていることは注目に値する[18][19]。障害物回避に並列計算を適用することで、ADMMは乱雑な環境でも自律的なナビゲーションを加速することが示されている[9]。衝突回避制約は非凸であり、ADMMは凸問題にしか適用できないので、ADMMを適用する前に、元の問題を凸化するために適切な近似と再定式化を行うべきである。

III. 問題定式化

OBCAに触発され、我々はMPCフレームワークで駐車場計画問題を定式化し、元のMIP問題が非線形で凸制約を持つMPC問題に変換されるように、衝突回避制約をその双対形式として再定式化する。

A. 車両動的モデル

自転車モデルを用いて自我の状態を表す。ここで、 X_k と Y_k は車両の横位置と縦位置、 φ_k は方位角、 v_k は車両速度である。制御指令は $u_k = [\delta_k, a_k]$ であり、 δ_k はステアリング、 a_k は加速度である。離散的な形の車両ダイナミクスは次式で与えられる。

$$\begin{aligned} X_{k+1} &= X_k + v_k t_k \cos \varphi_k, \\ Y_{k+1} &= Y_k + v_k t_k \sin \varphi_k, \\ \varphi_{k+1} &= \varphi_k + v_k t_k \frac{\tan \delta_k}{L}, \\ v_{k+1} &= v_k + a_k t_k, \\ \forall k &= 1, \dots, N, \end{aligned} \quad (1)$$

ここで、 t_k は可変時間ステップ、 L はホイールベースである。

B. フルディメンションモデリングと衝突回避

読者の便宜のため、OBCAと同じ表記に従う。詳細は[6] (Sec II)を参照されたい。与えられた状態 x_k に対して、自車両が占有する空間を $E(x_k) \sim \mathcal{R}_R^{(m)} \sim \mathcal{R}_O^{(m)}$ 、 $\forall m = 1, \dots, M$ は障害物が占める空間である。自車両の初期矩形 B と回転・平行移動矩形 $E(x_k)$ は次式で定義される。

$$\mathbb{E}(x_k) = R(x_k)\mathbb{B} + t(x_k), \quad \mathbb{B} := \{y : Gy \leq g\}. \quad (2)$$

ここで、 G と g は初期矩形 B を定義し、既知であると仮定する。同様に、障害物も同様に定義される

$$\mathbb{O}^{(m)} = \{y \in \mathbb{R}^2 : A^{(m)}y \leq b^{(m)}\}, \forall m = 1, \dots, M \quad (3)$$

無衝突制約は $E(x_k) \cap \mathbb{O}^{(m)} = \emptyset$, $\forall m = 1, \dots, M$ であり、非線形で非凸である。符号付き距離[14]を用いて、衝突回避を解析的に定式化する。

$$\text{sd}(\mathbb{E}(x), \mathbb{O}) := \text{dist}(\mathbb{E}(x), \mathbb{O}) - \text{pen}(\mathbb{E}(x), \mathbb{O}), \quad (4)$$

ここで、距離関数と侵入関数はそれぞれ次のように定義される。

$$\text{dist}(\mathbb{E}(x), \mathbb{O}) := \min_t \{\|t\| : (\mathbb{E}(x) + t) \cap \mathbb{O} \neq \emptyset\}, \quad (5a)$$

$$\text{pen}(\mathbb{E}(x), \mathbb{O}) := \min_t \{\|t\| : (\mathbb{E}(x) + t) \cap \mathbb{O} = \emptyset\}. \quad (5b)$$

7]によれば、衝突回避制約は、以下のように再定式化される追加の双対変数を導入することによって、その双対形式に変換することができる：

$$\begin{aligned} \text{sd}(\mathbb{E}(x), \mathbb{O}) &> d \\ \iff \exists \lambda \geq 0, \mu \geq 0 : -g^\top \mu + (A^\top(x) - b)^\top \lambda &> d, \\ G^\top \mu + R(x)^\top A^\top \lambda &= 0, \|A^\top \lambda\| = 1. \end{aligned} \quad (6)$$

次に、非凸制約(6)を凸制約として修正し、簡略化する。衝突のないロバストなウォームスタート軌道があるので、このような修正は妥当である：

$$\begin{aligned} \text{sd}(\mathbb{E}(x), \mathbb{O}) &> d \\ \iff \exists \lambda \geq 0, \mu \geq 0 : -g^\top \mu + (A^\top(x) - b)^\top \lambda &> d, \\ G^\top \mu + R(x)^\top A^\top \lambda &= 0, \|A^\top \lambda\| \leq 1. \end{aligned} \quad (7)$$

