

ハミルトン-ヤコビ到達可能性を用いた安全保証付き自律走行車 駐車のための高速経路計画

Xuemin Chi¹, Jun Zeng², Jihao Huang¹, Zhitao Liu^{1*}, Hongye Su¹

概要—一般的な駐車場の逼迫シナリオを解決するために、ハミルトン-ヤコビベースの双方向A*(HJBA*)と呼ばれる高速な計画アーキテクチャを提案する。このアルゴリズムは、高レベルのHJベースの到達可能性解析と低レベルの双方向A*探索アルゴリズムからなる2層構造である。高レベル到達可能性解析では、車両ダイナミクスに関する後方到達可能管(BRT)がHJ解析によって計算され、安全な到達可能集合を得るために安全な集合と交差する。安全集合は、環境中の障害物に対する正の符号付き距離の制約によって定義され、オフラインでQP最適化問題を解くことによって計算される。交差点集合の内部、すなわち安全な到達可能集合に対して、計算された後方到達可能管は、システムダイナミクスと入力境界に従う到達可能であることを保証し、安全集合は、異なる形状の障害物に関して、駐車場の安全性を満たすことを保証する。オンライン計算では、安全な到達可能集合からランダムな状態をサンプリングし、双方向A*探索で考慮すべきヒューリスティックなガイドポイントとして使用する。双方向A*探索は、安全な到達可能集合から各ランダム化状態に対して並列化される。提案する2レベル計画アルゴリズムが、典型的な駐車要求に対して、異なる駐車シナリオを効果的かつ計算的に高速に解くことができることを示す。我々は、大規模なランダム化駐車シナリオにおけるシミュレーションを通じて我々のアルゴリズムを検証し、他の最先端駐車計画アルゴリズムを凌駕できることを実証する。

索引用語—ハミルトン-ヤコビ到達可能性、経路計画、自律走行、タイト環境における経路計画。

I. INTRODUCTION

A. Motivation

自律走行車場は、selfA走行車にとって重要なトピックである。最適化ベース[1]の計画アルゴリズムの計算量は、障害物が集まっているため劇的であるため、高密度環境における自律走行車は困難である。これを節約するために、既存の研究では、通常、HybridA* [2]のように、初期状態と目標状態の間の運動学的に実行可能な経路を見つけるために、探索ベースのプランニングが使用されるグローバルプランニングアルゴリズムを利用している。しかし、これらのアプローチは、探索ベースのアルゴリズム設計における発見的コスト関数に起因するデッドロック[3]やタイムアウトに悩まされる可能性がある。したがって、これらのアプローチはオンライン計画問題の解決には適用できない。

This work was supported in part by National Key R&D Program of China (Grant NO. 2021YFB3301000); National Natural Science Foundation of China (NSFC:62173297), Zhejiang Key R&D Program (Grant NO. 2022C01035), Fundamental Research Funds for the Central Universities (NO.226-2022-00086).

¹ State Key Laboratory of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, China {chixuemin, jihao, ztliu, hysu}@zju.edu.cn.

² Hybrid Robotics Group at the Department of Mechanical Engineering, UC Berkeley, USA {zengjuns@berkeley.edu}.

Simulation results are shown in <https://youtu.be/6NPAYCv7KMQ>.

* Corresponding author.

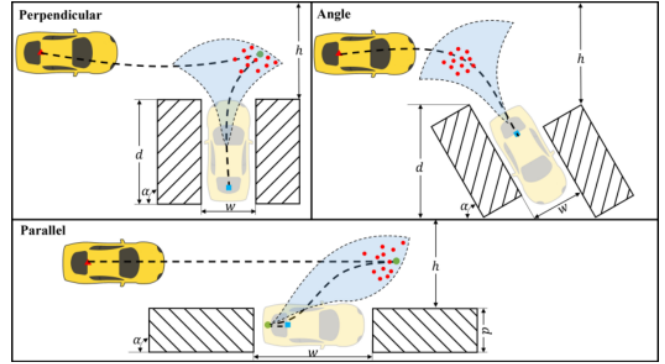


図1:一般的な駐車シナリオの3つの典型的なタイプ: 垂直駐車、角度駐車、平行駐車。初期ポーズは赤い三角形で、ゴールポーズは青い四角形で示されている。緑色の実線ドットは、方向が変わるカスプである。水色の領域はBRTの部分集合であり、探索の指針として使用される。赤丸はBRTのランダムな連結状態である。hは駐車場における自由走行スペースの高さ、dは駐車場の内側と外側の自由走行エリアまでの距離、wは駐車場の幅である。

実際には、駐車環境自体は通常、周囲の障害物や有限個の潜在的な駐車場所を含めて固定されており、初期状態の異なる同じ駐車環境下で、様々な駐車要求を迅速に処理することが期待される。各目標状態は、車両の位置や向きを含む各駐車場に関連付けられる。本論文では、この固定駐車場環境設定を動機として、駐車場経路計画問題をオフライン到達可能性計算とオンライン探索計算に分離した新しい計画アーキテクチャを提案する。我々のオフライン到達可能性解析は、有限の潜在的な目標状態を考慮し、各目標状態の安全な到達可能集合を計算する。安全到達可能集合は、車両の状態に関するハミルトンヤコビ(HJ)到達可能性解析から計算される後方到達可能管(BRT)の交点であり、安全集合は、車両の矩形形状に従った符号付き距離制約によって定義される。次に、目標状態が与えられると、この安全な到達可能集合から複数の連結状態がサンプリングされ、双方向ヒューリスティック探索を用いて、初期状態から連結状態、連結状態から目標状態までの間の運動学的に実現可能な2セグメント経路をそれぞれ見つける。これらの生成された2セグメントパスから、ユーザーの好みに応じて最適なパスを選択することができる。これらのサンプリング連結状態により、既存の研究と比較して過剰な検索を避けることができる。オフライン/オンライン計画アーキテクチャにより、図1に示すような様々な駐車シナリオをタイムアウトすることなく、最先端技術よりも低い計算量で解くことができる。

B. Related Work

モーションプランニングはロボット工学のコミュニティで非常に活発な研究分野であり、自律駐車計画は非ホロノミックロボットのための特化サブドメインである[4]。一般に、計画アルゴリズムは、グラフベース[5]~[7]、サンプルベース[8]~[10]、曲線補間[11]、軌道最適化[1]、[12]~[14]、到達可能性解析[15]~[19]の5つのカテゴリに分類される。

サンプルベースのアプローチ(例えば、RRT* [10])は、高次元経路計画において優れたスケール性能を示す。とはいえ、古典的なサンプリングに基づくアプローチでは、前方マヌーバと後方マヌーバの組み合わせを計画することはできない。そこで、[20]、[21]では、双方向RRTに基づくアプローチをReed-Sheep(RS)曲線[22]で拡張し、自律駐車に適用する。しかし、上記のアプローチは、コストを最小化することなく予測不可能な経路の実現可能性に焦点を当てており、以下の軌道最適化においてしばしば高いコストにつながる。構造化された道路を走行する場合、最適な候補曲線を選ぶことで滑らかで計算量が少ないため、曲線(例えば、ベジェ曲線[11])補間アプローチが好まれる。

TrajOpt[23]、CHOMP[24]、LQR[25]、MPC[26]などの最適化ベースの手法(別名。軌道最適化)は、運動計画問題を、目的関数と様々な制約条件を含む最適制御問題として定式化する。この問題は、勾配に基づく反復法によって直接解くこともできるし、一般的なソルバー(Gurobi, Cplex, IPOPTなど)によって解かれる数値最適化問題として再定式化することもできます[27]。最適化ベースのアプローチは、非常に滑らかで良い結果をもたらすことができるという利点を持ち、時にはユーザー定義の非線形制約を扱うことさえできるが、結果の質は強く関連し、初期推測に敏感である[28]。最適解からかけ離れている場合、結果は劣り、収束が遅くなる。一方、勾配情報が効率的で有用なガイダンスを提供しない場合、収束は不可能かもしれない。13]では、改良されたA-searchガイドツリー(i-AGT [29])が参照パスを提供するために使用されているが、更新頻度が低いため、動的な障害物との連携が制限される。30]では、初期推測を行うのではなく、車が駐車場の近くにあるときに最適化戦略を実行し、最適化問題を解けるようにしている。

上記の多くの計画アルゴリズムがロボットナビゲーションのために開発され、ほとんどの環境に対して実行可能な経路を生成することができるが、これらのアルゴリズムは、低い計算コストで一般的な駐車問題を確実に処理し、異なるケースにおける複雑な駐車操作による解を提供する実現可能性を維持するように設計されていない。駐車場計画領域では、グラフベースと幾何学ベースのアプローチがもう一つの手法の枝である。グラフベースの探索手法は、ロボットナビゲーション(例えば、A* [5]とその亜種 [6])で広く使用されており、事前に定義された運動プリミティブは、探索時に運動学的制約を考慮することを可能にする。初期の純粋な幾何学的研究は、駐車経路を計画するためにDubins曲線[31]とRS曲線に依存しているが、それらは障害物のない環境に限定されている。

一方、直線と円弧のセットの代わりに、いくつかの研究[32]~[34]は、連続曲率駐車経路の計画に焦点を当てており、異なる曲線が使用されている。それにもかかわらず、これらのアプローチは、最適でない駐車長、固定または制限されたスタート構成に悩まされており、また、障害物のないシナリオに限定されている。ARA*[6]やTheta*[7]のようなA*ベースのバリエーションのほとんどは、自律走行車の駐車問題を扱うことができない。

最先端の一般的なアプローチは、ハイブリッドA*(HA*)[2]であり、車両運動学的制約を受ける3D(x, y, θ)ノード運動プリミティブを事前に定義し、RS曲線を組み込んだものである。最初の部分は3D A*の探索結果であり、滑らかな曲率はあらかじめ定義されたモーションプリミティブから得られる。第2部はRS曲線の解析的展開の結果である。定義済みのモーションプリミティブは旋回角限界で囲まれ、密度は1つのハイパーパラメータで制御される。しかし、複雑なヒューリスティック関数は、後方移動、方向転換、旋回移動などの異なるペナルティ項間の慎重な手動トレードオフを必要とする。駐車場シナリオが異なると、高設計のウェイトチューニングが必要となり、一般的な駐車場シナリオでは不安定で性能が低下する。最近の研究[1]、[35]では、HA*を用いて駐車問題における軌道最適化を高速化し、実現可能性を保証している。しかし、HA*の更新頻度が低いため(例えば、数秒から数分に悪化するケース)、再計画は非現実的であり、静的な障害物しか考慮できない。HA*におけるコスト関数(前方、後方、旋回探索コストなど)からの複雑なペナルティパラメータのトレードオフも、異なる駐車シナリオにおいて不安定な性能をもたらす[36]。静的駐車環境におけるHA*の様々なバージョン[37]~[39]が開発されているが、計算負荷や不安定な性能に悩まされている。

