

自律駐車への応用を目的とした2値最適動作計画(BOMP)モデル

Shenglei Shi, Youlun Xiong, Jiankui Chen and Caihua Xiong, *Member, IEEE*

概要-本論文では、自律駐車のための2値最適動作計画(BOMP)モデルを提示する。BOMPモデルは、モーションプランニングを最適制御問題として扱い、上層は車両の非線形ダイナミクス用に設計され、下層はジオメトリ衝突のない制約用に設計されている。BOMPモデルの大きな特徴は、下位レベルが上位レベルの問題の制約となる線形計画問題であることである。すなわち、最適制御問題には制約条件として組み込まれた最適化問題が含まれている。従来の最適制御法では、BOMP問題を直接解くことができない。そこで、修正近似KarushKuhn-Tucker理論を適用して、一般的な非線形最適制御問題を生成する。そして、擬似スペクトル最適制御法により、変換された問題を解く。特に、下層は凸多面体間の距離関数として働く J_2 -関数である。多面体は球体や楕円体よりも高精度に車両を近似することができる。また、変数数と時間複雑度を低減するために、修正 J_2 -関数(MJ)とアクティブポイントに基づく修正 J_2 -関数(APMJ)を提案する。その結果、動的実現可能性と無衝突性に関する自律駐車のための反復2段BOMPアルゴリズムが提案された。初期段階ではMJ関数を用いて、衝突のない近似最適軌道とアクティブポイントを求め、最終段階ではAPMJ関数を用いて最適軌道を求める。Turtlebot3でのシミュレーション結果と実験により、BOMPモデルが検証され、面積基準に基づく衝突回避法と比較して、計算速度がほぼ2桁向上することが実証された。

索引用語-自律駐車、最適制御、バイレベル最適動作計画(BOMP)、 J_2 -関数。

I. INTRODUCTION

自律走行車のためのリアルタイム無衝突運動計画・制御は、かなりの量の注目を集めており、ロボット工学の文献[1]~[26]と多くの研究手法を共有している。一般的に、自律駐車は、特に大都市では重要な操作である。分離法[3]~[6]は駐車問題に対する一般的なアプローチであり、車両経路計画と経路追跡は別々に扱われる。直接学習[7]は、制御入力と駐車軌跡の間のマッピング関係を学習する、斬新でシンプルなアプローチである。

本研究は、中国国家科学基金会の助成金51327801の支援を受けた。

S. Shi, Y. Xiong, J. Chen and C. Xiong are with the State Key Laboratory of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, 1037 Luoyu Road, Wuhan, 430074, P.R. China (e-mail: shshlei@hust.edu.cn; famt@hust.edu.cn; Chenjk@hust.edu.cn; chxiong@hust.edu.cn).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier ...

しかし、複雑な高精度駐車問題に対しては、車両の運動計画と制御を統一的な最適制御問題として扱うという、複合的なアプローチ[22], [23]がより効果的である。本論文では、駐車問題を複合最適制御問題として扱う。

モーションプランニングアプローチは、グラフ探索法[8]、[9]、人工ポテンシャルフィールド法[10]、[11]、サンプリングベース法[12]~[16]、非線形最適化法[17]~[26]に分類できる。グラフ探索法では、まず環境やロボットの状態空間の系統的な離散化を適用してグラフを構築し、次に情報量の多いヒューリスティックを用いて探索を誘導し、連結された実現可能な経路や軌道を生成する。離散化は多数の離散セルに悩まされるため、実用的なアプリケーションは低次元空間に限定される。人工ポテンシャル場法は、静的または動的な環境において大きな成果を上げている。さらに、自動駐車への応用は、非ホロノミックシステムに対する能力を示している。しかし、人工ポテンシャル場法は局所最適経路にとらわれやすく、局所計画特性によりマルチマヌーバ駐車問題を扱うことができない。サンプリングに基づく手法は、ロボット構成空間またはロボット状態空間のサンプルをランダムに生成し、ローカルプランニング手法を用いて、生成されたグラフまたはツリーに新しいサンプルを接続しようとするものである。サンプリングに基づく方法は、障害物のない構成空間を明示的に表現しないため、計算効率が高く、高次元空間に適している。しかし、いくつかの制約や目的に関する滑らかさの問題や解決策の質については、真剣に検討する必要がある[20]。解の質の問題を考慮し、非線形最適化手法はロボットダイナミクスを正しく扱うことができ、タスク仕様を満たすために、いくつかの関連する目的と制約を構築することができる。とはいえ、障害物が存在するため、実行可能集合は不連続であり、非線形最適化問題は非凸であるため、問題の難易度と計算の複雑さが増す。

