

動的障害物を考慮した自律駐車のための最適化ベースのモーションプランニング：階層的フレームワーク

Xuemin Chi¹, Zhitao Liu¹, Jihao Huang¹, Feng Hong¹, Hongye Su¹

1. State Key Laboratory of Industrial, Control Technology, Zhejiang University, Hangzhou, 110004
E-mail: chixuemin@zju.edu.cn, ztliu@zju.edu.cn, jihaoh@zju.edu.cn, hogfeg@zju.edu.cn, hysu@ipc.zju.edu.cn

概要: 本論文では、グラフ探索アルゴリズムとモデル予測制御を統合し、制約環境における自律走行車(AV)の効率的な駐車操縦を促進する階層的フレームワークを紹介する。高レベル計画フェーズでは、従来のハイブリッドA*の最適化された変形であるシナリオベースのハイブリッドA*(SHA*)を組み込み、静的障害物を考慮しながら初期経路を生成する。このグローバルパスは、低レベルNLP問題の初期推測として機能する。低レベル最適化フェーズでは、動的障害物を回避するために、非線形モデル予測制御(NMPC)ベースのフレームワークが展開される。SHA*の性能は148のシミュレーションシナリオを通して経験的に検証され、提案された階層的フレームワークの有効性はリアルタイム並列駐車シミュレーションによって実証された。

Key Words: autonomous parking, trajectory planning, model predictive control, dynamic obstacles

1 INTRODUCTION

自律走行型駐車システムは、自律走行技術の重要な構成要素である。これらのシステムは、センシング、ローカライゼーション、意思決定、プランニング、制御など、相互に関連するいくつかのモジュールから構成されている[1]。本稿では、自律駐車システムの計画モジュールにおける課題に対処するため、焦点を絞る。自律駐車システムにおいて、計画モジュールは、車両が障害物に衝突することなく指定された空間に駐車することを可能にする実行可能な軌道を生成する主な役割を担っている。この軌道は、その後、低レベル制御モジュールによって実行される[2]。軌道計画のタスクは、快適性と衝突回避という2つの大きな懸念を包含している。最適な軌道は、乗客の快適性や車両の安定性などの要因も考慮しながら、時間を最小化する。この文脈において、我々は、動的障害物が発生する制約環境においても、安全で効率的な駐車軌道を生成するために、グラフ探索アルゴリズムと非線形モデル予測制御法を統合した階層的フレームワークを提案する。

1.1 グラフ探索に基づく方法

グラフ探索法は、一般的な最適化手法と比較して計算効率が良いため、経路計画の領域でよく使われる手法である。このアプローチでは、環境はグリッドに離散化され、その中でノードは特定のルールに基づいてサンプリングされる。その後、アルゴリズムは、所望の経路を形成する最適なノードを探索する。

これらの方法は、一般的に低次元空間では計算効率が良いが、高次元空間では性能が劣化する。さらに、結果として得られるパスは、しばしば最適でない。

A*やハイブリッドA[3]のような従来の決定論的グラフ探索アルゴリズムは、サンプリングに固定モーションプリミティブに依存している。ハイブリッドA*のモーションプリミティブは、特に車両の非ホロノミックな制約に対応するのに適している。Rapidly-exploring Random Trees (RRT) [4]やその亜種[5, 6]のような確率的グラフ探索技術は、グリッド内でランダムサンプリングを採用している。これらの方法はより柔軟であるが、曲率不連続性を持つパスを生成することが多く、さらなる改良なしにすぐに使用するには不向きである。

1.2 Related work

自律駐車のための経路計画と軌道計画の展望を支配するのは、探索に基づく方法と最適化に基づく方法の2つの主要なアプローチである。前者は計算効率が非常に高いが、後者はモデルベースのアルゴリズムにより、車両の動的または運動学的特性をより微妙に考慮することができる。最適化ベースの手法は、数学的モデリングによって、快適性や安定性といった複雑な制約を扱う能力も持っている。Zhangら[7]は、最適化ベースの衝突

avoidance (OBICA) algorithm, which reformulates collision avoidance as smooth constraints using duality and Slater's condition. Despite these advances, their method is not suitable for real-time applications and struggles with dynamic obstacles.