C. Problem Statement

MPCの枠組みで自律駐車問題を定式化すると、離散形式での状態進化は以下のようになる。

$$x_{k+1} = A_k x_k + B_k u_k + c_k, \quad \forall k = 1, \dots, N, \quad (8)$$

ここで、 A_k と B_k は(1)から容易に得られる係数行列である。また、制御 $u_{\min} u_k u_{\max}$ に対する制約を強制し、高次制御も有限離散化 $u_{\min} u_k u_{\max}$ によって極限まで適用できる($u_k = u_{k+1} - u_k$)。状態制約は k が必要ときにも適用できる。これらの制約を次のように組み合わせる。

$$h(x_k, u_k, u_{k-1}) \leq 0, \forall k = 1, \dots, N. \quad (9)$$

最適化問題の目的は、スムーズで安全な駐車軌道を計画することである。したがって、目的関数を以下のように定義する。

$$\begin{aligned} \mathcal{L}(x_k, u_k, t_k) &= w_x \sum_{k=1}^N \|x_k - \bar{x}_k\|_2^2 \\ &+ w_u \sum_{k=1}^N \|u_k\|_2^2 \\ &+ w_{x'} \sum_{k=1}^N \frac{\|x_k - x_{k-1}\|_2^2}{t_k} \\ &+ w_{\bar{u}} \sum_{k=1}^N \frac{\|u_k - \bar{u}_{k-1}\|_2^2}{t_k}, \end{aligned} \quad (10)$$

ここで、 x_{302} はウォームスタートからの参照状態を表し、 \bar{u} は前のステップ制御コマンドを表す。(10)において、第1項はウォームスタート軌道との状態差にペナルティを与える。第2項は、管理努力にペナルティを与える。第3項は2次の滑らかさを測定し、第4項は連続する最適化ループ間の状態差にペナルティを与える。

このため、駐車場計画は、制約条件(7)(8)(9)に従い、目的関数(10)を持つ最適化問題として定式化することができる。

しかし、このような定式化では、衝突回避のための下限を設定するだけであり、駐車軌跡は可能な限り障害物から離れることになっている。したがって、(7)のdを弛緩変数として導入し、指数M(d) = e^dの目的関数にプラグインする。ここで、最適化問題の正確な形を与えることができる。

$$\min_{x,u,d,t,\lambda,\mu} \sum_{k=1}^N \mathcal{L}(x_k, u_k, t_k) + w_d \sum_{k=1}^N \sum_{m=1}^M \mathcal{M}(d_{k,m}), \quad (11a)$$

$$s.t. \quad x_0 = x_S, x_{N+1} = x_F, \quad (11b)$$

$$x_{k+1} = f(x_k, u_k), \quad (11c)$$

$$h(x_k, u_{k-1}) \leq 0, \quad (11d)$$

$$d_{k,m} \leq 0, \quad (11e)$$

$$-g^T \mu_{k,m} + (A_m t_k - b_m)^T \lambda_{k,m} + d_{k,m} = 0, \quad (11f)$$

$$G^T \mu_{k,m} + R(x_k)^T A_m^T \lambda_{k,m} = 0, \quad (11g)$$

$$\|A_m^T \lambda_{k,m}\|_* \leq 1, \quad (11h)$$

$$\lambda_{k,m} \succeq_{k^*} 0, \mu_{k,m} \succeq_{\bar{k}^*} 0, \quad (11i)$$

$$\forall k = 1, \dots, N, m = 1, \dots, M,$$

ここで、w_dはdのペナルティ重み、x_Sは初期状態、x_Fは最終状態である。

IV. 衝突のない軌道プランナ

本節では、我々の並列計算フレームワークを紹介する。多数の双対変数を導入した軌道計画問題を高速化するために、元の問題をいくつかのサブ問題に注意深く分割し、各問題を並列に解くことができるようにした。そして、各サブ問題は、効率的かつロバストに解くことができるQP、SOCP、SQP問題としてよく整理されている。

A. 並列計算

ADMMを用いて(11)を解く。各反復は、系列凸部分問題を解くことを含む。各反復において、双対変数と決定変数の両方が、問題を効率的に解くことができ、変数を並行して更新できるように、いくつかのサブ問題に注意深く分割される。まず、(11)の拡張ラグランジアンを次のように与える。

$$\begin{aligned} \mathcal{L}_\rho(x, u, t, d, \lambda, \mu) = & \sum_{k=1}^N \mathcal{L}(x_k, u_k, t_k) + \sum_{k=1}^N \sum_{m=1}^M \mathcal{L}(d_{k,m}) \\ & + \sum_{k=1}^N I_m(x_k, u_k, t_k) \\ & + \sum_{k=1}^N \sum_{m=1}^M I_{k,m}(\lambda_{k,m}, \mu_{k,m}, d_{k,m}) \\ & + \frac{\rho}{2} \sum_{k=1}^N \|f(x_k, u_k) - x_{k+1} + \eta_k\|_2^2 \\ & + \frac{\rho}{2} \sum_{k=1}^{N+1} \sum_{m=1}^M \|G_{k,m}(x_k, \lambda_{k,m}, \mu_{k,m}, d_{k,m}) + \zeta_{k,m}\|_2^2 \\ & + \frac{\rho}{2} \sum_{k=1}^{N+1} \sum_{m=1}^M \|H_{k,m}(x_k, \lambda_{k,m}, \mu_{k,m}) + \xi_{k,m}\|_2^2 \end{aligned} \quad (12)$$