リーチャビリティ解析技術により、大域的に最適な後方リーチャブルチューブの計算が可能となり、探索ベースのアルゴリズムの発見的ガイダンスとして機能することが確認された。これにより、オフラインで計算されたBRTを繰り返し使用することで、検索速度を向上させ、安全性を確保することができ、上記の課題に対する解決策を提供することができる。ハミルトン-ヤコビ(HJ)到達可能性解析は、システムの安全性と性能を検証する手法であり、最近、運動計画や衝突回避のアプリケーションで普及している[17]~[19]。HJ到達可能性解析では、システムは常微分方程式(ODE)で表され、ターゲット集合は安全でない状態または望ましい状態の集合を記述する[16]。BRS/BRTは、HJ偏微分方程式(PDE)を解くことにより、レベルセット法[40]の正確な数値計算法により、HJ PDEの高速解法スキームを得ることができる。研究[41]では、プランニングとトラッキングのためのフレームワークがオンザフライで提案され、HJ PDEがプランニングとトラッキングの間の相対システムのTEBを取得するために計算される。HJベースのアプローチは駐車場計画に利用できる可能性があり、我々の知る限り、ヒューリスティック探索を組み合わせて駐車場計画の課題を解決するためにHJの到達可能性を拡張したのは初めてである。

しかし、HJの到達可能性は計算効率が低く、物体の形状をうまく扱えないという問題がある一方、制御バリア関数[42]は上記の欠点を克服できるが、局所的なアプローチである。

C. Contributions

自律駐車のための経路計画に関する先行研究では、リアルタイム経路計画アルゴリズムを提供することができなかった。この課題を解決するために、我々はハミルトン-ヤコビ到達可能性解析とグラフベースの探索アルゴリズムに基づく新しい2層計画戦略を提案する。我々の研究はいくつかの重要な貢献をしている：

- 我々は、一般的な駐車シナリオにおける高速なプランニングを提供するために、グラフベースの探索アプローチでHJ到達可能性分析を活用する駐車経路計画アルゴリズムを提示する。
- 我々のHJBA*駐車経路計画アプローチは、パラメータチューニングを必要とせず、例えば複数回の駐車要求など、様々な駐車シナリオにおいて高速かつ安定した計画を提供するように設計されている。
- 提案アルゴリズムの性能をバッチシミュレーションにより検証し、異なる駐車シナリオにおける最新の計画アルゴリズムと比較するベンチマークを行う。その結果、提案アルゴリズムは、効率と安定した性能の点で他のアルゴリズムを凌駕することが示された。

D. Paper Structure

本論文の構成は以下の通りである。まず、セクションIIで我々のアルゴリズムHJBA*のフレームワークを示す。次に、セクションIIIで安全な到達可能集合を計算するプロセスについて詳しく説明する。セクションIVでは、接続された状態に対する双方向探索のプロセスについて説明する。セクションVでは、様々な駐車シナリオにおける大規模バッチシミュレーションの結果を示し、他の最先端計画アルゴリズムと比較するベンチマークを行う。最後に、セクションVIで今後の研究の方向性と未解決の問題点について述べ、本稿を締めくくる。

TABLE I: Variable Table.

Notations	Descriptions
\mathbf{z}	vehicle states vector contains x, y, θ
$\Omega_{\mathbf{z}}$	\mathbf{z} の各次元に沿った離散化された領域空間
$\mathbf{z}_0, \mathbf{z}_g$	車両システムの初期状態ベクトルと目標状態ベクトル
G_0	parking goal set
C	後方到達可能チューブ
S	safe signed distance set
$R(\cdot), T(\cdot)$	回転行列と平行移動ベクトル
A_i, b_i	$A_i \mathbf{y}_i \leq b_i$ 多面的な領域を記述する
V_i	A_i と b_i で記述される車が占める幾何学的空間。
\odot_i	A_i と b_i で記述されるi番目の障害物
$g(\mathbf{z}_0, \mathbf{z}_g)$	computed cost-to-go from \mathbf{z}_0 to \mathbf{z}_g
$h(\mathbf{z}_0, \mathbf{z}_g)$	\mathbf{z}_0 から \mathbf{z}_g までの推定コスト
$c(\cdot)$	$g(\cdot, \cdot)$ と $h(\cdot, \cdot)$ からなるノードのコスト
$\mathcal{A}(\mathbf{z}_0, \mathbf{z}_g)$	path connected \mathbf{z}_0 and \mathbf{z}_g generated by node expansions
$\mathcal{RS}(\mathbf{z}_0, \mathbf{z}_g)$	path connected \mathbf{z}_0 and \mathbf{z}_g generated by RS curves
$P(\mathbf{z}_0, \mathbf{z}_g)$	path starts at \mathbf{z}_0 and ends \mathbf{z}_g
$Q_{\text{open}}, Q_{\text{close}}$	Open and close list maintained in forward or backward searching
\mathbf{z}_k^S	Sampling connected states from S
$\mathbf{z}_{\text{best}}^S$	States connected to the goal with collision-free \mathcal{RS} curves
$\mathbf{z}^F, \mathbf{z}^B$	States expanded in forward search and backward search

II. FRAMEWORK OF HJBA*

本節では、HJBA*のワークフローを示し、その詳細を図2にまとめる。

本アルゴリズムは、駐車場における自動車の最適な駐車経路を効率的に探索することを目的としている。目的の駐車場がすでに埋まっている場合、双方向探索ベースのアルゴリズムでは、代替駐車場を迅速に見つけるために、高速な再計画戦略が採用される。さらに、駐車操作の間、ドライバーは通常、駐車場所に徐々に近づき、駐車を成功させる前に安全を優先しながら、常に位置を調整する。この発見的で安全性を考慮した戦略を我々のアルゴリズムに組み込むために、安全な到達可能集合の概念を導入し、オフライン層で実装する。

本アルゴリズムのオフラインコンポーネントは、オンラインフレームワークで利用可能な安全な到達可能集合(S と表記)を生成することを目的としている。この集合は、安全性と実現可能性の両方の要件を満たす状態から構成される。具体的には、 S は G で示される後方到達可能管と C で示される安全集合の交点を取ることによって得られる。本アルゴリズムで到達可能性解析を行うために、その単純さと計算効率から、グローバルプランナーレベルでの車両のダイナミクスと摩擦を考慮する必要のないDubins車モデルを採用する。しかし、このアプローチは、車の幾何学的形状を考慮しない点表現を利用する。その結果、 G の状態はシステムダイナミクスに従うが、厳しい環境では安全性を保証できない可能性がある。この制限を克服するために、我々は二次計画法(QP)最適化問題を解くことで、車の凸矩形形状と環境との間の衝突回避に対処する。この最適化問題は、このようなシナリオにおける安全性を確保するために、正符号距離の制約を組み込んでいる。

実際には、我々のアルゴリズムでは \mathbf{z}_g で示される目標状態は、駐車環境で利用可能な有限の駐車スポットの1つを指す。駐車場環境に関する情報は、 G と C の両方を生成するために利用される。これらの集合の交点が S を形成し、それが我々のアルゴリズムのオンラインコンポーネントによって利用される。オンラインフレームワークのためのヒューリスティックな連結状態を作成するために、 S からランダムに状態をサンプリングする。

オンラインフェーズにおいて、我々のアルゴリズムは、駐車環境内の任意の初期状態 \mathbf{z}_0 を、ユーザの駐車要求に基づいて指定することを可能にする。 \mathbf{z}_0 と目標状態 \mathbf{z}_g の情報を利用し、並列双方向A*探索を行い、 \mathbf{z}_i^S で示される連結状態を得る。双方向A*探索は、前方探索と後方探索の両方を含む。前方探索では、 \mathbf{z}_0 と \mathbf{z}_i^S をつなぐ経路を計算し、 $P(\mathbf{z}_0 \rightarrow \mathbf{z}_i^S)$ とする。後方探索では、 \mathbf{z}_g と \mathbf{z}_i^S をつなぐ経路を計算し、 $P(\mathbf{z}_i^S \leftarrow \mathbf{z}_g)$ とする。これらの経路の各組を組み合わせることで、 $P_{\mathbf{z}} S_i(\mathbf{z}_0 \rightarrow \mathbf{z}_g)$ で示される連結駐車経路が得られる。ユーザが指定した基準に基づいて、これらの接続パスの1つが選択され、最適な駐車パスとして出力される。

III. SAFE REACHABLE SET COMPUTATION

This section introduces offline computation of HJBA*.

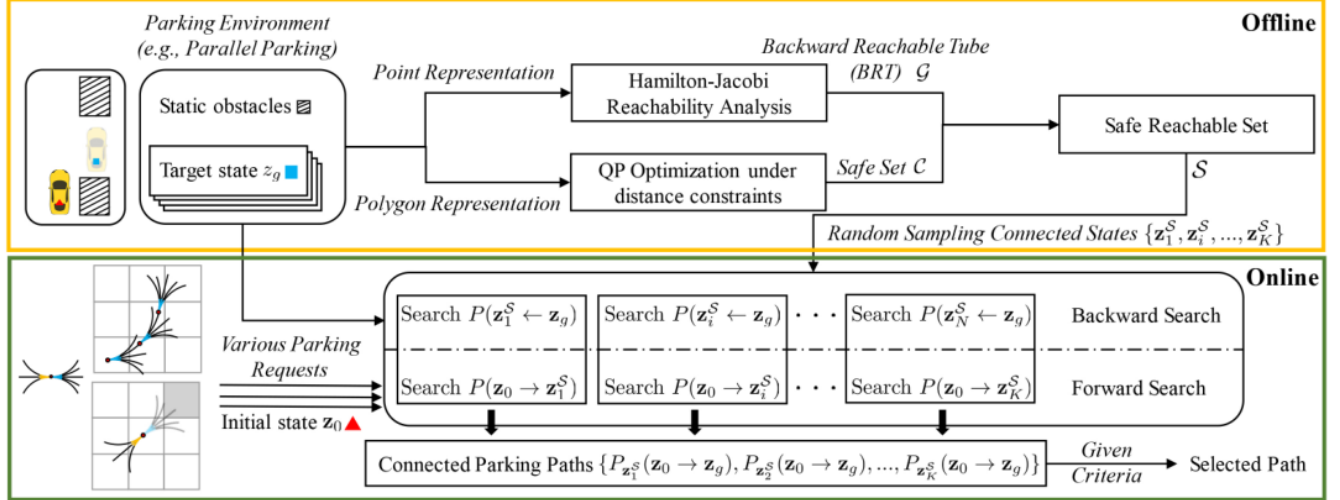


図2: 我々のアプローチは2層スキームを含む。駐車場環境には、駐車場と情報に関わる障害物が含まれる。オフライン計算では、安全な到達可能集合を計算し、連結状態のサンプリングを行う。オンライン検索層では、様々な駐車要求に対応するため、並列双方向検索を行う。