モデル予測制御は、複数の制約を効果的に管理する能力があるため、軌道計画や追跡に広く採用されている[8]。Solopertら[9]は、チューブベースのロバストMPCフレームワークにOBICAを適用した。

本研究の一部は、中国国家重点研究開発プログラム(助成金NO. 2021YFB3301000)、中国国家自然科学基金創造研究グループ科学基金(助成金NO. 61621002)、中国国家自然科学基金(NSFC:62173297)、浙江省重点研究開発プログラム(助成金NO. 2021C01198, 2022C01035)の支援を受けた。

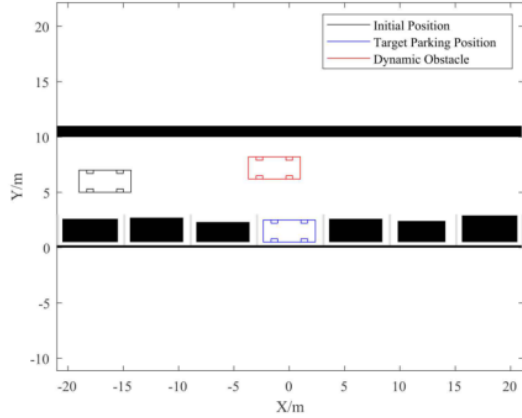


図1: 動的な障害物(赤実線)を避けながら、AVが最初から最後まで駐車操作を実行する様子(黒実線)

を考慮した。しかし、構造化された道路におけるリアルタイム適用のロバスト性は十分に確立されていない。Brüdigamら[10]は、追い越し操作のための確率的MPCスキームを実装したが、前方走行と後方走行の両方を含む駐車シナリオの複雑さには対処していない。

貢献する: 本論文では、リアルタイム駐車操作を行うために設計された新しい階層的フレームワークを提示する。我々の研究の貢献は以下のように要約される:

- 制約のある環境における一般的な駐車シナリオに対応できる階層的なフレームワークを紹介する。
- 我々は、シナリオベースハイブリッドA*(SHA*)と呼ばれる伝統的なハイブリッドA*の高速な変形を提案し、その計算上の利点は様々な文脈で活用できる。
- 我々のフレームワークは、動的な障害物を明示的に考慮することで、自律的な駐車操作のリアルタイム実装を可能にする。

2 問題の説明

本節では、駐車シナリオを検討する。一シオンと問題定式化の概要を説明する。図1に示すように、制約付き駐車シナリオに注目する。この環境では、AVは与えられたスタート構成から所定のゴール構成に移行することを目指す。駐車場内に動的障害物(DO)が存在することが大きな課題となっている。AVは目的を達成するだけでなく、DOとの衝突を回避する駐車マヌーバを実行しなければならない。注目すべきは、シナリオ全体が2次元空間でモデル化されていることである。

2.1 車両の説明

検討したシナリオでは、駐車操作は低速で発生する。車両のダイナミクスをモデル化するために、運動学的自転車モデルを採用する:

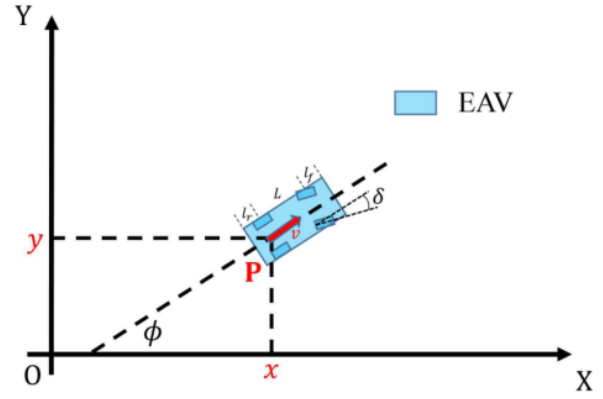


図2: 提案する階層型駐車フレームワーク

$$\dot{z} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \cos(\phi) \\ v \sin(\phi) \\ \frac{v}{L} \tan(\delta) \\ a \end{bmatrix}, u = \begin{bmatrix} \delta \\ a \end{bmatrix}, \quad (1)$$

ここで、システム状態 $z_t \in \mathbb{Z} \subseteq \mathbb{R}^4$ 、 x と y はグローバル座標におけるリアアクスルの中心位置に対応する状態である。状態 ϕ は x 軸に関するヨー角、 v はリアアクスルの中心に対する速度である。前面ステアリング角 δ_f と加速度 a は入力 $u_t \in \mathbb{U} \subseteq \mathbb{R}^2$ である。 L は図2に示すリアアクスルとフロントアクスルの間の長さである。