ここで $\eta_k, \zeta_{k,m}, \xi_{k,m}$ は等式制約(11c) (11f), (11g)に対応する双対変数であり、 ρ はラグランジュ関数のペナルティ・パラメータである。(11b) (11d)を $I_m(x_k, u_k, t_k)$ と書き直すと、(11h) (11i)を $I_{k,m}(\lambda_{k,m}, \mu_{k,m}, d_{k,m})$ と書き直す。 $G_{k,m}(\cdot)$ と $H_{k,m}(\cdot)$ は(11f)と(11g)の簡略形である。

$$G_{k,m}(x_k, \lambda_{k,m}, \mu_{k,m}, d_{k,m}) = d_{k,m} - g^T \mu_{k,m} + (A^T(x_k) - b_m)^T \lambda_{k,m}, \quad (13)$$

$$H_{k,m}(x_k, \lambda_{k,m}, \mu_{k,m}) = G^T \mu_{k,m} + R(x_k)^T A_m^T \lambda_{k,m}. \quad (14)$$

9]では、ADMMを用いて同様の駐車計画問題を解決しているが、変数は2つのグループに分解されるだけであり、並列計算は効率的でない。変数を4つのグループ $\{\lambda, \mu, v, a, t, x, y, \phi, \delta, d, \sin \phi, \cos \phi, \tan \phi\}$ に分解し、各サブ問題を効率的かつロバストに解くことができるようにする。なお、双対変数 $\{\lambda, \mu\}$ は衝突回避制約にのみ関係し、(11f)によれば、弛緩変数dは双対変数と密接な関係がある。そこで、まず、二重変数と弛緩度変数を分割する。簡単のため、添え字 $(\cdot)_{k,m}$ は省略し、上付き添え字 $(\cdot)^i$ と $(\cdot)^{i+1}$ はそれぞれ現在のステップ変数と次のステップ変数を示す。

$$\begin{aligned} \{\lambda^{i+1}, \mu^{i+1}, d^{i+1}\} = & \arg \min_{\lambda, \mu, d} \mathcal{L}_\rho(\{\lambda, \mu\}, \{v^i, a^i\}, \{t^i\}, \\ & \{x^i, y^i, \phi^i, \delta^i, d^i, \sin \phi^i, \cos \phi^i, \tan \phi^i\}). \end{aligned} \quad (15)$$

(15)を解くには、双対変数 $\mu_{k,m}$ と $\lambda_{k,m}$ をウォームスタートさせることでSQPを使うことができる。(7)より、双対変数は衝突回避制約にのみ関係することが示される、すなわち(11f)-(11i)であるため、 $\mu_{k,m}$ と $\lambda_{k,m}$ をスラックネス変数dを導入して計算し、衝突回避制約を持つウォームスタート問題を定義することができる。

$$\begin{aligned} \min_{\mu, \lambda, d} \quad & \sum_{m=1}^M \sum_{k=1}^K d_{m,k} \\ s.t. \quad & -g^T \mu_{m,k} + (A_m t_k - b_m)^T \lambda_{m,k} + d_{m,k} = 0, \\ & G^T \mu_{m,k} + R(\bar{x}^*(k))^T A_m^T \lambda_{m,k} = 0, \\ & \lambda_{m,k} \succeq 0, \mu_{m,k} \succeq 0, d_{m,k} < 0, \|A_m^T \lambda_{m,k}\|_2 \leq 1 \\ & \forall k = 1, \dots, K, m = 1, \dots, M, \end{aligned} \quad (16)$$

そして(16)は(15)と等価である。なお、(11h)は2次の錐制約であり、ウォームスタート問題を単純化するために、[8]は制約 $\|A_m^T \lambda_{m,k}\|_2 \leq 1$ を目的関数に入れる。すると、ウォームスタート問題は次式で与えられる。

$$\begin{aligned} \min_{\mu, \lambda, d} \quad & \frac{1}{\beta} \sum_{m=1}^M \|A_m^T \lambda_m(k)\|_2^2 + \sum_{m=1}^M \sum_{k=1}^K d_{m,k} \\ s.t. \quad & -g^T \mu_{m,k} + (A_m t_k - b_m)^T \lambda_{m,k} + d_{m,k} = 0, \\ & G^T \mu_{m,k} + R(\bar{x}^*(k))^T A_m^T \lambda_{m,k} = 0, \\ & \lambda_{m,k} \succeq 0, \mu_{m,k} \succeq 0, d_{m,k} < 0, \\ & \forall k = 1, \dots, K, m = 1, \dots, M, \end{aligned} \quad (17)$$

ここで¹はペナルティ重みであり、元の問題は β をQP問題とする。QP問題を定式化することで双対変数をウォームスタートさせるのは効率的であるが、

(17)は元の最適化問題(16)を変更する。その結果、(17)を解くことは必ずしも厳密な衝突回避を保証しないかもしれない。OCEANでは、(15)をSOCPにポーズするが、これは既存の原始-双対内点法(PMIPM)で解くことができ、ウォームスタートに依存しない。SOCPのIPMアルゴリズムは、少なくとも局所最適に収束することが保証されている[20]。(15)において、最適化変数 $\{\lambda^{i+1}, \mu^{i+1}, d^{i+1}\}$ は幾何学的に明示的に解釈されており、問題をSOCPとして提起することで、よりロバストで正確な解を期待できる。21][22]によれば、決定論的収束特性を持つSOCPソルバーがあり、これは問題を多項式時間で解くことができ、大域的最適値は反復回数の決定論的上限で計算できることを意味する。