A. 後方到達可能管

ODEに従って次のように進化する連続時間制御系を考える：

$$\frac{dz(\tau)}{d\tau} = \dot{z}(\tau) = f(z(\tau), u(\tau)), \tau \in [t, T] \quad (1)$$

$$z(\tau) \in \mathcal{Z}, u(\tau) \in \mathcal{U},$$

ここで、 $z(\tau) \in \mathcal{Z} \subseteq \mathbb{R}^n$ はシステムモデルの状態を表す。制御入力を $u(\tau) \in \mathcal{U} \subseteq \mathbb{R}^m$ とする。制御関数 $u(-)$ は、測定可能な関数の集合から引かれる： $(t, T) \rightarrow \mathcal{U}$ 。ここで \mathcal{U} はコンパクトで安全制御制約を満たす。システムダイナミクス $f: \mathbb{R}^n \times \mathcal{U} \rightarrow \mathbb{R}^n$ は、 u を固定して z において一様にリプシッツ連続であると仮定する。したがって、 $u(-) \in \mathcal{U}$ が与えられたとき、時刻 t の状態 z から始まる一意な軌跡が常に存在し、 $l_{z,t}^u(\tau)$ と表記される： $[t, T] \rightarrow \mathbb{R}^n$ である。BRT は、システムが時間地平 $[T - t, T]$ 内でゴール集合 \mathcal{G}_0 に追い込まれる状態 $z \in \mathbb{R}^n$ の集合を表す。これは正式には次のように記述される：

$$\mathcal{G}(t) = \{z : \forall u(\cdot) \in \mathcal{U}, \exists \tau \in [t, T], l_{z,t}^u(\tau) \in \mathcal{G}_0\}, \quad (2)$$

目標集合 \mathcal{G}_0 を部分レベル集合関数 $l(z)$ で表現することが多い： $\mathcal{Z} \rightarrow \mathbb{R}$ を $\{ \mathcal{G}_0 = \{z \in \mathcal{Z} : l(z) \leq 0\} \}$ とする。

備考 1. HJ 到達可能性の文脈では、後方到達可能集合はターゲット集合か安全でない集合のどちらかを表すことができる。本論文では、ターゲット集合を表現し、図2に示すように、与えられた駐車場における有限の駐車空間をモデル化するために利用することを考える。

ハミルトン-ヤコビ-ヤコビ-アス変分不等式と本アルゴリズムで使用するレベルセット解法の定式化方法の詳細については、[18] を参照してください。使用したシステムモデルは

$$\dot{z} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \cos \theta \\ v \sin \theta \\ \omega \end{bmatrix}, \quad (3)$$

ここで、 (x, y, θ) は直交座標における位置と方位角を表し、 (v, ω) は等速と制御入力角速度を表す。

B. Safe Set

\mathcal{G} は $\mathbb{R}^n \times [T, t]$ の固定直交格子上で計算されたと考える。グリッドは $\Omega_z = \Omega_x \times \Omega_y \times \Omega_\theta$ で表す。簡単のために、各有界区間を n 個の要素数で離散化すると仮定する。状態 $z_k \in \Omega_z$ に対して、以下の幾何学空間を占有とする：

$$\mathbb{V}_k := R(z_k)\mathbb{V}_0 + T(z_k), \quad (4)$$

ここで、 $R(-) \in \text{SE}(2)$ は回転行列、 $T(-): \mathbb{R}^{np} \rightarrow \mathbb{R}^{np}$ は並進ベクトル、 n_p は位置次元である。集合 \mathbb{V}_0 は $\mathbb{V}_0 := \{y \in \mathbb{R}^n : A_0 y \leq b_0\}$ と記述され、ここで A_0 と b_0 は車の長さ L と幅 W に依存する。与えられた駐車環境において、 N 個の静的障害物が存在すると仮定する：

$$\mathbb{O}_i := \{y \in \mathbb{R}^n : A_i^\top y \leq b_i^\top\}, \quad (5)$$

ここで、 $A_i^\top \in \mathbb{R}^{s \times n}$ 、 $b_i^\top \in \mathbb{R}^s$ 、 s は障害物の顔を表す。 \mathbb{O}_i 、 $i \in \{1, \dots, N_0\}$ は凸コンパクト集合で空でないものとする。通常、グリッド Ω_z 内の障害物を考慮するだけでよく、 \mathbb{O}_i で示される。

\mathbb{V}_k と \mathbb{O}_i のポリトープ表現を構築した後、各ペア $(\mathbb{V}_k, \mathbb{O}_i)$ 間の最小符号付き距離 $d_i^{z_k}$ を求める QP 問題を定式化できる。計算の詳細は[3]を参照されたい。安全集合は z_k を含み、正の符号付き距離は次式で示される。

$$\mathcal{C}_i := \{z_k \in \Omega_z : d_i^{z_k} > 0\}, \quad (6)$$

ここで、 \mathcal{C}_i は \mathbb{O}_i を考慮した安全集合であり、完全安全集合は $\mathcal{C} := \bigcap_{i=1}^{N_0} \mathcal{C}_i$ である。

C. 安全な到達可能集合

BRT G は、システムダイナミクスを考慮しながら、チューブ内の状態が駐車場に到達することを保証する。一方、安全集合 C は、状態が衝突しないことを保証する。状態 z_k の値は G 内で負であり、境界 ∂G 上ではゼロである。 S で示される安全な到達可能集合を得るために、後方到達可能管と安全な集合の交差操作、すなわち、 $S := G \cap C$ を実行する。これにより、結果として得られる集合は、安全で効率的な駐車に不可欠な、運動学的に実行可能かつ衝突のない状態のみを含むことが保証される。

D. 接続状態のサンプリング

与えられた駐車シナリオにおいて、安全な到達可能集合 S を計算する。接続状態は以下のように一様サンプリングによって取得される。

$$\mathbf{z}_k^S = \{(x_k, y_k, \theta_k) | x_k \geq x_g, y_k \geq y_g, (x_k, y_k, \theta_k) \in S\}, \quad (7)$$

ここで、 x_g, y_g は駐車ゴールポーズの座標である。接続状態の数は20である。

目標とする駐車ポーズは、どの状態がつながっているかを決定することで、駐車シナリオの性質を決定することが多い。BRTと安全集合の交点を表す安全な到達可能集合を定式化した。このアプローチは、時間が最小のときにBRTで発生する可能性のある望ましくない状態を排除するのに役立ち、その結果、接続された状態が駐車ゴールに近すぎる可能性がある。この概念を理解する直感的な方法は、パーキングスポットに人が立っていることが私たちに指示することの望ましくないことを視覚化することである。むしろ、駐車場の外や周辺にいる人々からのガイダンスを得る方が実現しやすい。サンプリング状態の量はユーザー依存である。

安全集合 C は衝突しやすい状態を持たないが、駐車場と安全集合内の状態との間の障害物による潜在的な障害物のために、その実行可能性を考慮に入れていない。特定の駐車シナリオにおいて、安全ではあるが、運動学的な実行可能性がないため、探索を指示することが困難または非効率になる可能性のある接続状態をサンプリングすることは最適ではない。これは安全な到達可能集合 S によって捉えられる重要な側面であり、安全な集合では考慮されない。

IV. オンライン双方向探索

本節では、図2に示すように、連結点の助けを借りて高速探索を実現するA*ベースの変形アプローチを探索する。

A. 検索戦略

与えられた駐車環境における探索速度を向上させるために、前方探索と後方探索の両方を採用し、状態を接続する。プランニングアルゴリズムの双方向の変形は、狭い環境や高次元の構成空間のような困難な領域を持つ問題で頻繁に採用される[43]。

Algorithm 1: BA*:Search Strategy

Input: $\emptyset, \mathbf{z}_0, \mathbf{z}_g, \mathbf{z}_i^S$

```

1  $Q_{\text{open}}^F \leftarrow \emptyset, Q_{\text{close}}^F \leftarrow \emptyset, \text{flag}^F \leftarrow \text{False};$ 
2  $Q_{\text{open}}^B \leftarrow \emptyset, Q_{\text{close}}^B \leftarrow \emptyset, \text{flag}^B \leftarrow \text{False};$ 
3  $Q_{\text{open}}^F \leftarrow \text{Initialize}(\mathbf{z}_0), Q_{\text{open}}^B \leftarrow \text{Initialize}(\mathbf{z}_g);$ 
4 while  $Q_{\text{open}}^F \neq \emptyset$  and  $Q_{\text{open}}^B \neq \emptyset$  do
5   if  $\text{flag}^F$  and  $\text{flag}^B$  then
6     break;
7   if not  $\text{flag}^F$  then
8      $\mathbf{z}_{\text{best}}^F = Q_{\text{open}}^F.\text{pop}: c(\mathbf{z}_{\text{best}}^F) < c(\mathbf{z}^F), \forall \mathbf{z}^F \in \mathcal{F}$ 
9      $(\text{flag}^F, \mathcal{RS}(\mathbf{z}^F, \mathbf{z}_i^S)) \leftarrow \text{RSExpansion}(\mathbf{z}^F, \mathbf{z}_i^S)$ 
10     $\text{ForwardExpand}(\mathbf{z}_{\text{best}}^F, \mathcal{F}, Q_{\text{open}}^F, Q_{\text{close}}^F)$ 
11  if not  $\text{flag}^B$  then
12     $\mathbf{z}_{\text{best}}^B = Q_{\text{open}}^B.\text{pop}: c(\mathbf{z}_{\text{best}}^B) < c(\mathbf{z}^B), \forall \mathbf{z}^B \in \mathcal{B}$ 
13     $(\text{flag}^B, \mathcal{RS}(\mathbf{z}^B, \mathbf{z}_i^S)) \leftarrow \text{RSExpansion}(\mathbf{z}^B, \mathbf{z}_i^S)$ 
14     $\text{BackwardExpand}(\mathbf{z}_{\text{best}}^B, \mathcal{B}, Q_{\text{open}}^B, Q_{\text{close}}^B)$ 
15  $P(\mathbf{z}_0 \rightarrow \mathbf{z}_i^S) = \text{GetPath}(Q_{\text{close}}^F, \mathcal{RS}(\mathbf{z}^F, \mathbf{z}_i^S, \mathbf{z}_0));$ 
16  $P(\mathbf{z}_i^S \leftarrow \mathbf{z}_g) = \text{GetPath}(Q_{\text{close}}^B, \mathcal{RS}(\mathbf{z}^B, \mathbf{z}_i^S, \mathbf{z}_g));$ 
17  $P(\mathbf{z}_0, \mathbf{z}_g) = \text{CombinePath}(P(\mathbf{z}_0 \rightarrow \mathbf{z}_i^S), P(\mathbf{z}_i^S \leftarrow \mathbf{z}_g));$ 
Return:  $P(\mathbf{z}_0, \mathbf{z}_g)$ 

```

駐車場が駐車車両や障害物に囲まれていることが多い駐車場計画問題の場合、双方向探索戦略が非常に有効である可能性がある。我々のアプローチの探索ロジックをアルゴリズム1に示す。 $P(\mathbf{z}_0, \mathbf{z}_g)$ で示される駐車経路は、 $A(-, -)$ で表されるノード展開によって探索される経路と、 $\mathcal{RS}(-, -)$ で表されるReeds-Shepp曲線によって探索される経路の2つの部分から構成される。集合の大きさは $|\cdot|$ で示される。駐車場計画問題でよく使われる定式化は次式で与えられる:

$$P(\mathbf{z}_0, \mathbf{z}_g) = \mathcal{A}(\mathbf{z}_0, \mathbf{z}_{\text{best}}) \oplus \mathcal{RS}(\mathbf{z}_{\text{best}}, \mathbf{z}_g), \quad (8)$$