備考 1. AVの駐車速度が5m/s以下と仮定するため、タイヤのスリップ角や慣性効果は無視できる。

タイトパーキングロットでは、全次元モデルによりAVはより保守的にナビゲートできる。制御されたAVを全次元物体として以下のようにモデル化した。

$$\mathbb{E}(z_t) := R(\phi_t) \mathbb{B}_0 + T(x_t, y_t), \quad (2)$$

ここで、 $z_t \in \mathbb{Z} \subseteq \mathbb{R}^4$ は時刻 t におけるAVの状態、 $\mathbb{E}(z_t)$ はAVが占める「空間」、 $R(-)$ は回転行列、 $T(-)$ は並進演算、 \mathbb{B}_0 は凸多面体で表される矩形である。

$$\mathbb{B}_0 := \{y \in \mathbb{R}^2 \mid Gy \leq g\}, \quad (3)$$

ここで、 G と g は車の形に関係し、 $G = [1, 0; 0, 1; -1, 0; 0, -1]$ 、 $g = [L/2, W/2, L/2, W/2]$ であり、AVの運動は回転と並進による多面体とみなすことができる。

2.2 障害物の説明

本論文では、AVが避けなければならない駐車場内の $M \in \mathbb{N}$ 、 $M \geq 0$ の動的障害物を考える。DOはコンパクトな多面体として記述することができる。

$$\mathbb{O}_t^m := \{y \in \mathbb{R}^2 \mid A_t^m y \leq b_t^m\}, m = 1, \dots, M, \quad (4)$$

ここで、 $A_t^m \in \mathbb{R}^{h_m \times 2}$ と $b_t^m \in \mathbb{R}^{h_m}$ は時刻 t における障害物に関する行列であり、 h_m は m 番目の障害物を定式化する超平面の数である。

仮定1. D0の将来のNステップの情報は既知であると仮定した。将来のショートホライズンの情報は、予測可能であると考えられ、現実には妥当な情報でもある。と定義できる。

$$\mathbb{O}_{k+t|t}^m := \left\{ y \in \mathbb{R}^2 \mid A_{k+t|t}^m y \leq b_{k+t|t}^m \right\}, k = 0, \dots, N, \quad (5)$$

2.3 衝突回避の説明

AVのタスクは、すべての障害物を回避しながら駐車マヌーバを実行することです(4)。形式的には、衝突回避とは、AVとD0が占有する空間の交点为空であることを意味する。dist関数を用いると、次のように表すことができる。

$$\text{dist}(\mathbb{E}(z_t), \mathbb{O}_t^m) \geq d_{\min}, \forall m = 1, \dots, M \quad (6)$$

ここで、dist(-)は多面体間の最短距離である。衝突回避(6)は一般に非凸かつ非微分であるため、滑らかな距離を提供する命題を利用し、定式化されている。

命題1. 双対変数と強い双対性を導入することにより、[7]で確立された等価な時変制約は、以下のように表される。

$$(A_k^m T(z_k) - b_k^m)^\top \lambda_k^m - g^\top \mu_k^m > d_{\min} \quad (7a)$$

$$G^\top \mu_k^m + R(z_k)^\top A_k^m \lambda_k^m = 0, \quad (7b)$$

$$\|A_k^m \lambda_k^m\| \leq 1, \quad (7c)$$

$$\lambda_k^m \geq 0, \mu_k^m \geq 0, \quad (7d)$$

$$\forall k = 0, \dots, N-1, \quad m = 1, \dots, M,$$

ここで、 $\lambda_k^m \in \mathbb{R}^m$ はm番目の障害物に関連する双対変数、 $\mu_k^m \in \mathbb{R}^4$ はAVに関連する双対変数である。(7a)-(7c)は(6)を制約条件として明示的に再定式化し、シナリオがタイトになるにつれてパラメータ d_{\min} を小さく設計することができる。

3 階層的構文解析のフレームワーク

提案する自律駐車システムのアーキテクチャを本節で解明し、図3にグラフ化する。階層構造は、高レベルの経路計画と低レベルの軌道最適化という2つの主要な構成要素を統合している。

高レベルの経路計画フェーズでは、シナリオベースのハイブリッドA*(SHA*)アルゴリズムを利用して、初期実現可能な経路を生成する。このアルゴリズムは、時間情報のない $[x, y, \phi]$ として表される粗い経路を決定するために、初期車両と目標車両の構成を考慮する。この粗い経路に速度プロファイルを追加するために、最小時間最適制御戦略を適用する。このフェーズでは、時間的ダイナミクスでパスを豊かにし、実世界のアプリケーションに適したものにします。