車両の動的制約(11c)が非線形である場合、決定変数 v , a を分割して、(11c)が線形制約に変換されるようにする。

$$\{v^{i+1}, a^{i+1}\} = \arg \min_{v, a} \mathcal{L}_p(\{\lambda^{i+1}, \mu^{i+1}\}, \{v, a\}, \{t^i\}, \{x^i, y^i, \phi^i, \delta^i, d^i, \sin \phi^i, \cos \phi^i, \tan \delta^i\}) \quad (18)$$

同様に、変数 t は非線形制約 (11c) と (11f) にのみ関係するので、(11c) と (11f) が線形になるように t を分割する。

$$\{t^{i+1}\} = \arg \min_t \mathcal{L}_p(\{\lambda^{i+1}, \mu^{i+1}\}, \{v^{i+1}, a^{i+1}\}, \{t\}, \{x^i, y^i, \phi^i, \delta^i, d^i, \sin \phi^i, \cos \phi^i, \tan \delta^i\}) \quad (19)$$

最適化問題決定変数は $\{x, y, \phi, \delta, d\}$ であり、車両の動的制約(11c)を線形化するために、 $\{\sin \phi, \cos \phi, \tan \delta\}$ を導入し、線形化を保証するために追加の制約を強制する。一方、(11g)において、回転行列 $R(x_k)$ は非線形項を含み、 $\{\sin \phi, \cos \phi\}$ (11g)を導入することでも線形制約に変換される。 $X := \{x, y, \phi, \delta, d, \sin \phi, \cos \phi, \tan \delta\}$ と定義し、 X を次のように分割する。

$$\mathcal{X} = \begin{bmatrix} x \\ y \\ \phi \\ \delta \\ d \end{bmatrix} \quad (20)$$

次に、 $\{\phi, \delta\}$ と $\{\sin \phi, \cos \phi, \tan \delta\}$ の相関を示すために、一次テイラー展開を用いて、追加の弛緩制約を導入する。上付き文字 $(.^j)$ と $(.^{j+1})$ を導入し、それぞれSQP反復における現在のステップ変数と次のステップ変数を示す。具体的には、 $\{\sin \phi, \cos \phi, \tan \delta\}$ は、最後の最適化ループからそれぞれの変数の周りに次のように線形化される。

$$\begin{aligned} \sin \phi^{j+1} - \phi^{j+1} \cos \phi^j &= \sin \phi^j - \phi^j \cos \phi^j \\ \cos \phi^{j+1} + \phi^{j+1} \sin \phi^j &= \cos \phi^j + \phi^j \sin \phi^j \\ \tan \delta^{j+1} - \delta^{j+1} \sec^2 \delta^j &= \tan \delta^j - \delta^j \sec^2 \delta^j \end{aligned} \quad (21)$$

(15)と(20)の両方が d を更新できるため、弛緩度変数 d を完全に分割することはできないので、収束を良くするために d を両方で更新する。

最後に、ADMM の二重変数を以下のように更新する。

$$\begin{aligned} \eta_k^{i+1} &= \eta_k^i + f(x_k^{i+1}, u_k^{i+1}) - x_{k+1}^{i+1} \\ \xi_{k,m}^{i+1} &= \xi_{k,m}^i + H_{k,m}(x_k^{i+1}, \lambda_{k,m}^{i+1}, \mu_{k,m}^{i+1}), \\ \zeta_{k,m}^{i+1} &= \zeta_{k,m}^i + G_{k,m}(x_k^{i+1}, \lambda_{k,m}^{i+1}, \mu_{k,m}^{i+1}, d_{k,m}^{i+1}) \end{aligned} \quad (22)$$

アルゴリズム1 OCEANプランナーアルゴリズム

```

1: ハイブリッドA*軌道によるウォームスタート状態と制御変数
2: for iteration  $k = 1, 2, \dots$ , do
3:   Update variables  $\{\lambda^{i+1}, \mu^{i+1}, d^{i+1}\}$  by solving (15)
4:   Update variables  $\{v^{i+1}, a^{i+1}\}$  by solving (18)
5:   Update variables  $\{t^{i+1}\}$  by solving (19)
6:   Update variables  $\{\mathcal{X}^{i+1}\}$  by solving (20) with addition constraints (21)
7:   Update Lagrangian multipliers  $\eta_k, \xi_{k,m}, \zeta_{k,m}$  by (22)
8:   if (23a) and (23b) are satisfied then
9:     break
10:  end if
11: end for