最適制御[27]~[29]は、ロボットの高品質な軌道を生成するための注目すべき手法であり、実用化において大きな成功を収めている。ロボットダイナミクスやその他の軌道制約をコンパクトかつ統一的な形で考慮し、事前に定義された最適化目標を扱うことができる。ロボットの運動計画における最適制御法の適用には、ロボットダイナミクス制約とジオメトリ無衝突制約の2つの本質的な制約が含まれる。しかし、[21]の円や[22]の矩形のような初等的な形状に限定して適用している。本研究では、矩形を仮定した複雑なシナリオに最適制御法を適用できるようにし、この問題を高速に解く。

A. Problem Analysis

複雑な運動計画問題において最適制御法を適用するという課題は、形状の衝突のない制約をいかに効果的に表現するかということから生じる。[21]の円の点間距離と[22]の矩形の面積基準は、どちらも単純な衝突回避法である。しかし、円の境界体積近似は、複雑で高精度なシナリオで運動を実現するには保守的すぎる。また、矩形の面積基準は非線形で冗長であるため、最適制御問題を解くことは困難である。また、3次元環境における高精度な運動計画問題に対する無衝突制約の構築はまだ困難である。本論文では、 $\frac{1}{2}$ -関数[30], [31]がこれらの課題を解決し、自律駐車のための高速な高精度運動計画を達成することを示す。この $\frac{1}{2}$ -関数は、任意の次元空間における凸多面体間の衝突チェックのための線形計画法であり、 $\frac{1}{2} > 0$ が衝突がないことを示す距離関数として振る舞う。 δ は安全距離、 $\frac{1}{2}$ はロボット軌道の関数である)を全体最適制御問題の制約条件として扱い、特別な $\frac{1}{2}$ -関数に基づく2値最適運動計画(BOMP)モデルを得る。

BOMPモデルは、上部最適制御問題の制約の中で最適化問題を含んでいる。さらに、下側の最適化問題は、ロボットの状態の連続的な軌跡に依存し、無限次元にもなる。我々の知る限り、この特殊なモデルは文献に登場していないが、このモデルの複雑さは他のいくつかの問題から間接的に見ることができる。[32]では、相対的に単純な2値最適制御問題が検討されている。上段は常微分方程式、制御制約、初期状態制約、最終状態制約に関する最適制御問題であり、下段の問題は物理系の最終状態にのみ依存し、有限次元である。この場合、非平滑解析、バナッハ空間での最適化、バイレベル最適化 [33], [34] を用いて、必要な線形化ポントリャーギン型最適化条件を導出する。非凸、非微分、あるいは切断される可能性があるため、階層的2値最適化問題は本質的に困難である。最も単純な線形-線形バイレベル最適化問題であっても、強いNP困難性[35]が証明され、単に最適性を評価することもNP困難性[36]であることがわかった。

BOMPモデル、バイレベル問題、最適制御問題から2つの重要な課題が分類される。最適制御は、数学と工学においてかなりの発展を遂げている。特に、擬似スペクトル最適制御(PSOC)法[29]は、大域的に微分方程式を満たし、陰的ルンゲクッタ法によって積分を扱うため、高次精度、高次安定性、指数関数的収束速度を達成することができる。したがって、PSOC法が選択される。二階層最適化問題では、低階層最適化問題のKarush-Kuhn-Tucker (KKT)再定式化が、伝統的な単一階層最適化問題を構築するために常に使用される[37]。BOMP問題は最適化問題ではなく最適制御問題であるため、KKT再定式化は一見BOMPモデルには適用できない。

しかし、PSOCアルゴリズムは変数全体を離散化し、多項式として近似するため、実際には大規模なスパース非線形最適化問題(NLP)を解くことができる。この場合、KKT再定式化は離散化NLPに直接使うことができる。さらに、BOMPモデルの下位レベルでは微分方程式は発生しないので、連続KKT再定式化(ラグランジュ乗数も時間の軌跡関数である)の概念を明確に利用することができる。

しかし、KKT条件における非凸性により、上位レベルの問題も凸であるため、変換された問題はまだ解くのが難しい。その上、相補制約は本質的に組合せ的であり、KKT点の小さな近傍におけるKKT条件の違反の程度を非平滑にする[38]。したがって、最適値の近傍での収束のpropertyは悪く、現在の値が最適値に十分近いかどうかを判断するための効率的な情報を提供することはできない。Dutta [38]は、KKT条件が緩和され、近傍での違反が十分に滑らかであるという修正近似KKT (MAKKT)理論を提案した。KKT条件は緩和されるので、緩和係数を減少させ、最適値に近づくように反復戦略が用いられる。そこで、本論文では、MAKKT理論と反復収束戦略を選択した。