低レベルコントローラは、精密化機構として機能する。最小時間経路を参照として、非線形MPCアルゴリズムはD0を組み込んでリアルタイムの軌道を生成する。これにより、結果として得られる軌道は衝突がないだけでなく、乗客の安定性と快適性を維持することができる。

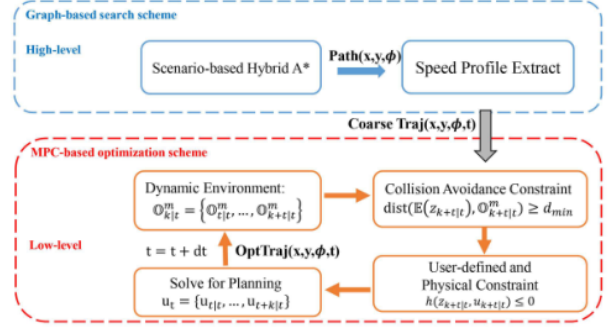


図3:提案する階層型駐車フレームワーク

これらの要素を合成することで、提案するフレームワークは、計算効率と実世界の実用性のバランスをとりながら、自律駐車問題の包括的な解を実現する。

3.1 シナリオベースハイブリッドA*(SHA*)アルゴリズムによる高レベル経路計画

このサブセクションでは、SHA*アルゴリズムを採用することによってもたらされる改善に焦点を当て、高レベルの経路計画コンポーネントに専念する。従来のHA*アルゴリズムでは、ノード拡張時に、背中動き、方向転換、操舵角、操舵角勾配など、いくつかのペナルティパラメータを利用している。これらのペナルティパラメータを微調整することは、しばしば時間のかかる作業となる。さらに、固定されたパラメータセットは、様々な初期駐車条件において最適なパフォーマンスを提供しない可能性がある。これらの限界に対処するために、特定の駐車シナリオに応じてペナルティパラメータを適応させるSHA*アルゴリズムを紹介する。SHA*内のノード拡張プロセスを調整するために、2つのシナリオ固有のペナルティが実装されている。アル駐車場シナリオのために開発されたとはいえ、中核的な設計戦略は他の文脈にも転用可能である。

3.1.1 ヤウ角ペナルティ

最初のペナルティはアルゴリズム1の4行目から11行目までに指定される。並列駐車のようなシナリオでは、従来のハイブリッドA*アルゴリズムは、ヨー角が60度から90度の範囲で多数のノードを展開する傾向がある。このような拡張は、駐車を成功させるためには非現実的であることが多い。この問題を軽減するために、図4に示すペナルティ機構では、ヨー角探索を30度の範囲に制限している。ヨー角がこの範囲を超えるノードにはペナルティが漸増的に適用される。

3.1.2 x軸のオーバートラベルペナルティ

2つ目のペナルティは、ノード拡張時にx軸に沿った過剰拡張を抑制することを目的とする。駐車場シナリオの文脈では、一般に、指定された駐車場空間への車両の配置を確定する前に、近位位置に戻ることが賢明である。このため、SHA*アルゴリズムは、意図した駐車位置からのユークリッド距離に基づいてノードにペナルティを課す。

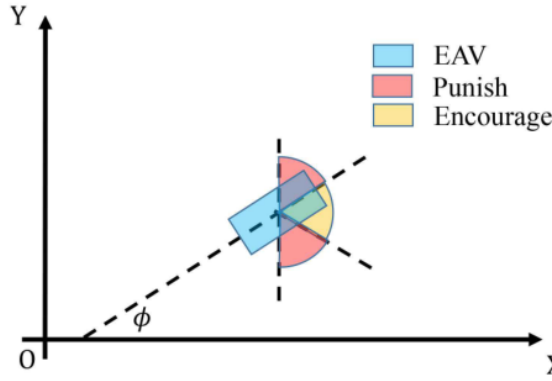


図4:冗長なノード拡張を最小化するための戦略

このメカニズムにより、探索アルゴリズムは目標位置に近いノードに偏り、それによって過大評価を最小化することができる。移動の効率化、計算効率の最適化などである。第二のペナルティは、x軸に沿ったノードの過伸長を最小化することを目的とする。駐車場シナリオにおける一般的なアプローチは、最終的に目標位置に落ち着く前に、おおよその位置に戻すことである。したがって、意図した駐車場からのユークリッド距離に基づいて、ノードにペナルティを課す。これにより、アルゴリズムがターゲットにより近いノードを優先するように促し、オーバートラベルを減らすことができる。