```

反復を終了するための停止基準は以下のように与えられる。

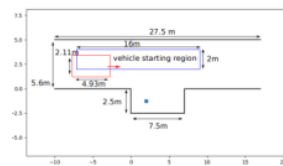
$$\sum_{k=1}^N \|f(x_k, u_k) - x_{k+1}\|_2^2 + \sum_{k=1}^N \sum_{m=1}^M \|G_{k,m}(x_k, \lambda_{k,m}, \mu_{k,m}, d_{k,m})\|_2^2 + \sum_{k=1}^N \sum_{m=1}^M \|H_{k,m}(x_k, \lambda_{k,m}, \mu_{k,m})\|_2^2 \leq \epsilon^{\text{pri}}, \quad (23a)$$

$$\sum_{k=1}^N \|\eta_k^{i+1} - \eta_k^i\|_2^2 + \sum_{k=1}^N \sum_{m=1}^M \left[\|\xi_{k,m}^{i+1} - \xi_{k,m}^i\|_2^2 + \|\zeta_{k,m}^{i+1} - \zeta_{k,m}^i\|_2^2 \right] \leq \epsilon^{\text{dual}}. \quad (23b)$$

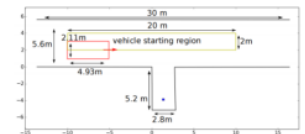
ここで、(23a)は素残差が小さいことを保証し、(23b)は双対残差が小さいことを保証する。アルゴリズムはAlg. 1.

V. シミュレーションと結果

このセクションでは、シミュレーションと実世界の道路テストの両方でOCEANを検証する。まず、同じシナリオ設定と初期状態の下で、OCEANとベンチマーク手法、すなわちV-BのH-OBICAとTDROBCAの比較を行い、OCEANが効率的かつ頑健に動作することを示す。V-Cでは、異なるシナリオと初期状態の下でOCEANをシミュレートし、V-DではOCEANの実世界でのテスト結果を示す。このコードはC++で開発され、クロック速度2.10GHzのIntel(R) Xeon(R) Gold 6130 CPUプロセッサにデプロイされている。



(a) 平行駐車シーン



(b) Vertical Parking Scene

図1: ベンチマークシナリオ

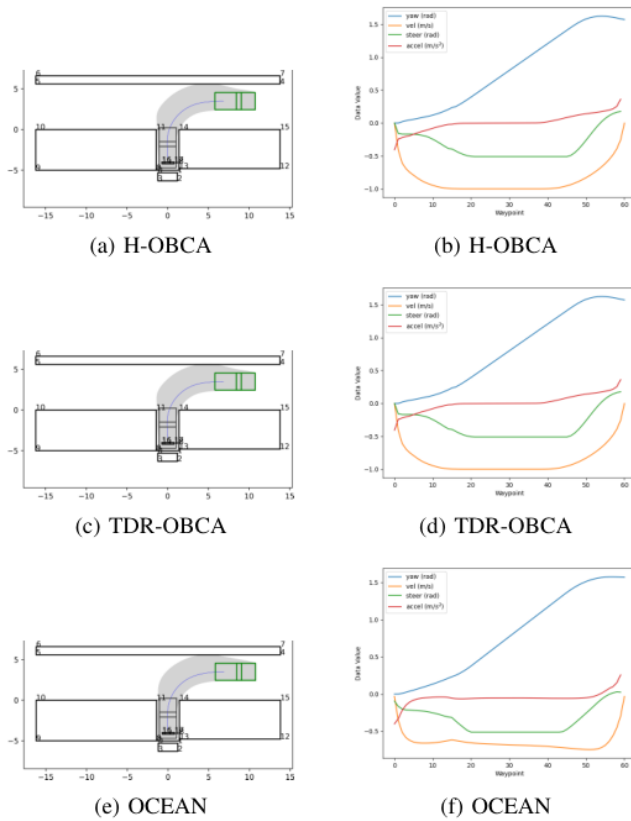


図2: 構文解析軌跡の可視化と運動学的パラメータのサンプリング。

A. ベンチマークシミュレーション結果

公正な比較のために、他のベンチマーク手法のシミュレーション結果を最大限に復元することを試みる。図1aに示すように、[23]と同じ並列駐車シナリオを設定し、H-OBCAのテストを行った。その結果、システムの性能と消費時間は[23]のものと一致した。TDR-OBCAでは、CasAdi自動微分とIPOPT(線形ソルバーとしてMa27を使用)を用いてNLP問題を解き、OSQPを用いて双対問題をウォームスタートさせる。その結果、ウォームスタート問題は平均13msで解け、NLP問題は1.39sで解ける。全体として、TDR-OBCAは並列駐車シナリオにおいてH-OBCAより11.5%の効率改善を達成し、H-OBCAとTDR-OBCAの時間消費は[8][23]と同様である。

B. ベンチマーク手法との比較

OCEANとH-OBCA、TDR-OBCAを比較するために、[8][23]で説明したものと同一垂直駐車シナリオを図1bに示すように設定した。一方、最適化結果が初期推測に影響されないように、全ての手法で同じフロントエンドラフパス生成技術を使用する。また、曲率が継続する経路を生成する改良型ハイブリッドA*法を採用する[24]。