B. 貢献と組織化

本論文では、上段をロボットの非線形ダイナミクス用に設計された最適制御問題、下段を形状衝突のない制約のための $\frac{1}{2}$ -関数線形計画法とする一般的なBOMPモデルを紹介する(セクションII)。MAKKT理論を用いてモデルを従来の最適制御問題に単純化し、PSOC法を用いて変換問題を解く(セクションIV)。また、セクションIIIでは、時間複雑性を軽減するために、修正 $\frac{1}{2}$ -関数(MJ)とアクティブポイントベースの修正 $\frac{1}{2}$ -関数(APMJ)を導出する。MJ関数とAPMJ関数を自然に組み合わせる反復収束戦略を提案し、2段階のBOMPアルゴリズムを提案する。本論文のハイライトは以下の通りである：

- 1) BOMPモデルは、最適制御の柔軟性と線形計画法の簡便性の両方の利点を兼ね備えており、複雑な運動計画問題に対して最適制御を高速かつ高精度な手法とすることができる。
- 2) 2段階アルゴリズムでは、初期段階でMJ関数を用いて衝突のない近似最適軌道を生成し、アクティブな点を選び出す。最終段階でAPMJ関数を用いて正確な最適軌道を見つける。
- 3) BOMPアルゴリズムは、反復収束戦略とMJ関数とAPMJ関数の単純さによって利益を得る。自律駐車におけるシミュレーションでは、面積基準に基づく方法(セクションV)と比較して、計算速度がほぼ2桁向上することが実証された。

最後に、セクションVIで結論を述べ、この分野で研究されるべき問題点を指摘する。

III. 衝突回避

ロボット衝突のない運動計画アルゴリズムの性能は、ロボットと環境との間で使用される形状衝突回避方法に大きく依存する。実世界では、ロボットや環境の形状は常に洗練されており、衝突回避問題を単純化するために近似技術が常に利用されている。3次元環境では多面体近似や球近似、2次元環境では多角形近似や円近似が従来手法である。本節では、球(または円)間の点-点距離制約、凸多角形間の面積基準制約、 J_2 -関数を最初にまとめる。この J_2 関数は、任意の次元空間における凸多面体間の衝突回避法の一つを表す。 J_2 -関数における人工変数の詳細な幾何学的解釈を示す。次に、幾何学的性質に従って、MJ関数とAPMJ関数を提案する。

A. 基本的な衝突回避制約

ロボットBを半径 r_b の球または円で近似し、障害物Aを半径 r_a の球または円で近似するとする。近似オブジェクトの中心は p_b と p_a である。そして、AとBの間の幾何学的無衝突制約は次のように近似される：

$$d(p_a, p_b) - r_a - r_b \geq \delta > 0 \quad (6)$$

ここで $d(p_a, p_b)$ は2点間のユークリッド距離を表す。通常、ロボットと障害物は、近似精度を上げるために、いくつかの重なり合った球や円で近似されるが、この場合、ジオメトリ衝突のない制約はまだ簡単である。平面ロボットBと平面障害物Aが、それぞれ n_b 個の頂点と n_a 個の頂点を持つ凸多角形で近似されているとする。ポリゴンの頂点座標はそれぞれ $\{p_i\}$, $i = 1, 2, \dots, n_b$, $\{p_j\}$, $j = 1, 2, \dots, n_a$ である。そして、面積基準は、AとBの間の幾何学的な無衝突制約を記述することができます：

$$\begin{aligned} \sum_{j=1}^{n_a-1} S_{\Delta p_i p_j p_{j+1}} + S_{\Delta p_i p_{n_a} p_1} - S_A &\geq \delta > 0, \quad i = 1, 2, \dots, n_b \\ \sum_{i=1}^{n_b-1} S_{\Delta p_j p_i p_{i+1}} + S_{\Delta p_j p_{n_b} p_1} - S_B &\geq \delta > 0, \quad j = 1, 2, \dots, n_a \end{aligned} \quad (7)$$

ここで、 S_{Δ} は三角形の面積、 S_A と S_B は2つの近似多角形の面積である。さらに、3次元環境における多面体については、体積基準により衝突回避制約を導出することができる。