結論として、SHA*アルゴリズムは、自律駐車システムにおいて、より適応的で効率的な高レベル経路計画パラダイムに貢献する。シナリオに基づくペナルティをインテリジェントに取り入れることで、このアルゴリズムは計算コストを最小化するだけでなく、多様な駐車場構成に対して、より焦点を絞った実用的な探索戦略を提供する。これらの改良により、SHA*は、特に自律走行車の駐車領域において、従来のHA*アルゴリズムよりも大きな進歩であると位置づけられる。

アルゴリズム1:SHA*ノード制約法

```

input : node  $N_{current}$ , penalty parameters  $p$ 
output: next node  $N_{next}$ 

1 ノードの状態情報には、 $x, y, \phi, d$  が含まれる;
2 ノードコスト情報には、 $f, g, h$ 、コストが含まれる;
3  $count \leftarrow 0$ ;
4 while  $count \leq 6$  do
5   if  $N_{parent.\phi} \geq limit\ angle$  then
6      $N_{next.cost} \leftarrow N_{parent.cost} + \frac{|\phi - lim}{\pi/2 - lim} * p$ ;
7   else
8      $N_{next.cost} \leftarrow N_{parent.cost}$ ;
9   end
10   $count \leftarrow count + 1$ 
11 end
12 if  $N_{next.x} \geq limit\ x$  then
13    $N_{next.f} \leftarrow N_{next.cost} + \frac{|x - x_{lim}}{x_{target} - x_{lim}} * f_{Eu-dist}$ 
14   ;
15 else
16    $N_{next.f} \leftarrow N_{next.cost} + f_{Eu-dist}$ ;
17 end

```

速度プロファイルは最適制御問題によって取得される。高レベルの主なステップをAlgorithm2にまとめる。

アルゴリズム2:SHA*ベースの高レベル

```

input :  $z_{initial}, z_{target}, O$  と  $E$  の構成出力:
軌道  $z_{ref}$ .

1 SHA*によるグローバルパス $[x, y, \phi]$ の探索
2 粗い軌跡 $[x, y, \phi, v]$ をOCPで計算する。
3 粗い軌跡 $z^{ref}$ は低レベルの参照軌跡として用いられる。

```

3.2 低レベル: モデルベース最適化スキーム

以下では、提案する非線形MPC方式を紹介する。局所軌道リファレンスは、以下の条件で生成される。

current state of the AV. $z_{t+k|t}^{high-level} = [z_{t+k|t}^{ref}, \dots, z_{t+k|t}^{ref}]^T$ is generated from a start point. The start point is obtained by searching the closest point of reference trajectory to the current state.

駐車場操縦のためのSHA*-NMPC定式化は、以下のように示されるNLPとして定式化される。

$$\min_{z, u, \lambda, \mu} \sum_{k=0}^{N-1} J_1(z_{k+t|t}, u_{k+t|t}) + J_2(z_{N+t|t}) \quad (8a)$$

$$\text{s.t. } z_{k+t+1|t} = f(z_{k+t|t}, u_{k+t|t}), \forall k=0, \dots, N-1 \quad (8a)$$

$$z_{k+t|t} \in \mathcal{Z}, u_{k+t|t} \in \mathcal{U}, \forall k=0, \dots, N \quad (8b)$$

$$z_t = z_t \quad (8c)$$

$$(A_{k+t|t}^m T(z_{k+t|t}) - b_{k+t|t}^m)^T \lambda_{k+t|t}^m - g^T \mu_{k+t|t}^m > d_{min} \quad (8d)$$

$$G^T \mu_{k+t|t}^m + R(z_{k+t|t})^T A_{k+t|t}^m \lambda_{k+t|t}^m = 0, \quad (8e)$$

$$\|A_{k+t|t}^m \lambda_{k+t|t}^m\| \leq 1, \quad (8f)$$

$$h(z_{k+t+1|t}, u_{k+t|t}) \leq 0, \quad (8g)$$

$$\lambda_{k+t|t}^m \geq 0, \mu_{k+t|t}^m \geq 0, \quad (8h)$$

$$\forall k=0, \dots, N-1, \quad m=1, \dots, M,$$

ここで、(8a)は予測地平上で(1)を一次オイラー前進法で離散化して得られる車両モデルのダイナミクス、(8d)~(8f)は衝突回避制約である。(8g)は、物理世界における実現可能性、安定性、快適性を保証するための機械的制約である。AVと駐車場の場合