表1: 3つのアルゴリズムの比較: H-OBCAと比較した失敗率、実行時間の比較、時間削減率。

Algorithm	Failure Rate	Runtime(ms)	Reduced Rate
H-OBCA	0%	2723	
TDR-OBCA	0%	2431	11.5%
OCEAN	0%	233	91%

表 II: 成功例に対する軌道評価結果。

		H-OBCA	TDR-OBCA	OCEAN
Steering Angle (rad)	Mean	-0.3382	-0.3382	-0.3487
	Max	0.1753	0.1753	0.0292
	Min	-0.5105	-0.5105	-0.5105
	Std Dev	0.2102	0.2102	0.1734
Velocity (m/s)	Mean	-0.8556	-0.8556	-0.6383
	Max	0.0000	0.0000	-0.0300
	Min	-1.0000	-1.0000	-0.7466
	Std Dev	0.2361	0.2361	0.1382
Acceleration (m/s ²)	Mean	0.0095	0.0095	-0.0529
	Max	0.3611	0.3611	0.2585
	Min	-0.4057	-0.4057	-0.4000
	Std Dev	0.1265	0.1265	0.0851

提案手法は、制御の実現可能性と計算効率の両方において大きな利点を示す。シミュレーションには、自車両の初期位置が異なる80のシナリオが含まれる[8]。初期位置は、図1bに示すように、黄色のグリッドから一様にサンプリングされる。初期方位角は右を向くゼロに設定され、最終的な駐車位置はすべてのシナリオで一定である。効率性を実証するために、3つのアルゴリズムの実行時間を比較し、その結果を表.Iに示す。

実験によると、ハイブリッドA*探索が衝突のない初期結果を提供する限り、H-OBCAとTDRアルゴリズムは最適化フェーズで一般的に失敗しないことがわかった。主な違いは、デュアル問題を解くためにウォームスタート技術を使用することによって達成された解答時間の11.5%削減にある。

ロバスト性に加えて、アルゴリズムの制御出力も比較する。図2は、80ケースのうち、典型的なシナリオにおける各アルゴリズムの出力を示している。軌道プロットとそれに対応する運動曲線プロットから、OCEANはH-OBCAやTDR-OBCAで得られた軌道とほぼ同じ軌道を生成することがわかる。

図2の補足として、表IIの最小、最大、平均の制御出力を比較する。平均して、OCEANはTDR-OBCAと比較して、加速度、速度、操舵角の分散が小さい軌道を生成しており、生成された軌道がより滑らかな特性を持つことを示している。

C. 海洋シミュレーション結果

異なる駐車シナリオでOCEANをシミュレートするために、自然のシーンから抽出された平行、垂直、

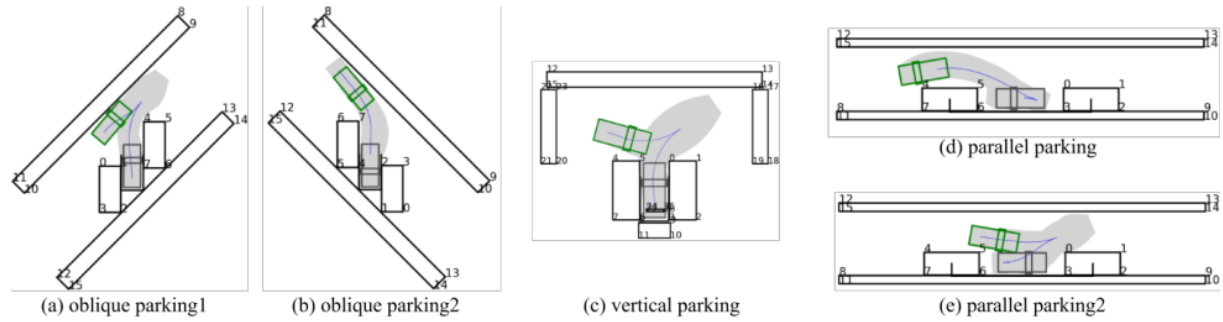


図3:シナリオを含むシミュレーションにおける計画軌道: 1)斜め、2)垂直、3)平行。

斜めの駐車スペースをテスト用を選択する。表 III:成功したケースに対する軌跡の評価結果。

Case ID	(a)	(b)	(c)	(d)	(e)
N_{OA}	3	4	6	10	14
N_{OB}	3	3	5	8	10
$t_{f,TDR}(ms)$	1027.07	1395.43	1459.12	1691.88	1897.96
$t_{f,OCEAN}(ms)$	164.42	159.84	283.44	319.24	325.30
Improved Rate	87.61%	85.41%	84.75%	64.21%	78.28%

表 IV: OCEAN における部分部分問題の計算時間。obs var は (15) の双対変数衝突回避問題を解くこと、speed pro は (18) の速度・加速度問題を解くこと、time interval は (19) の最適時間間隔問題を解くこと、path pro は (20) の MPC パスプラン問題を解くことを表す。

	obs var	speed pro	time interval	path pro
time(ms)	104.66	1.3	0.74	7.37

の障害物がある。6~7個の障害物がある116個の並列駐車ケース、8~10個の障害物がある139個の垂直駐車ケース、105個の斜め駐車ケースがある。各テストケースにおいて、他車両と自車両の位置と姿勢はランダムに選択される。これらのケースのいくつかを図3に示す。