このように、多角形の組に対して、(7)には $n_a + n_b$ 個の非線形制約があり、この数は通常、円近似法よりも大きい。しかし、理論的には、複雑な形状は、球や円近似法よりも少ない近似オブジェクトで、複数の多面体や多角形によって、任意の精度と低い重複率で組み込むことができる。また、障害物が長く薄いシナリオで計画と制御を実現するためには、多面体または多角形近似がより効率的である。

したがって、凸多面体または多角形近似法は、複雑で高精度なロボットの運動計画問題において重要な役割を果たすことになる。

B. J_2 -関数の定義と基本特性

2つの点集合 $A = \{a_i\}$, $i = 1, 2, \dots, n_1$ と $B = \{b_j\}$, $j = 1, 2, \dots, n_2$ の凸包で表現される2つの凸多面体、すなわち $\text{conv}A$ と $\text{conv}B$ がそれぞれ与えられたとする。ここで、 $a_i \in \mathbb{R}^m$ と $b_j \in \mathbb{R}^m$ は点座標、 m は次元数、 n_1 と n_2 は点数である。次に、 J_2 -関数は2つのオブジェクト $\text{conv}A$ と $\text{conv}B$ が衝突しているかどうかをチェックする：

$$J_2(A, B) = \min \sum_{k=1}^{m+2} z_k \quad (8)$$

$$\text{s.t. } Ax - By + z = 0, \quad (8a)$$

$$\sum_{i=1}^{n_1} x_i + z_{m+1} = 1, \quad \sum_{j=1}^{n_2} y_j + z_{m+2} = 1, \quad (8b)$$

$$x \geq 0, y \geq 0, z \geq 0 \quad (8c)$$

$$z_{m+1} \geq 0, z_{m+2} \geq 0 \quad (8d)$$

ここで、 $x = [x_1 \ x_2 \ \dots \ x_{n_1}]^T$ 、 $y = [y_1 \ y_2 \ \dots \ y_{n_2}]^T$ 、 $z = [z_1 \ z_2 \ \dots \ z_{m+2}]^T$ 。ベクトル z とスカラー z_{m+1} と z_{m+2} を人工変数と呼ぶ。制約条件(8a)において、記号AとBは実際には点座標からなる行列、すなわち $A = [a_1 \ a_2 \ \dots \ a_{n_1}]$ 、 $B = [b_1 \ b_2 \ \dots \ b_{n_2}]$ であるため、本稿では点集合と行列の表記を明示的に区別せず、文脈を通して真の意味を直接かつ容易に見ることができる。また、ある物体の点集合を参照するとき、その物体を特徴付けるために、その物体の凸包が実際に使われる。本論文では、ベクトルの不等式を要素ごとに理解する。

この J_2 -関数は次のような基本性質を持つ：

- (a) the feasible region (8a)-(8d) is nonempty;
- (b) the range of the optimal value is $0 \leq J_2 \leq 2$;
- (c) $J_2 > 0$ is equivalent to $\text{conv}A \cap \text{conv}B = \emptyset$, i.e., collision free between the objects;
- (d) it provides a concept of pseudodistance between convex polyhedrons and is pseudomonotonic with respect to the Euclidean distance.

C. J_2 -関数の幾何学的解釈

$\sigma_A = \sum_{i=1}^{n_1} x_i$, $\sigma_B = \sum_{j=1}^{n_2} y_j$ とすると、 $0 \leq \sigma_A, \sigma_B \leq 1$ であり、図2(a)に示すように、それぞれ原点に対する物体A、Bの分数縮小を表す。このようにして、人工変数 z_{m+1} と z_{m+2} の意味を直接見ることができる：

$$z_{m+1} = 1 - \sigma_A, \quad z_{m+2} = 1 - \sigma_B \quad (9)$$

Bの定義から、 $By = \sigma_B p_B$ ($p_B \in \text{conv}B$) が得られる。ここで、人工ベクトル z の理解を簡単にするために、 J_2 -関数において点集合Aと点bを考える。まず、 $\sigma_A = \sigma_B = 1$ (または $z_{m+1} = z_{m+2} = 0$)と仮定し、ベクトル z の意味に注目する。 $z \geq 0$ であるから、制約条件(8a)から、基本的な実行可能領域は $b \in \text{conv}A + \mathbb{R}_+^m$ であり、 \mathbb{R}_+^m は非負の正射を表す。