$$\begin{bmatrix} \Phi_{min} \\ v_{min} \\ \delta_{min} \\ a_{min} \end{bmatrix} \leq \begin{bmatrix} \phi_{k+t|t} \\ v_{k+t|t} \\ \delta_{k+t|t} \\ a_{k+t|t} \end{bmatrix} \leq \begin{bmatrix} \Phi_{max} \\ v_{max} \\ \delta_{max} \\ a_{max} \end{bmatrix}, \forall k=0 \dots N \quad (9)$$

コスト関数において、 J_1 はステージコスト、 J_2 はターミナルコストであり、以下の通りである。

$$J_1(z_{t+k|t}, u_{t+k|t}) = \|z_{t+k|t} - z_{t+k|t}^{high-level}\|_Q^2 + \|u_{t+k|t}\|_R^2 + \|u_{t+k+1|t} - u_{t+k|t}\|_\Delta^2$$

$$J_2(z_{t+N|t}) = \|z_{t+N|t} - z_{t+N|t}^{high-level}\|_{Q_N}^2 \quad (10)$$

表1: パラメータ設定

Parameter	Description	Value
L	length of the AV	4.7 m
W	width of the AV	2 m
Φ	the limit of yaw rate	0.7 rad
v	velocity at the center of rear	$[-1, 2]$ m/s
δ	mechanical steer limit	0.6 rad
a	acceleration bound	$[-1, 1]$ m/s ²
N	prediction horizon	5

表2: HA*とSHA*の計算時間

	min	max	average
HA*	0.8774 s	84.6162 s	9.6519 s
SHA*	0.8474 s	12.7495 s	2.1445 s

4 シミュレーション結果

本節では、パラレルパーキングのシミュレーション結果を示す。シミュレーションはMATLAB 2021環境で行い、CPUクロック2.7GHz、16GB RAMのIntel(R) Core(TM) i7-6820HKを搭載したMSI Titanで行った。高レベルのプログラミング定式化言語はCasADi [11]、低レベルはYALMIPである。興味のある読者は、[12]と[13]を参照されたい。(8)のNMPCは一般ソルバーIPOPTによって解かれる。シミュレーションの基本パラメータ設定を表1に示す。

SHA*法の改良を評価するために、148種類の初期駐車位置をテストした。HA*と比較した結果を表2に示す。xは-18から18の範囲、yは5から8の範囲で、それぞれ1ずつ増加する。

結果を図5に示す。HA*とSHA*の計算時間が短い場合がある。しかし、ほとんどの場合、我々のSHA*は従来のHA*よりも時間が短く、特に従来のHA*の計算時間が異常に長くなる場合がある。HA*とSHA*の性能は、いくつかのペナルティパラメータを調整することで改善できるが、時間浪費の作業であり、固定パラメータはすべての初期駐車位置に対してロバスト性に欠けることに注意すべきである。一方、本手法はパラメータチューニングを必要とせず、表2に示すように良好な性能を有している。