ここで、 \oplus はCombinePath演算を表す。

(8)を用いて駐車計画問題の一般的な解を記述することができるが、HA*アルゴリズムは非効率的なノード拡張に依存するため計算コストが高く、 $|\mathcal{A}|$ の大きなセットサイズにつながる可能性がある。ヒューリスティック双方向探索はパスを構築する

$$P_{(\mathbf{z}_0, \mathbf{z}_g)}^{\text{BA}^*} = P(\mathbf{z}_0 \rightarrow \mathbf{z}_i^S) \oplus P(\mathbf{z}_i^S \leftarrow \mathbf{z}_g). \quad (9)$$

この戦略では、接続状態 \mathbf{z}_i^S へのRSExpansionが比較的容易であるため、 $|\mathcal{RS}|$ が大きくなる可能性が高くなる。HA*[2]や[44]で用いられている単一方向探索戦略では、 $|\mathcal{A}|$ を効率的に減少させることはできない。

備考2. HJBA*の目的は、接続状態 \mathbf{z}_i^S の助けを借りて、 $|\mathcal{A}|$ のサイズを小さくし、 $|\mathcal{RS}|$ のサイズを大きくすることである。例えば、 $\circ = \triangle$ の場合、常に $|\mathcal{A}| = 0$ となり、パス P は \mathcal{RS} 曲線と等価である。この場合、計画時間は解析的な計算時間と等しく、非常に速く無視できる。この機能は、我々のフレームワークで十分に活用されている。

B. Node Expansion

ForwardExpandとBackwardExpandで異なるノード展開を使用する。双方向性のためのモーションプリミティブ

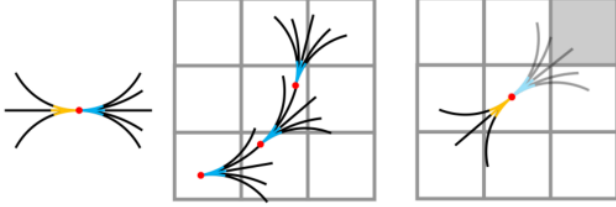


図3: 探索アルゴリズムのモーションプリミティブと展開モード。

検索は次のように定義される:

$$\begin{aligned} \mathcal{F} &= \{f(\mathbf{z}_{\text{best}}, u_k)d_k : u_k \in \mathbf{U}_F\}, \\ \mathcal{B} &= \{f(\mathbf{z}_{\text{best}}, u_k)d_k : u_k \in \mathbf{U}_B\}, \end{aligned} \quad (10)$$

ここで、 $f(-)$ はシステムダイナミクスである。便宜上、 $\mathbf{F} = \mathbf{F}^F \Delta \mathbf{F}^B$ とする。ここで、 \mathbf{F}^F は行動 $u_k \in \mathbf{U}_F$ による順ノード展開を含み、 $d_k = 1$ であり、 $d_k = -1$ の \mathbf{F}^B と類推する。

前方探索のノード展開は常に連結状態 \mathbf{z}_i^F に向かう。全ての前方展開 \mathbf{F}^F が失敗した場合、図3に示すように、後方展開の最大旋回角 \mathbf{F}^B を用いて探索を続ける。ForwardExpandのノード展開の詳細はアルゴリズム2に示す。BackwardExpandのノード展開は、駐車場が狭いことが多く、動きの解像度 $r^B < r^F$ が小さい方が好ましいため、両方向に伸びている。

備考3. 提案するノード拡張戦略は、障害物に近い場合にのみ後方拡張に焦点を当てるため、従来のA*ベースの探索アルゴリズムに対してより積極的な拡張につながり、探索時間の増加をもたらす可能性がある。しかし、ヒューリスティックな点として連結状態を用いることで、 \mathbf{z}_{best} を求めるための後方ノード展開への依存度が低くなる。

アルゴリズム2: BA*:NodeExpand

Input: $\mathcal{F}, \mathbf{z}_{\text{best}}, Q_{\text{open}}^F, Q_{\text{close}}^F$

```

1 while  $k \leq |\mathcal{M}|$  do
2    $\mathbf{z}_k \leftarrow \text{GetNode}(\mathcal{F}^F, \mathbf{z}_{\text{best}})$ 
3   CollisionCheck( $\mathbf{z}_k$ )
4   if  $k = |\mathcal{F}^F|$  then
5     break
6   if  $\mathbf{z}_k \in Q_{\text{close}}^F$  then
7     continue
8   if  $\mathbf{z}_k \notin Q_{\text{open}}^F$  then
9      $Q_{\text{open}}^F \leftarrow Q_{\text{open}}^F \cup \mathbf{z}_k$ 
10  else if  $c(Q_{\text{open}}^F(\mathbf{z}_k)) > c(\mathbf{z}_k)$  then
11     $Q_{\text{open}}^F(\mathbf{z}_k) \leftarrow \mathbf{z}_k$ 

```

Return: $Q_{\text{open}}^F, Q_{\text{close}}^F$

C. Cost Function

A*ベースの探索アルゴリズムの重要な構成要素はコスト関数であり、コスト-結果関数 $g(z_0, z)$ とヒューリスティック関数 $h(z, z_g)$ の2つの項からなる。 g と h に割り当てられた値の和は、探索プロセス中のノード選択に大きく影響し、これは探索アルゴリズムの重要な側面である。

広範なパラメータチューニングの必要性を回避するため、方向転換、後方移動、旋回移動に対するペナルティは組み込んでいない。これらのパラメータは、背中や寝返りの動きを減らすなど、様々な探索効果に対する選好を作成するために使用することができるが、チューニングに時間がかかり、すべてのシナリオではなく、特定の駐車シナリオにのみ適している。前方探索または後方探索の段階では、 $\mathbf{F} = \phi$ または $\mathbf{B} = \phi$ でない限り、ノードはサンプリング状態に向かってのみ拡張する。 g_{value} を運動プリミティブの長さとして計算する。

HA*はハイブリッドヒューリスティック関数を使用し、 $h(z, z_g) = \max(h_{rs}(z, z_g), h_{A*}(z, z_g))$ と表記され、障害物のある乱雑な環境では有効である。しかし、「コスト・トゥ・コメ」と「コスト・トゥ・ゴー」の間でペナルティを均衡させることは困難であり、望ましくない結果をもたらす可能性がある。この問題に対処するため、HJBA*では、ノード拡張戦略を補完するヒューリスティック関数として実対角距離を採用する。

D. 計画された駐車経路の継続性

計画駐車経路は、RS曲線と拡張区間Aを組み合わせで構築される。本節では、計画駐車経路が連続的であるという主張を支持する証拠を提供する。

図4aは、直感的な方法で経路の継続性を視覚的に示している。パスの連結セグメントはRS曲線によって形成され、パスの異なる部分間のスムーズな遷移を保証する。連結領域をより明確に見るために、図4bは連続パスの範囲を示している。

駐車場の経路は、方向が変わる地点で1つのカスプを示すことに注意することが重要である。このカスプは、パスの方向変化の位置を表している。継続性をさらに解析するために、図4cは、 θ に関連するx、y、曲率情報に沿った経路の計算された導関数を表示する。この結果は、パスの継続性を確認するものである。

全体として、視覚的デモンストレーションと数値微分分析は、我々のアプローチにおける計画された駐車経路の継続性を支持する実質的な証拠を提供する。

E. モーションプリミティブと探索戦略のデモ

我々のHJBA*アルゴリズムの特筆すべき利点の一つは、パラメータ調整を必要とせず、様々な駐車シナリオに対応できる汎用性である。対照的に、従来のHA*ベースのアルゴリズムは、最適な結果を得るためにパラメータチューニングを必要とすることが多い。

設計した探索とノード拡張戦略の利点と、パラメータチューニングを行わない連結状態 \mathbf{z}_i^F の利用を示すために、図5aに並列駐車の場合を示す。我々の探索木は一貫してノードを連結状態に向かって拡張し、後方ノード拡張の必要性を排除する。その結果、後方移動を抑制するためのペナルティパラメータが不要になる。このロバスト性は、スタートポーズが反対側であっても維持され、車両は接続された状態に向かって拡大する。したがって、方向転換のためのペナルティパラメータを微調整する必要はない。

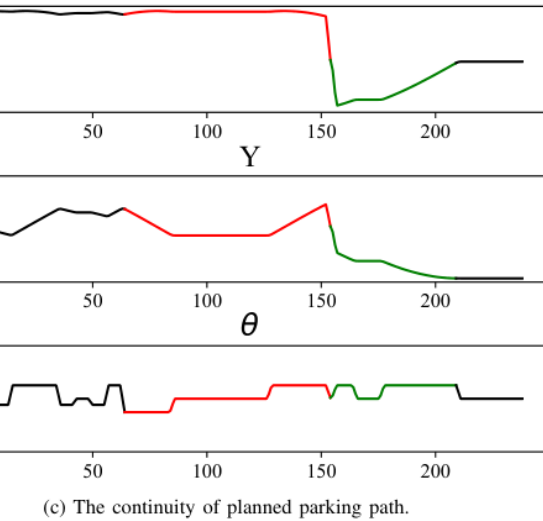
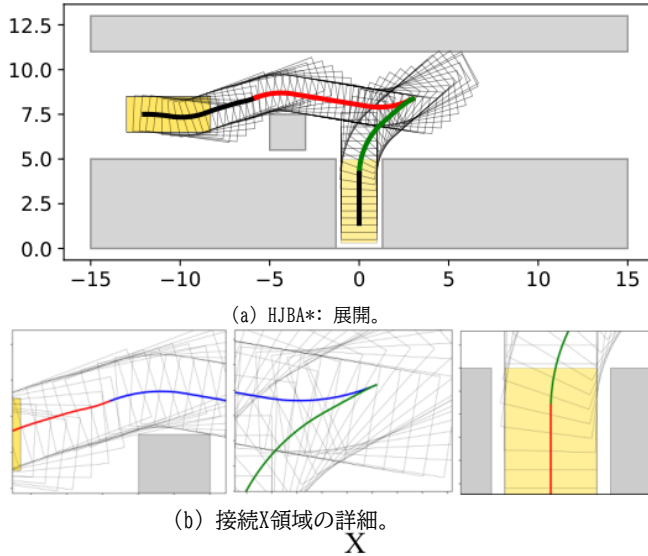


Fig. 4: The highly efficient searching strategy.