また、解答時間が障害物の数によってどのように変化するかについても検討した。 N_{OA} は、非貫通境界である車両と障害物の数を表し、 N_{OB} は、最適化空間が大きくなりすぎた場合に軌道生成領域を制限するために使用される、追加的に追加された仮想障害物と仮想境界の数を表す。表. IIIは、H-OBBCA、TDR-OBBCA、OCEANの異なるシナリオでの解答時間と、障害物と境界の数を示している。5つのテストシナリオにおいて、OCEANは少なくとも60%の解答時間短縮を達成した。

360以上のテストケースを実施し、結果をヒストグラムで図4に示す。計画問題は、ウォームスタート、最適化問題の構築、最適化問題の解法から構成される。IV-Aで述べたように、OCEANではウォームスタートは不要であり、TDR-OBBCAとH-OBBCAはウォームスタートプロセスで生成される軌道品質に大きく依存する。

一方、H-OBBCAとTDR-OBBCAは、ラグランジュ関数と制約の勾配とヘシアンを計算するCasadiソルバーとIpoptソルバーを使用するため、NLP問題を定式化するのにさらに時間を必要とすることは注目に値する。一方、OCEANはADMMのサブ問題をSOCPまたはQP問題として提起し、ECOSとOSQPで解くことができる。問題の定式化時間は、ほとんど無視できるほどよい。

表. IVに各サブ問題の詳細な解答時間を示す。表から、主な消費時間は、双対変数(15)の更新とMPC問題(20)の解法によるものであり、それぞれ解法時間の約63.63%と4.48%を占めていることがわかる。ADM問題分解と並列最適化のおかげで、我々の手法は平均160msで計画ループを完了するのに対し、TDR-OBBCAとH-OBBCAは1200ms以上かかる。

D. 実世界での道路テスト

我々は、その堅牢性と効率性を検証するために、オートパイロットプラットフォーム上で数百時間の運転テストを行った。軌道提供者の全体的な上流アーキテクチャは、[8]に記述されているものと同様である。しかし、知覚モジュールでは、純粋に視覚的なアプローチに従って、カメラのみを入力ソースとして使用する。

V-Cで紹介する3種類のシミュレーションシナリオは、数百時間の実際の道路テストから選択されたものである。例えば、Fig. 5 は典型的な垂直駐車テストシナリオである。すべての路面試験において、横方向の制御精度は0.01~0.2mと高い。

表V:Valet parking: 実際の道路テストにおける軌跡端の姿勢精度。

駐車場シナリオ(パーティション)	Lateral Error (m)	Heading Error (deg)
Vertical parking	0.0020	0.298
Parallel parking	0.0014	0.037
Oblique parking	0.0019	0.043

表. Vは、各軌跡終点における横方向誤差と方位誤差を含む、実世界の道路テストの制御性能を示している。自律走行車のコントローラは、OCEANによって生成された軌道をたどる。この結果から、OCEANによって生成された軌道は滑らかであり、

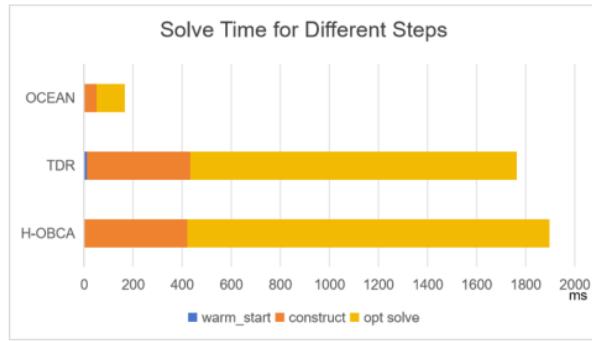


図4:3つのアルゴリズムにおける異なる解ステップでの時間消費の比較。

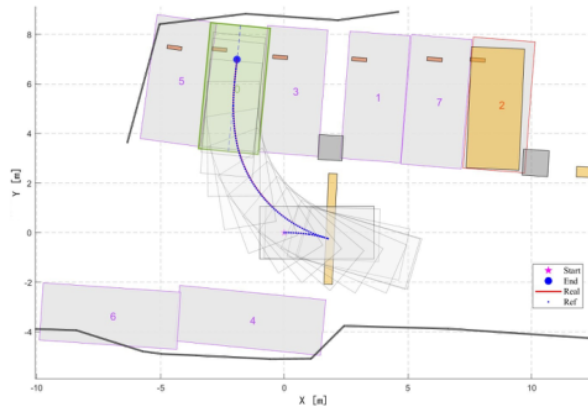


図5: 実テストによるトラッキング軌道の可視化。

車両の運動学的制約を受け、実世界の道路試験における追跡誤差が小さくなることが確認された。

VI. CONCLUSIONS

本論文では、駐車計画問題の解決において高い効率性と頑健性を示すOCEANと呼ばれる自律駐車軌道最適化手法を提案し、検証する。数百のシミュレーションシナリオと実世界の駐車場で検証を通じて、本手法がシステム性能の点で他のベンチマーク手法を凌駕することを確認した。本手法により、リアルタイム性能が要求される低コンピューティングパワープラットフォームに大規模パーキングプランナを導入することが可能となる。