We take $z_{initial} = [-18, 6, 0, 0]$ as a case to test our hier-

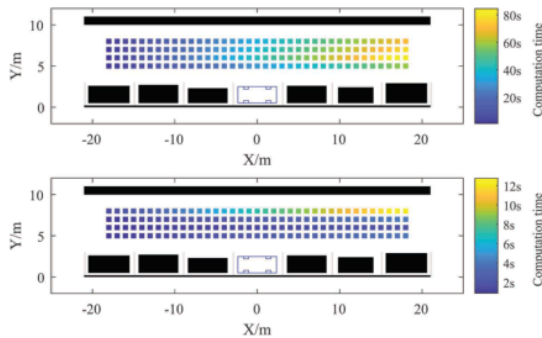


図5: ハイブリッドA*とSHA*の初期駐車形態の違いによる計算時間

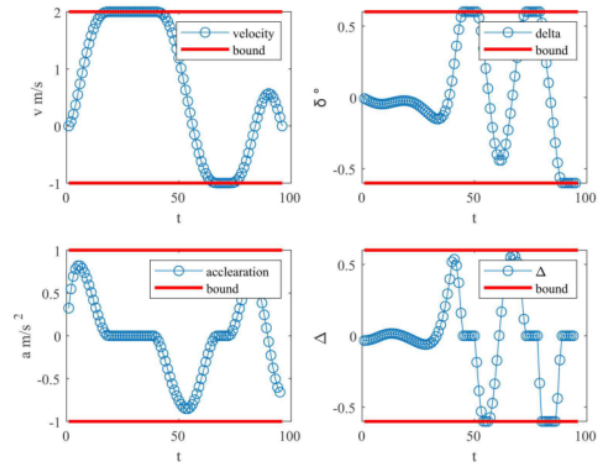


図6: ハイレベルな状態から見た状態の重要な値

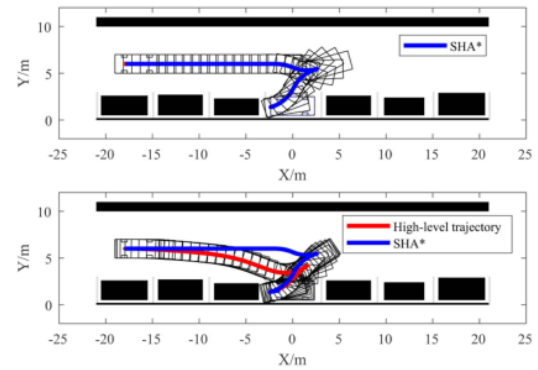


図7: 高レベルモジュールが計画する軌道

アーキカルモーションプランニングシステム高レベルSHA*と時間ベース最適化の結果を図7に示す。高レベルではD0は無視され、地図と障害物の情報は既知であるため、高レベルではオフラインで計算できることがわかる。

いくつかの重要な状態の値を図6に示す。上位層は動的な障害物を考慮する必要がなく、シナリオが単純であるため、上位層の制約は下位モジュールの制約よりも厳しくなる。動的な障害物を考慮する場合、より緩い制約を使用することで、より操作の余地を与える傾向がある。解くことができる状況である。低レベルでは、非線形MPCベースのスキームが、動的環境におけるリアルタイムの駐車操作を実行するために実装されている。駐車場シナリオを図8に示す。動的障害物は、水平方向から左水平方向へ、1m/sの一定速度で移動している。D0の状態は $z = [0, 6.5, -\pi, 0]$ である。D0は図8の左側に直進するように設定されている。AVは左から右へ駐車する過程でD0を回避しなければならない。低レベルでは、赤の基準は図9に示すような衝突につながる。高レベルモジュールで計算された赤の基準軌道を追跡している間、D0との衝突を避けるために、AVをある程度下方に移動させていることがわかる。

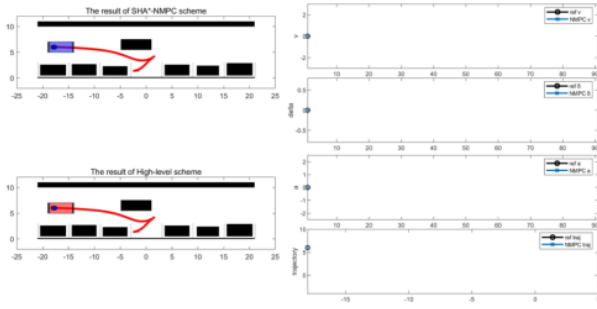


図8:自律駐車シナリオの構成

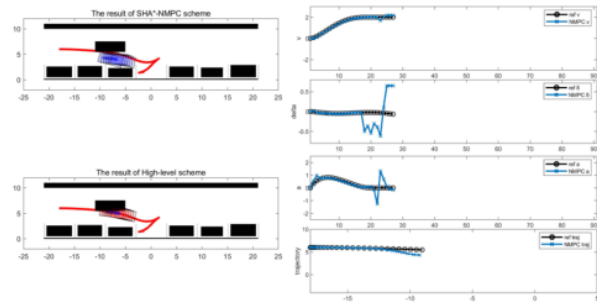


図9:衝突のない軌道の生成について

この結果は、動的な障害物を避けるために基準値からずれたステアリング角 δ の変化にも見ることができる。これとは逆に、赤いAVの左下隅には衝突が発生した領域が表示されている。

最終的な結果を図10に示す。このグラフから、障害物回避の過程では、ステアリングの旋回角度が主な調整条件であり、ステアリング角度と角度変化量が制約されるため、障害物回避に対応するためには加速度の変化も必要であることがわかる。障害物回避後、AVは素早く基準軌道をたどり、最終的に駐車を実現した。