図5bに示すように、障害物によって探索が妨害されるような状況では、我々のアルゴリズムはノードを後方に拡張することで適応する。衝突のない経路が特定されると、探索は接続された状態に向かって続けられる。後方探索の段階では、障害物や狭い駐車スペースが発生することが多く、異なる動作解像度や探索戦略が重要である。これらのシナリオにおいて、我々のアルゴリズムは、狭い環境における衝突のない経路の発見に大きく役立つため、後方探索フェーズで双方向モーションプリミティブを採用している。

実証された例は、我々のモーションプリミティブと検索戦略の有効性を効果的に強調している。接続された状態を活用し、適切な探索戦略を採用することで、我々のアプローチは、様々な駐車シナリオにおいてロバストで効率的なプランニングを提供しながら、パラメータ調整の必要性を排除する。

F. 他の双方向A*アプローチとの特徴の区別

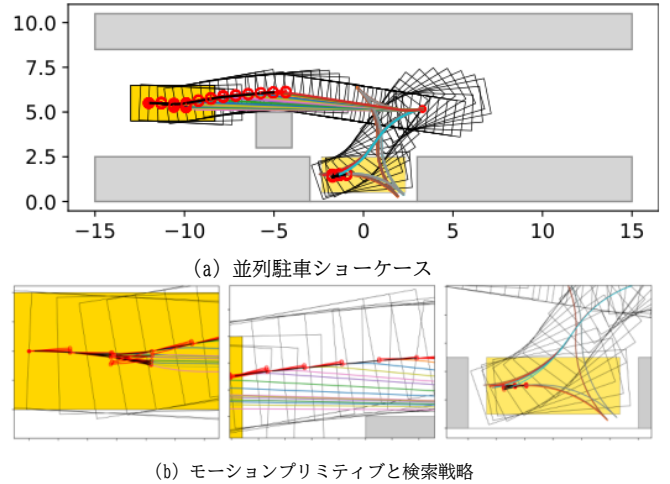


図5: パラメータチューニングを行わない検索戦略のデモ。

A*ベースの手法[39]のような探索ベースのアルゴリズムや、RRTベースの手法[21]のようなサンプリングベースのアルゴリズムでは、双方向探索戦略が広く採用されている。しかし、我々のアルゴリズムは、接続状態の概念を導入することで、これまでの双方向A*アプローチとは異なる。

前方探索がゴールポーズを、後方探索が初期ポーズをターゲットとする従来の双方向探索戦略とは異なり、我々のアルゴリズムは、特定の領域でマージするのではなく、接続された状態に向かって両方の探索を導く。この区別は、クラスター化され狭い駐車場シナリオや複雑な環境など、対象が障害物に囲まれているシナリオで特に有益である。このような場合、従来の双方向探索アルゴリズムは非効率を示す傾向がある。

さらに、我々のアプローチはRS曲線の効率を考慮し、安全な到達可能集合を活用して、接続された状態にガイダンスを提供する。これにより、ヒューリスティックベースの探索アルゴリズムのハイパーパラメータを微調整する必要がなくなる。さらに、我々のモーションプリミティブは、接続された状態に向かってノードを効率的に拡張することを容易にし、アプローチの全体的な有効性に寄与する。備考 4. 我々のアプローチは双方向探索を用いるが

とRS曲線を用いて完全な駐車経路を生成する場合、経路は終始連続的であることに注意することが重要である。本論文の後のセクションで、詳細なデモンストレーションを行う。

V. SV. シミュレーションのセットアップと結果

A. シミュレーションの設定

すべてのシミュレーションは、Intel Core i7-6820HK CPUとGeForce GTX 1070 GPUを搭載したUbuntu 18.04のラップトップ上で実施した。理想的な駐車計画アルゴリズムは、様々な駐車シナリオ(例えば、一般的な駐車シナリオやバリエーション)に対応する汎用性が高く、様々な駐車タスク(例えば、任意の初期ポーズやゴールポーズから)を扱うのに常に頑健であり、分散の少ない駐車タスクを解くのに安定であり、高速な解を提供するために高い計算効率を持つべきである。

表II: アルゴリズムと車両に関連するシミュレーションパラメータのセットアップ

Setup	Parameter	Values
HA*	Grid size	0.5 m
	Yaw resolution	5°
MHHA*	Motion resolution	0.1 m
	Switch back penalty	5.0
	Back penalty	5.0
	Steer change penalty	5.0
	Steer penalty	5.0
HJBA*	前方探索のためのグリッドサイズ	0.5 m
	後方探索のためのグリッドサイズ	0.3 m
	Yaw resolution	5°
	並列計算スレッド	12
Vehicle	車両の幅	2.0 m
	車両の長さ	4.7 m
	Max steering angle	0.6 rad
	Wheelbase	2.7 m
Parking spot size	Perpendicular(w, d, h, α)	2.6, 5.0, 6.0, 90°
	Angle (w, d, h, α)	2.6, 5.0, 6.0, 45°
	Parallel(w, d, h, α)	6.0, 2.5, 6.0, 90°

上記のメトリクスに基づく我々の提案するHJBA*アルゴリズムの顕著な性能を実証するために、まずいくつかの簡単な例によって我々のアルゴリズムを検証し、次にバッチシミュレーションで我々の駐車プランナを評価し、前述の他の最先端の計画アルゴリズムHA* [2]やMulti-heuristic hybrid A* (MHHA*) [44], [45]と比較する。駐車場の大きさと使用したパラメータを表IIに示す。II

B. バリデーションとデモ

本節では、3つの一般的な駐車シナリオを通じて、HJBA*のワークフローを検証し、実証する。

1) 安全な到達可能集合の実証

我々が考える最初のシナリオは、図6aに示すように、狭い空間における垂直な駐車状況である。グリッドの寸法は 61×61 で、開始ポーズは $[-12, 7.5, 0.0]$ に位置し、ゴールポーズは $[0, 1.3, \pi/2]$ に位置する。この文脈では、安全な到達可能集合とは、駐車時にドライバーが最初に通過しなければならない領域を指し、接続された状態は、駐車プロセス中にドライバーが利用できる様々な選択肢を表す。

2つ目の例は、図6bに示すようなアングルパーキングである。このシナリオで使用される Ω_z の寸法は $101 \times 101 \times 101$ であり、開始ポーズは $[-18, 7.5, 0.0]$ に位置し、ゴールポーズは $[-1.7, 3.5, -\pi/4]$ に位置する。この文脈では、ドライバーは駐車場に接近する前に駐車操作を開始しなければならない、その結果、サンプリング状態は主に左エリアに集中する。さらに、サンプリング状態がドライバーを駐車場に戻るように誘導することができるため、最初の駐車場ポーズが駐車場の右エリアに位置することが可能である。

第3のシナリオは、図6cに示すような並列駐車の状態である。このシナリオで使用される計算グリッドの寸法は $101 \times 101 \times 101$ で、開始ポーズは $[-12, 5.0, 0.0]$ に位置し、ゴールポーズは $[-1.35, 1.5, 0.0]$ に位置する。

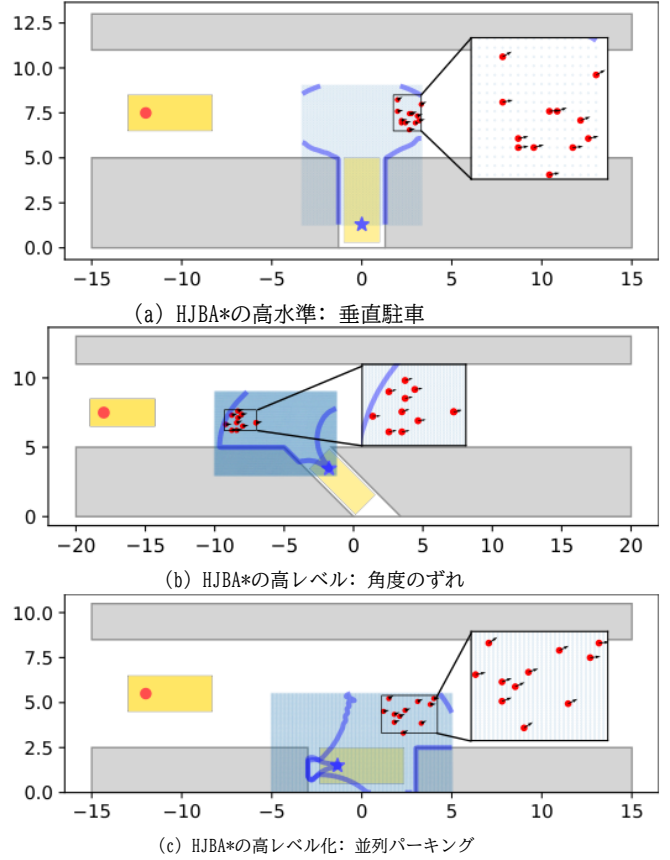


図6: サンプリングチューブの投影と、例のランダムなサンプリング状態。障害物は灰色でパッチされている。赤い円は初期ポーズ、青い星は駐車ゴールのポーズである。水色の領域はBRT計算のためのグリッドである。青の実線は2次元平面に投影されたGの境界であり、12個のランダムなサンプリング状態があり、方向は黒い矢印で赤い点で示されている。

平行駐車は、狭い平行駐車スポットでは、自車両が駐車スポットを離れるために何度も往復する必要があるため、他のタイプの駐車よりも一般的に複雑である。一方、垂直駐車や角度駐車における位置の調整は、走行空間の大きさ h に依存することがほとんどである。平行移動の狭い駐車スペースでは、後続の段階で前進するためのスペースを増やすために、最初に車が戻るが必要になることが多い。この問題に対処するため、最初のステージの S を計算し、次のステージのゴールパーキングセットとして使用する。障害物により駐車地点の後側から駐車できないため、接続状態は右エリアにランダムに分布している。

2) 双方向探索と安全な到達可能集合の協調

図7に、3つの典型的な駐車シナリオにおける探索ベースのアルゴリズムを示す。右側の図から明らかなように、各接続状態 $z_k \in S$ は、並列計算によって得られる灰色の駐車経路 $P(z_0, z_g)$ によって始点と終点に接続される。ユーザは、最も短い長さまたは計算時間で、好ましい経路を選択することができる。駐車場のシナリオにかかわらず、駐車場の経路は主にRS曲線とA*ベースの拡張セグメントで構成される。

これまでの設計目標と同様に、RS曲線の長さは長くなり、グラフベースの展開の長さは短くなり、計算速度が大幅に向上する。

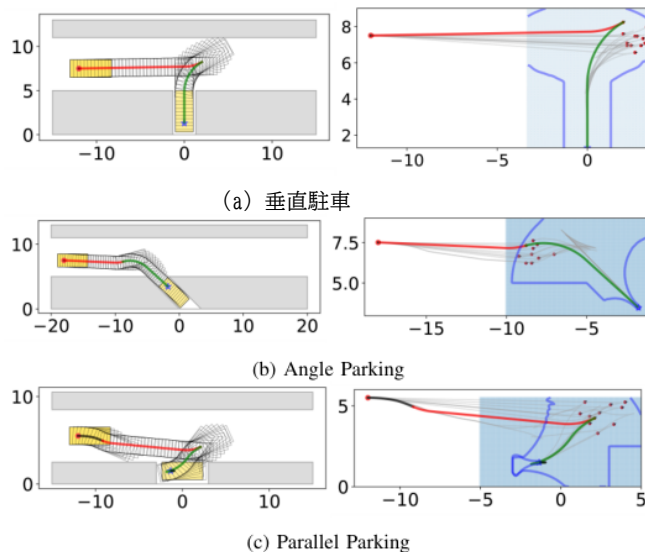


図7:異なる駐車シナリオにおけるHJBA*の説明図。左:駐車経路 $P(z_0, z_g)$ は、右側に対応する線分から構成される。右:赤のセグメントは $RS(z_{Fbest}, z_i^S)$ 、黒のセグメントは $A(z_0, z_{Fbest})$ と $A(z_g, z_{Bbest})$ 、緑のセグメントは $RS(z_{Bbest}, z_i^S)$ を表す。