REFERENCES

- [1] Liang, Zhao, Guoqiang Zheng, and Jishun Li. "Automatic parking path optimization based on bezier curve fitting." 2012 IEEE International Conference on Automation and Logistics. IEEE, 2012.
- [2] Alonso-Mora, Javier, et al. "Collision avoidance for aerial vehicles in multi-agent scenarios." *Autonomous Robots* 39 (2015): 101-121.
- [3] da Silva Arantes, Marcio, et al. "Collision-free encoding for chance-constrained nonconvex path planning." *IEEE Transactions on Robotics* 35.2 (2019): 433-448.
- [4] Wang, Qianhao, et al. "A linear and exact algorithm for whole-body collision evaluation via scale optimization." 2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023.

- [5] Oleynikova, Helen, et al. "Signed distance fields: A natural representation for both mapping and planning." RSS 2016 workshop: geometry and beyond-representations, physics, and scene understanding for robotics. University of Michigan, 2016.
- [6] Zhang, Xiaojing, et al. "Autonomous parking using optimization-based collision avoidance." 2018 IEEE Conference on Decision and Control (CDC). IEEE, 2018.
- [7] Zhang, Xiaojing, Alexander Liniger, and Francesco Borrelli. "Optimization-based collision avoidance." *IEEE Transactions on Control Systems Technology* 29.3 (2020): 972-983.
- [8] He, Runxin, et al. "TDR-OBICA: A reliable planner for autonomous driving in free-space environment." 2021 American Control Conference (ACC). IEEE, 2021.
- [9] Han, Ruihua, et al. "Rda: An accelerated collision free motion planner for autonomous navigation in cluttered environments." *IEEE Robotics and Automation Letters* 8.3 (2023): 1715-1722.
- [10] Erlien, Stephen M., Susumu Fujita, and J. Christian Gerdes. "Safe driving envelopes for shared control of ground vehicles." *IFAC Proceedings Volumes* 46.21 (2013): 831-836.
- [11] Chen, Jianyu, Wei Zhan, and Masayoshi Tomizuka. "Constrained iterative lqr for on-road autonomous driving motion planning." 2017 IEEE 20th International conference on intelligent transportation systems (ITSC). IEEE, 2017.
- [12] Liu, Sikang, et al. "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments." *IEEE Robotics and Automation Letters* 2.3 (2017): 1688-1695.
- [13] Jiang, Yuncheng, et al. "Robust Online Path Planning for Autonomous Vehicle Using Sequential Quadratic Programming." 2022 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2022.
- [14] Schulman, John, et al. "Motion planning with sequential convex optimization and convex collision checking." *The International Journal of Robotics Research* 33.9 (2014): 1251-1270.
- [15] Tan, Shuhang, Zhiling Wang, and Yan Zhong. "RCP-RF: A Comprehensive Road-car-pedestrian Risk Management Framework based on Driving Risk Potential Field." *arXiv preprint arXiv:2305.02493* (2023).
- [16] Zhang, Tingrui, et al. "Continuous Implicit SDF Based Any-shape Robot Trajectory Optimization." *arXiv preprint arXiv:2303.01330* (2023).
- [17] Li, Guoliang, et al. "Edge Accelerated Robot Navigation with Hierarchical Motion Planning." *arXiv preprint arXiv:2311.08983* (2023).
- [18] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1-122, 2011.
- [19] Huang, Zhenmin, Shaojie Shen, and Jun Ma. "Decentralized iLQR for Cooperative Trajectory Planning of Connected Autonomous Vehicles via Dual Consensus ADMM." *arXiv preprint arXiv:2301.04386* (2023).
- [20] Domahidi, Alexander, Eric Chu, and Stephen Boyd. "ECOS: An SOCP solver for embedded systems." 2013 European control conference (ECC). IEEE, 2013.
- [21] Lobo, M. S., Vandenberghe, L., Boyd, S., & Lebret, H. (1998). Applications of second-order cone programming. *Linear Algebra and Its Applications*, 193-228. [https://doi.org/10.1016/S0024-3795\(98\)10032-0](https://doi.org/10.1016/S0024-3795(98)10032-0)
- [22] Acikmese, B., & Ploen, S. R. (2007). Convex Programming Approach to Powered Descent Guidance for Mars Landing. *Journal of Guidance, Control, and Dynamics*, 1353-1366. <https://doi.org/10.2514/1.27553>
- [23] Zhou, Jinyun, et al. "DI-iaps and pjso: A path/speed decoupled trajectory optimization and its application in autonomous driving." *arXiv preprint arXiv:2009.11135* (2020).
- [24] Fraichard, Thierry, and Alexis Scheuer. "From Reeds and Shepp's to continuous-curvature paths." *IEEE Transactions on Robotics* 20.6 (2004): 1025-1035.