(10)のパラメータの選択は、衝突回避の動作に影響を与える可能性がある。具体的なパラメータは表3に示す通りである。平行移動行動では、従うべき角度と水平座標が多く、垂直座標と速度が二次的な重要性を持っていると考えられる。自律走行駐車システムが間違っただけで計画されている場合、それは適格な駐車行動ではない可能性が高い。

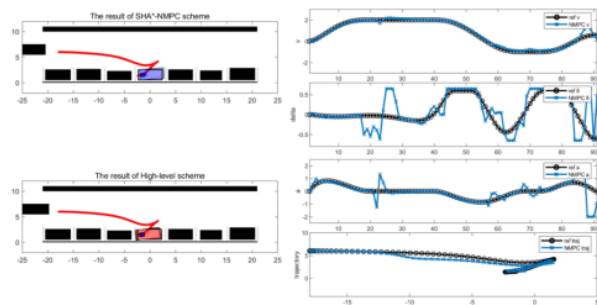


図10:駐車操作の最終段階

表3:NMPCウェイトマトリックスパラメータ設定

Parameter	Description	Value
Q	stage weight of 4 states	$\text{diag}[4, 1, 2, 5]$
R	weight of 2 control states	$\text{diag}[0.01, 0.01]$
Δ	入力の度合いの重み	$\text{diag}[0.1, 0.1]$
Q_N	terminal weight	$\text{diag}[8, 2, 4, 10]$

なお、本論文では動的障害物の不確かさや確率的誤差は考慮されておらず、今後の研究課題として検討する予定である。

5 CONCLUSION

本論文では、AVの自律駐車操縦のために設計された新しい階層的フレームワークであるSHA*-NMPCを紹介した。このフレームワークにおけるSHA*アルゴリズムの有効性と頑健性は、148の多様な初期駐車構成によって検証された。さらに、提案したフレームワークの実用性を、並列駐車の数値シミュレーションにより実証した。包括的な結果と分析により、実装されたシステムの強力な性能が実証された。

However, there are limitations to our approach. The primary drawback lies in the computational complexity of the nonlinear optimization problem formulated in the low-level control scheme. While we employ warm starting techniques to accelerate the optimizer's convergence, challenges arise when incorporating more intricate Dynamic Obstacles (DOs). For example, when faced with high-speed DOs, our system may necessitate advanced strategies such as early yielding or circumvention, which could require extending the prediction horizon and consequently increasing computational time. Furthermore, although duality theory can effectively handle non-smooth problems, it introduces additional variables, complicating the optimization process. 今後の課題として、本システムの安全性向上を目指す。探索の1つの道は、現在の予測地平よりも早く制約を生成できるインテリジェントなアルゴリズムの開発である。さらに、確率的最適制御戦略の採用は、不確実性を扱うためのロバストなフレームワークを提供する可能性がある。線形化技術は、非線形最適化問題を二次最適化問題として再定式化することにより、計算負荷を大幅に削減することも期待される。

REFERENCES

- [1] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.
- [2] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2015.
- [3] 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.
- [4] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceed-*

2000 ICRA. ミレニアム会議. IEEEロボティクスとオートメーション国際会議. シンポジウム講演論文集 (Cat. No. 00CH37065), vol. 2. IEEE, 2000, pp. 995–1001.

- [5] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, “Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 2997–3004.
- [6] A. Perez, R. Platt, G. Konidaris, L. Kaelbling, and T. Lozano-Perez, “Lqr-rrt*: Optimal sampling-based motion planning with automatically derived extension heuristics,” in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2537–2542.
- [7] X. Zhang, A. Liniger, and F. Borrelli, “Optimization-based collision avoidance,” *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 972–983, 2020.
- [8] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [9] R. Soloperto, J. Köhler, F. Allgöwer, and M. A. Müller, “Collision avoidance for uncertain nonlinear systems with moving obstacles using robust model predictive control,” in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 811–817.
- [10] T. Brüdigam, M. Olbrich, D. Wollherr, and M. Leibold, “Stochastic model predictive control with a safety guarantee for automated driving,” *IEEE Transactions on Intelligent Vehicles*, 2021.
- [11] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi – A software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.
- [12] L. T. Biegler, *Nonlinear programming: concepts, algorithms, and applications to chemical processes*. SIAM, 2010.
- [13] J. Lofberg, “Yalmip: A toolbox for modeling and optimization in matlab,” in *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*. IEEE, 2004, pp. 284–289.