3) HJBA*の性能実証

我々のHJBA*の優位性を示すために、図8に示す例を示す。我々のHJBA*とHA*アルゴリズムは同じ初期状態 $z_0 = [-12, 5.5, 0, 0]$ から出発し、ゴール状態 $[-1.35, 1.5, 0, 0]$ への経路を見つけることを目的とする。

図8aは、HJBA*におけるノード展開とRSExpansionの詳細を示している。前方探索段階 $(z_0 \rightarrow z_i^S)$ において、 $RS(z_{best}, z_i^S, z_0)$ が存在するような z_{best} 状態に接続する無衝突経路を見つけることが目的である。HJBA*は、安全な到達可能集合から選ばれた連結状態 z_i^S を利用することで、図8bの黒と赤のセグメントで示される $P(z_0 \rightarrow z_i^S)$ を、最小限のノード拡張を伴いながら高速に求める。前述したように、図2に記述され、図3に描かれている我々のノード拡張戦略は、最初に z_i^S に向かって探索する計画を伴う。障害物に遭遇した場合、エージェントの姿勢を調整するために後方ノード展開を行い、その後再び接続状態 z_i^S に前方へ進む。同様に、 $P(z_i^S \leftarrow z_g)$ において、黒いセグメントは比較的短く、緑のセグメント $|RS(z_{best}, z_i^S, z_g)|$ は駐車経路 $P(z_i^S \leftarrow z_g)$ の大きな割合を占め、タスクの難易度を低くしている。その結果、RSExpansionの試行回数はわずかである。この結果は、我々の予想と一致する。

従来のモーションプリミティブでは、ノードが前方に拡張すると、後方に拡張するため、不必要な拡張が発生する。自律駐車は通常狭い環境で行われるため、解決ベースの探索アルゴリズムでは、駐車スポットに近い領域で z_{best} を特定することは困難である。したがって、衝突のないRSExpansion(z_{best}, z_g)を見つけるには、長い時間かタイムアウトになる可能性がある。

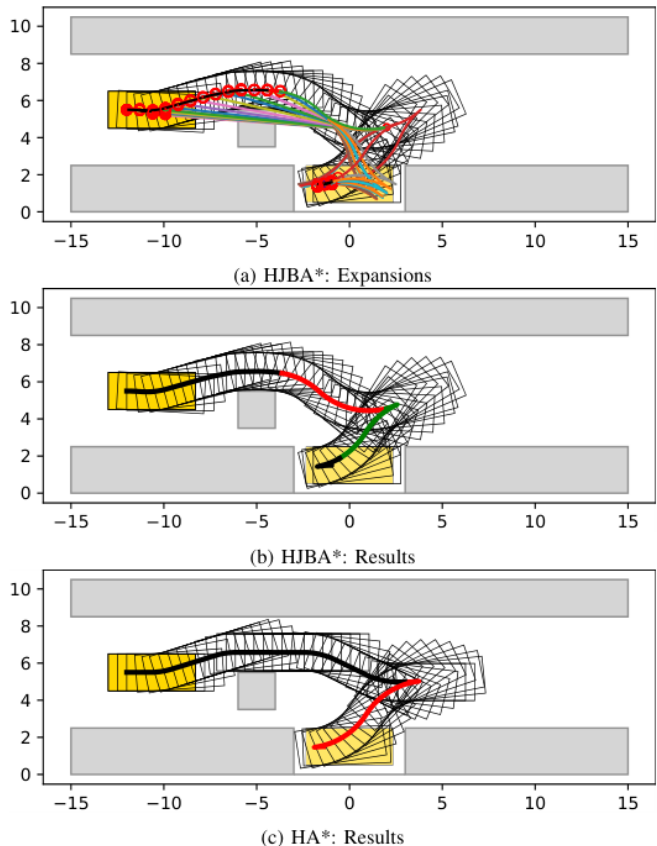


図8:1つのケースにおけるHJBA*とHA*の検索と比較のデモ。左辺(a)と(c)において、カラフルな曲線はRS曲線であり、赤丸は探索過程における拡張ノードの可視化である。右辺(b)と(d)において、黒いセグメントは無衝突A、赤と緑のセグメントは無衝突RSである。

より高解像度のマップやチューニングパラメータを使用することで、この問題は軽減されるが、根本的な解決にはならない。HJBA*は、安全な到達可能集合 S の連結状態を利用することで、この課題を解決する。連結状態は、 $|RS|$ の割合を増やすだけでなく、探索問題を容易にする。

C. 複数駐車場のデモンストレーション

複数の駐車要求を処理するHJBA*の能力を示すために、HJBA*が検証された2つの実用的なシナリオを提供する。2つの駐車シナリオでは、ドライバーが様々なスタート地点から駐車要求を提出し、HJBA*がリアルタイムで駐車ソリューションを生成する。

図9に示すように、HJBA*は複数の駐車要求に対して、信頼性が高く、効果的な管理能力を有していることが明らかになった。ある解決策は他の解決策に比べて時間がかかることに注意すべきである。これは、接続状態の不確実性に起因すると考えられ、自動車が障害物境界に近接しているため、駐車場内の他の位置と比較して、接続状態への直接経路を見つけることがより困難になる。とはいえ、計算時間はまだ許容範囲内にあり、全体的な性能は安定している。

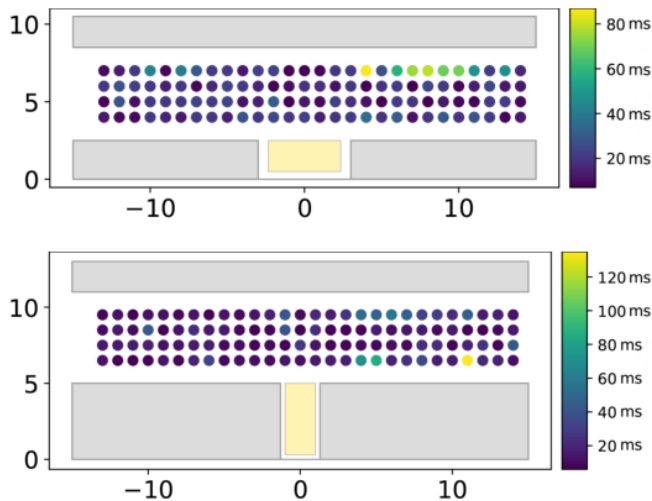


図9:異なる駐車要求の下での並列駐車(上)と垂直駐車(下)におけるHJBA*の解答時間。

D. 狭い複雑な駐車シナリオの処理

このセクションでは、通常の駐車シナリオだけでなく、クラスター化し、複雑で、狭い駐車条件を持つ困難な環境においても、本アルゴリズムの有効性を示すことで、本アルゴリズムの汎用性を実証する。

図10は、評価に使用したセットアップを示している。 $z_0 = [-15, 1.5, 0.0]$, $z_0 = [-15, 6.5, 0.0]$, $z_0 = [-15, 12.5, 0.0]$, $z_0 = [-10, 4.2, \pi/4]$, $z_0 = [6.0, 12.5, \pi/6]$ 。青と緑のパスは我々のアルゴリズムによって生成されたRS曲線を表し、計算効率が良い。

HJBA*アルゴリズムにより、連結点 z_k^i に向かって衝突のないRS曲線を示す準最適姿勢 z_{best} を特定することに成功した。これは、狭い空間に障害物が集まっているものを含む、様々な駐車シナリオを扱う我々のアプローチの能力を強調している。

障害物がクラスター化され、狭い環境において探索ベースのアルゴリズムが直面する重要な課題は、グリッドサイズやヒューリスティックパラメータなどのパラメータを調整する必要性である。これらの調整は、計算時間の増加という代償を払うか、あるいは人間の多大な労力を必要とすることが多い。

しかし、我々の大規模なテストでは、管理された評価環境を確保するために、特に古典的な駐車場シナリオに焦点を当てていることに注意することが重要である。

E. 性能比較

我々のHJBA*アルゴリズムの有効性を総合的に評価し、他の最先端計画アルゴリズムと比較するために、我々は4つの評価基準を設定した：

- i 計算時間: 計算時間は、あるアルゴリズムが駐車場のシナリオを与えられた解を返す時間である。HA*やMHHA*のヒューリスティック関数やHJBA*のBRTのようなオフラインの計算時間は無視される。
- ii ノード番号: 計算時間はノードの拡張数に直接影響され、これは次のことを示している。

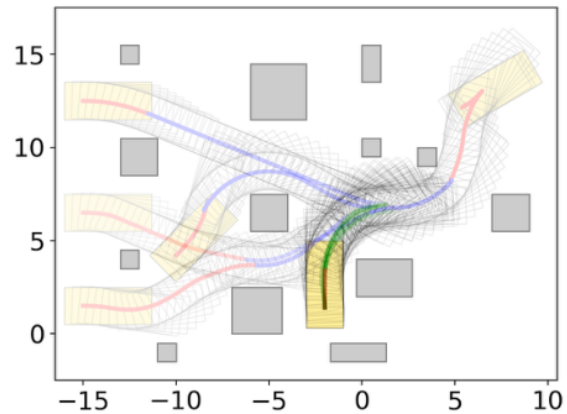


図10:複雑な駐車シナリオのデモ

ヒューリスティック関数の合理性とアルゴリズムの構造の両方について。

- iii Path Length: The path length is calculated as the summation of Euclidean distances between pairs of nodes along the solution.
- iv Direction Changes: When the parking environment is narrow, the parking maneuvers will be composed of forward and backward movements at the cusp.
- v Failure Rate: We define a failure condition for the algorithm where it is deemed unsuccessful if the number of nodes in the closed list exceeds a threshold value, denoted by N . The specific value of N is determined by the complexity of the parking scenario.

1) 計算時間

900回のテストのバッチシミュレーション結果は、我々のHJBA*アルゴリズムが、一般的な駐車シナリオにおいて、一貫した計算速度を示すことを示している。III. 典型的な駐車経路 $P(z_0, z_g)$ は、 $A(z_0, z_{best})$ と $RS(z_{best}, z_g)$ の2つのセグメントから構成され、計算量の大部分は空間でのノード展開により前者に割かれる。HA*とMHHA*は、最小の計算時間から明らかなように、迅速に解を提供できる場合があるが、駐車問題を解決する際の真の課題は、異なる初期状態に対して迅速な解を提供することにある。我々のHJBA*アルゴリズムは、全てのシナリオにおいて、最も速い平均計算速度を維持する。並列駐車の場合、HA*とMHHA*の計算速度が遅いのは、最適な z_{best} を見つけるために必要なノードの過剰な拡張に起因する。ヒューリスティック関数を最適化した後でも、駐車場の近傍に到達するためには、多数のノードを拡張する必要がある。駐車場の近傍に近づくにつれて、故障率は増加し、この周辺の各ノードのヒューリスティックコストは同程度であるため、探索時間は延長される。ガイダンスがなければ、HA*とMHHA*は最適な z_{best} を見つけるために、より多くのノードを拡張しなければならない。

一方、我々のアルゴリズムHJBA*では、双方向メカニズムによって駐車問題を高速に解くのに役立つような連結点がある。我々のHJBA*では、 $P_{(z_0, z_g)}^{BA*} = P(z_0 \rightarrow z_i^S) \div P(z_i^S \leftarrow z_g)$ となる。

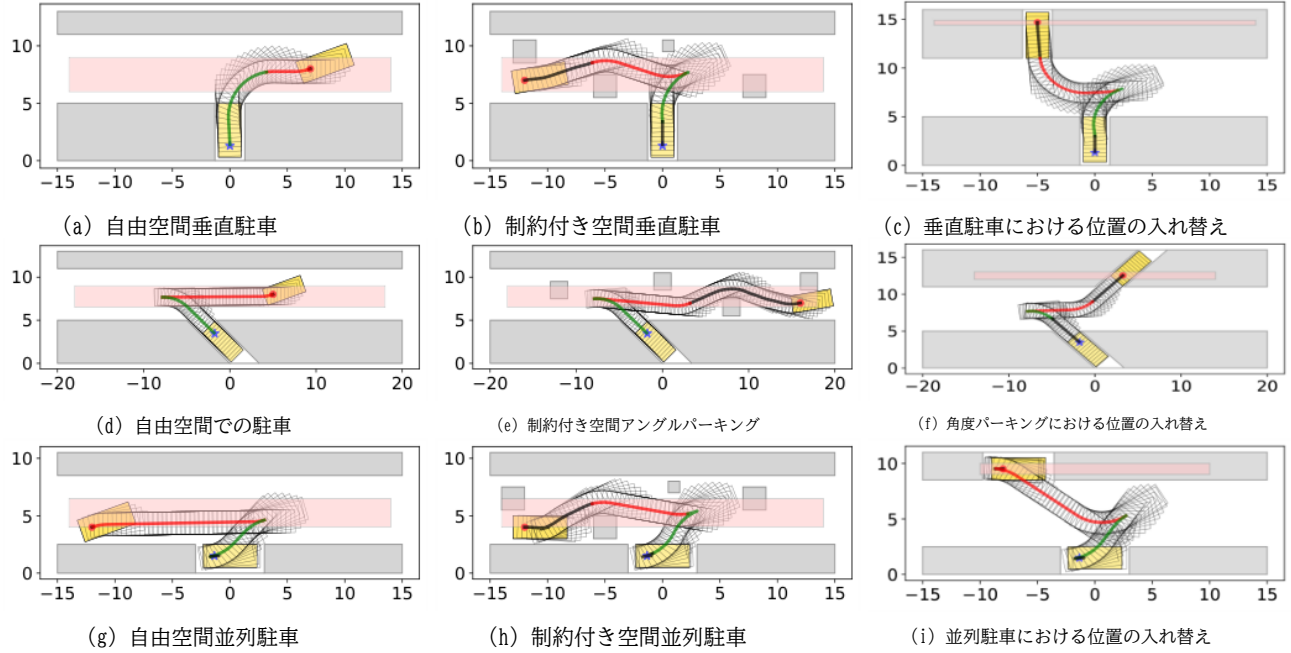


図11:一般的な駐車シナリオにおけるHJBA*の説明図。赤い点は初期ポーズで、ピンクの領域でランダムに生成される。一般的な垂直駐車では、初期姿勢 $x_0 \in [-14, 14]$, $y_0 \in [6.0, 9.0]$, $\theta_0 \in [0, \pi/4]$ である。一般的な角度の駐車では、 $x_0 \in [-18, 18]$, $y_0 \in [6, 8]$, $\theta_0 \in [0, \pi/9]$ を考える。並列駐車では、ランダムな初期駐車姿勢領域は、 $x_0 \in [-14, 14]$, $y_0 \in [4, 6.5]$, $\theta_0 \in [0, \pi/9]$ である。

図11bに示すように、前方探索の目標は z_{Fbest} を見つけ、 z_{Fbest} を z_i^S に無衝突RS曲線で結ぶことである。赤いセグメントは $RS(z_{Fbest}^F, z_i^S)$ であり、駐車問題の計算時間を大幅に節約できる。最悪の場合、HJBA*はHA*やMHHA*に比べて計算時間が短い。障害物が集合した駐車シナリオの場合、主に $A(z_0, z_i^S)$ の増加により計算時間が長くなる。

2) Node Numbers

表IIIから、我々のアルゴリズムはIIIから、我々のアルゴリズムHJBA*は、すべての駐車シナリオで最小の平均ノード数を持つ。この傾向は計算時間と同様である。並列駐車では、HA*とMHHA*の性能は、 z_i^S の助けを借りてHJBA*が影響を受けないランダムな初期状態に対して安定ではないため、悪くなる。とはいえ、いくつかの単純なランダムケースでは、HA*とMHHA*は1つのノードで解を提供できるが、HJBA*は導入されたサンプリング状態により少なくとも2つのノードを必要とする。

3) Path Length

HA*、MHHA*、および我々のHJBA*は、「行くコスト」と「来るコスト」によって導かれる $A(z_0, z_{best})$ セグメントのパスが最適拡張であることが保証されないため、最適探索アルゴリズムに分類される。ただし、 $RS(z_{best}, z_g)$ の区間は解析的に最適である。理論的には、無衝突経路Pに対して、RS曲線の割合が高いほど、経路長が短くなる。垂直駐車と並列駐車両方のシナリオにおいて、我々のHJBA*アルゴリズムは、最も速い計算速度を維持しながら、最も短い平均経路長を達成する。しかし、アングルパーキングでは、HJBA*の平均経路長は長くなる。

これは、連結状態と双方向探索メカニズムが導入され、HJBA*の解に $RS(z_{Fbest}, z_i^S)$ と $RS(z_{Bbest}, z_i^S)$ という2つのRS曲線が生じるためである。この2つの曲線の和が、直接のRS曲線よりも大きくなる可能性がある。この点で、HJBA*アルゴリズムは最適性と計算時間のバランスをとり、これはアングルパーキングの場合にも明らかである。

4) Direction Changes

駐車場の経路におけるカスプの数は、一般的に計算時間ではなく経路の長さに関連付けられる。我々のHJBA*アルゴリズムは、垂直駐車と平行駐車シナリオにおいて、HA*とMHHA*と同程度のカスプ数を示し、その結果、HJBA*の平均経路長が優れている。しかし、アングルパーキングの場面では、HA*やMHHA*に比べ、HJBA*はより多くの犬歯数を示す。

この不一致は、パスの長さのカスプの数の関係に起因すると考えられる。アングルパーキングでは、パーキングパスを構築するために2つのRSカーブを介することで、カスプの数が増加する。逆に、垂直駐車と平行駐車で観察されるカスプの数が同程度であることは、その逆駐車性質に起因している。本研究におけるアングルパーキングは、もっぱら前方パーキングを対象としており、このことが、この特殊なシナリオにおいてHJBA*で観察されるカスプの数が多く一因となっていることに注意することが重要である。

5) Failure rate

HJBA*アルゴリズムが無衝突垂直駐車シナリオで達成した0失敗率は、HA*やMHHA*と同じ有効性を持つ解を見つける能力を示している。しかし、アングル駐車と平行駐車におけるHA*とMHHA*の故障率は比較的高い。グリッドサイズ、ステップサイズ、ヒューリスティック関数のパラメータチューニングは成功率を向上させるかもしれないが、時間のかかるプロセスであり、パラメータチューニングだけでは安定したプランナを保証することはできない。

表III:異なる駐車シナリオとバリエーション試行における評価指標².

Scenarios ¹	Algorithms	Failure Rate	計算時間 (ms)			Node Numbers			Path Length (m)			Direction Changes		
			mean	min	max	mean	min	max	mean	min	max	mean	min	max
a	HA*	0%	568	106	1520	372	90	870	26.9	25.3	28.3	1	0	1
	MHHA*	0%	285	133	573	213	105	394	26.7	25.1	28.3	1	0	1
	Ours	0%	27	10	52	7	2	21	23.9	21.7	26.7	1	0	1
b	HA*	0%	947	5	3973	422	1	1598	12.5	6.6	22.5	1	1	3
	MHHA*	0%	732	4	2430	349	1	1024	12.4	6.6	23.4	1	1	3
	Ours	0%	63	9	306	23	2	80	12.0	8.1	20.8	2	1	3
c	HA*	0%	1135	393	1952	399	149	510	23.7	17.9	30.6	1	1	1
	MHHA*	0%	1058	24	2253	425	12	1002	24.5	17.9	30.6	1	1	1
	Ours	0%	85	23	144	25	7	40	22.8	18.0	29.4	1	1	1
d	HA*	5%	852	10	3727	512	7	1643	15.0	8.1	29.1	1	0	1
	MHHA*	14%	1335	9	4085	688	7	1574	13.7	8.1	28.1	1	0	2
	Ours	0%	82	11	280	13	2	79	17.2	9.3	34.0	3	1	3
e	HA*	2%	901	8	8186	454	1	1521	17.4	8.2	34.2	1	0	3
	MHHA*	1%	1214	6	6955	523	1	1463	16.4	8.2	38.1	1	0	3
	Ours	0%	115	9	885	32	2	176	19.2	9.1	37.3	4	2	7
f	HA*	0%	534	46	2239	169	13	1169	21.8	15.7	31.5	1	1	1
	MHHA*	0%	387	44	3642	117	13	1287	21.0	15.7	27.0	1	1	1
	Ours	0%	187	22	429	21	2	55	22.0	15.9	30.2	4	1	4
g	HA*	58%	1826	96	5632	940	89	1644	16.0	7.7	26.0	3	2	4
	MHHA*	35%	2622	237	7613	1156	163	2524	17.0	7.7	26.8	3	2	5
	Ours	0%	48	11	122	17	8	24	14.4	7.8	22.0	3	2	4
h	HA*	34%	2484	286	7201	875	141	2017	14.7	7.2	32.2	4	2	5
	MHHA*	37%	1552	249	4739	613	140	1455	14.0	6.9	28.7	4	2	5
	Ours	0%	81	13	278	21	8	84	14.0	7.5	25.7	3	2	5
i	HA*	49%	5581	142	10793	2163	87	3929	24.2	18.0	34.7	6	4	6
	MHHA*	23%	3298	272	8139	1531	196	3213	23.2	18.9	29.9	6	4	6
	Ours	0%	199	23	564	33	17	114	20.1	17.5	25.3	5	4	8

¹ シナリオ中の文字は、図 11 の文字に対応する。² 失敗事例の評価指標は含まれていない。

注目すべきは、ランダムな一般的な並列駐車におけるHA*の失敗率が、クラスタ化された並列駐車シナリオにおける失敗率よりもさらに高いことである。後者の障害物は、いくつかの初期駐車状態を駐車ゴールのポーズに近づけ、それによってHA*の成功率を向上させるからである。一方、我々のHJBA*アルゴリズムは、全てのテストにおいて、0.0%の失敗率で極めて安定した性能を示す。

VI. CONCLUSIONS

本論文では、一般的な駐車シナリオにおいて、高い計算効率と安定した性能という利点を持つ、新しい駐車計画アルゴリズムを提示する。HJBベースのPDEの解法により、各駐車シナリオのBRTを計算する方法を示した。広範なシミュレーション(異なる垂直駐車、アングル駐車、平行駐車で900回のテスト)により、我々のアルゴリズムが最も低い故障率0%と極めて速い計画時間(数十~200ミリ秒)を維持することを証明した。今後の課題としては、動的駐車アプリケーションへの拡張、変異エージェントを含む駐車シナリオの調査、車両プラットフォームへの実装などが挙げられる。

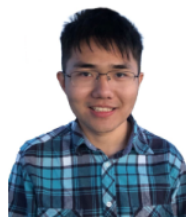
REFERENCES

- [1] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 972–983, 2021.
- [2] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," *Ann Arbor*, vol. 1001, no. 48105, pp. 18–80, 2008.
- [3] A. Thirugnanam, J. Zeng, and K. Sreenath, "Duality-based convex optimization for real-time obstacle avoidance between polytopes with control barrier functions," in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 2239–2246.
- [4] W. Wang, Y. Song, J. Zhang, and H. Deng, "Automatic parking of vehicles: A review of literatures," *International Journal of Automotive Technology*, vol. 15, no. 6, pp. 967–978, 2014.
- [5] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [6] M. Likhachev, D. I. Ferguson, G. J. Gordon, A. Stentz, and S. Thrun, "Anytime dynamic a*: An anytime, replanning algorithm," in *ICAPS*, vol. 5, 2005, pp. 262–271.
- [7] A. Nash, K. Daniel, S. Koenig, and A. Felner, "Theta*: Any-angle path planning on grids," in *AAAI*, vol. 7, 2007, pp. 1177–1183.
- [8] S. M. LaValle *et al.*, "Rapidly-exploring random trees: A new tool for path planning," in *TR 98-11, Computer Science Dept., Iowa State Univ.* Ames, IA, USA, 1998.
- [9] J. Cortés and T. Simeon, "Sampling-based tree planners (rrt, est, and variations)," 2021.
- [10] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The international journal of robotics research*, vol. 20, no. 5, pp. 378–400, 2001.
- [11] S. He, J. Zeng, and K. Sreenath, "Autonomous racing with multiple vehicles using a parallelized optimization with safety guarantee using control barrier functions," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 3444–3451.
- [12] P. Zips, M. Bock, and A. Kugi, "A fast motion planning algorithm for car parking based on static optimization," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 2392–2397.
- [13] J. Leu, Y. Wang, M. Tomizuka, and S. Di Cairano, "Autonomous vehicle parking in dynamic environments: An integrated system with prediction and motion planning," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 10 890–10 897.

- [14] X. Chi, Z. Liu, J. Huang, F. Hong, and H. Su, "Optimization-based motion planning for autonomous parking considering dynamic obstacle: A hierarchical framework," *arXiv preprint arXiv:2210.13112*, 2022.
- [15] I. Mitchell, A. Bayen, and C. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.
- [16] C. Tomlin, G. Pappas, J. Lygeros, D. Godbole, S. Sastry, and G. Meyer, "Hybrid control in air traffic management systems," *IFAC Proceedings Volumes*, vol. 29, no. 1, pp. 5512–5517, 1996.
- [17] M. Chen, J. C. Shih, and C. J. Tomlin, "Multi-vehicle collision avoidance via hamilton-jacobi reachability and mixed integer programming," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 1695–1700.
- [18] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Sastry, "Reach-avoid problems with time-varying dynamics, targets and constraints," in *International Conference on Hybrid Systems: Computation & Control*, 2015.
- [19] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-jacobi reachability: A brief overview and recent advances," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 2242–2253.
- [20] H. Banzhaf, L. Palmieri, D. Nienhüser, T. Schamm, S. Knoop, and J. M. Zöllner, "Hybrid curvature steer: A novel extend function for sampling-based nonholonomic motion planning in tight environments," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–8.
- [21] J.-H. Jhang, F.-L. Lian, and Y.-H. Hao, "Forward and backward motion planning for autonomous parking using smooth-feedback bidirectional rapidly-exploring random trees with pattern cost penalty," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, 2020, pp. 260–265.
- [22] J. Reeds and L. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific journal of mathematics*, vol. 145, no. 2, pp. 367–393, 1990.
- [23] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.
- [24] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "Chomp: Covariant hamiltonian optimization for motion planning," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1164–1193, 2013.
- [25] T. S. Lembono and S. Calinon, "Probabilistic iterative lqr for short time horizon mpc," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 579–585.
- [26] E. Dantec, R. Budhiraja, A. Roig, T. Lembono, G. Saurel, O. Stasse, P. Fernbach, S. Tonneau, S. Vijayakumar, S. Calinon, M. Taix, and N. Mansard, "Whole body model predictive control with a memory of motion: Experiments on a torque-controlled talos," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 8202–8208.
- [27] K. Kondak and G. Hommel, "Computation of time optimal movements for autonomous parking of non-holonomic mobile platforms," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 3, 2001, pp. 2698–2703 vol.3.
- [28] T. S. Lembono, A. Paolillo, E. Pignat, and S. Calinon, "Memory of motion for warm-starting trajectory optimization," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2594–2601, 2020.
- [29] Y. Wang, "Improved a-search guided tree construction for kinodynamic planning," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 5530–5536.
- [30] P. Zips, M. Böck, and A. Kugi, "Optimisation based path planning for car parking in narrow environments," *Robotics and Autonomous Systems*, vol. 79, pp. 1–11, 2016.
- [31] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of mathematics*, vol. 79, no. 3, pp. 497–516, 1957.
- [32] L. Cai, H. Guan, H. L. Zhang, X. Jia, and J. Zhan, "Multi-maneuver vertical parking path planning and control in a narrow space," *Robotics and Autonomous Systems*, vol. 149, p. 103964, 2022.
- [33] H. Vorobieva, N. Minoiu-Enache, S. Glaser, and S. Mammar, "Geometric continuous-curvature path planning for automatic parallel parking," in *2013 10th IEEE INTERNATIONAL CONFERENCE ON NETWORKING, SENSING AND CONTROL (ICNSC)*, 2013, pp. 418–423.
- [34] C. Sungwoo, C. Boussard, and B. d'Andréa Novel, "Easy path planning and robust control for automatic parallel parking," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 656–661, 2011.
- [35] J. Zhou, R. He, Y. Wang, S. Jiang, Z. Zhu, J. Hu, J. Miao, and Q. Luo, "Autonomous driving trajectory optimization with dual-loop iterative anchoring path smoothing and piecewise-jerk speed optimization," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 439–446, 2021.
- [36] P. Abbeel, D. Dolgov, A. Y. Ng, and S. Thrun, "Apprenticeship learning for motion planning with application to parking lot navigation," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 1083–1090.
- [37] Z. Zhang, R. Wu, Y. Pan, Y. Wang, Y. Wang, X. Guan, J. Hao, J. Zhang, and G. Li, "A robust reference path selection method for path planning algorithm," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4837–4844, 2022.
- [38] S. Sedighi, D.-V. Nguyen, and K.-D. Kuhnert, "Guided hybrid a-star path planning algorithm for valet parking applications," in *2019 5th International Conference on Control, Automation and Robotics (ICCAR)*, 2019, pp. 570–575.
- [39] B. Adabala and Z. Ajanovic, "A multi-heuristic search-based motion planning for autonomous parking," in *30th International Conference on Automated Planning and Scheduling: Planning and Robotics Workshop*, 2020.
- [40] S. O. R. Fedkiw and S. Osher, "Level set methods and dynamic implicit surfaces," *Surfaces*, vol. 44, no. 77, p. 685, 2002.
- [41] M. Chen, S. L. Herbert, H. Hu, Y. Pu, J. F. Fisac, S. Bansal, S. Han, and C. J. Tomlin, "Fastrack: a modular framework for real-time motion planning and guaranteed safe tracking," *IEEE Transactions on Automatic Control*, vol. 66, no. 12, pp. 5861–5876, 2021.
- [42] J. Zeng, B. Zhang, Z. Li, and K. Sreenath, "Safety-critical control using optimal-decay control barrier function with guaranteed point-wise feasibility," in *2021 American Control Conference (ACC)*, 2021, pp. 3856–3863.
- [43] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.
- [44] S. Aine, S. Swaminathan, V. Narayanan, V. Hwang, and M. Likhachev, "Multi-heuristic a*," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 224–243, 2016.
- [45] J. Huang, Z. Liu, X. Chi, F. Hong, and H. Su, "Search-based path planning algorithm for autonomous parking: Multi-heuristic hybrid a*," *arXiv preprint arXiv:2210.08828*, 2022.



Xuemin Chiは、2017年に中国瀋陽市の瀋陽工科大学から車両工学の工学学士号を、M. S. 中国・大連理工大学より2019年に車両工学の学位を取得。
現在、博士号取得を目指している。中国杭州市浙江大学制御工学部にて博士号取得。研究テーマは、モーションプランニング、安全なモデル予測制御アルゴリズム。



Jun Zengは、2022年に米国カリフォルニア大学パークレイ校機械工学科で制御とロボット工学の博士号を取得し、Diplを取得した。2017年にフランスのEcole Polytechniqueで工学学士号を、2016年に中国のShanghai Jiao Tong University (SJTU)で学士号を取得。研究テーマは、最適化、制御、計画、学習と様々なロボットプラットフォームへの応用の交差点にある。



Jihao Huangは、杭州天子大学(中国・杭州市)で2020年に自動化のB. Eng. 学位を取得した。現在、博士号取得を目指している。現在の研究テーマは、マルチロボットシステム、運動計画、制御理論などである。



Zhitao Liu (M'13) received the B.S. degree from Shandong University, China, in 2005, and the Ph.D. 2010年、中国杭州の浙江大学にて制御科学・工学の学位を取得。2011年から2014年まで、シンガポールのTUM CREATEで研究員を務める。2015年から2016年まで浙江大学助教授、2017年から2021年まで浙江大学准教授を務め、現在は浙江大学サイバーシステム制御研究所教授。現在の研究テーマは、ロバスト適応システムとエネルギー管理システム。

control, wireless transfeー管理システム。



ホンイエ・スー(SM'14)は1969年生まれ。理学士号を取得。1990年に中国江蘇省南京化学技術大学から産業自動化の学位を取得し、M. S. 1993年に浙江大学(中国・杭州市)で、1995年に浙江大学(中国・杭州市)で工業オートメーションの博士号を取得。

1995年から1997年まで浙江大学化学工学部講師。1998年から2000年まで浙江大学先端プロセス制御研究所准教授、

現在はサイバーシステム制御研究所教授。現在の研究テーマは、ロバスト制御、時間遅延システム、先進プロセス制御の理論と応用。