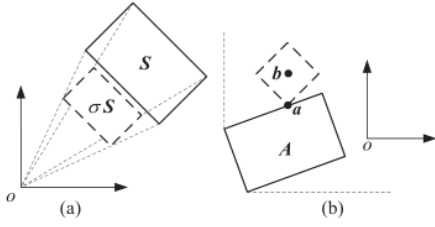


図2. 人工変数 z , z_{m+1} , z_{m+2} の幾何学的意味. (a) 凸多角形 S の原点に対する縮小モデル、 σ は分数縮小。 z_{m+1} と z_{m+2} は物体 A と B の分数縮小に関する。(b) 点と凸多角形との距離モデル J_1 、 z は点 a から b へのベクトルを表す。

次に、 $\text{conv}A + = \text{conv}A + R_+^m$ とし、図2(b)に示すように、 $b \in \text{conv}A +$ 、 $b \cap \text{conv}A = \emptyset$ と仮定する(例外については後述する)。関数 J_2 の目的と組み合わせると、最適解 z^* の J_1 ノルムは最適値 $J_2^*(A, b)$ に等しい、つまり人工ベクトル z は2つの凸多面体間の J_1 距離を測定していることがわかる。そして、次の2つの幾何学的問題(10)と(11)が得られる。

点集合 A 、点 b を考え、 $\sigma A = \sigma B = 1$ 、 $b \in \text{conv}A +$ とすると、 J_2 -関数は次の幾何学的問題と等価である：

$$J_2(A, b) = \min_{S_c \cap \text{conv}A \neq \emptyset, c \geq 0} c \quad (10)$$

ここで、 S_c は点 b に関する J_1 距離関数のレベル集合を表す。

$$S_c = \{p \in \mathbb{R}^m \mid c = \|p - b\|_1\}$$

点集合 A と B を考え、 $\sigma A = \sigma B = 1$ 、 $\text{conv}B \cap \text{conv}A + \neq \Delta$ とすると、 J_2 -関数は次の幾何学的問題と等価である：

$$J_2(A, B) = \min_{S_c \cap \text{conv}A \neq \emptyset, c \geq 0} c \quad (11)$$

ここで、 S_c は物体 B に関する J_1 距離関数のレベル集合を表す。

$$S_c = \{p \in \mathbb{R}^m \mid c = \min_{p \in \text{conv}A, b \in \text{conv}B} \|p - b\|_1\}$$

z , z_{m+1} , z_{m+2} の幾何学的な意味が明確であるため、次に $b \sim c \in \text{conv}A +$ のとき J_2 -関数がどのように働くかを議論する。このケースは、図に示すように3つのサブケースに分類することができる。3(a)-(c): (a) $0 \in \text{int conv}A +$, $b \sim \text{conv}A +$; (b) $0 \in \text{bd conv}A +$, $b \sim \text{conv}A +$; および (c) $0 \sim \text{conv}A +$, $b \sim \text{conv}A +$ 。ここで、 $\text{int conv}A +$ と $\text{bd conv}A +$ はそれぞれ $\text{conv}A +$ の内部と境界を表す。サブケース(a)において、線形計画法が解けるような $\exists B > 0$, $\sigma B b \in \text{conv}A +$ 。サブケース(b)では、制約を実現可能にするために σB はゼロに等しくなければならない。サブケース(c)では、 $a^* = 0$, $b^* = 0$, $J_2 = 2$ となり、直感的に最適解が得られる。最後に、最適値 $J_2 = 0$ は $b \in \text{conv}A$ または $\text{conv}A \cap \text{conv}B \neq \Delta$ のときのみ発生することを指摘する。

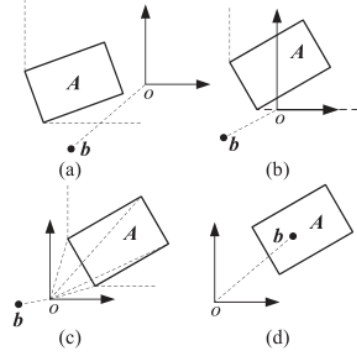


図3. 平面上の凸多角形 A に関する点 b のいくつかの特別な関係。

D. 修正 J_2 -関数

人工変数の幾何学的な意味は、 J_2 -関数が σA , σB の縮約を最大化し、 $\sigma A \text{ conv}A$ と $\sigma B \text{ conv}B$ の距離を最小化し、制約 $\sigma_B \text{ conv}B \cap (\sigma_A \text{ conv}A + R_+^m) \neq \Delta$ を達成することを示す。本節では、 J_2 -関数が σB を最大化するだけで実現できることを示し、幾何学的問題(12)を得る。

点集合 A と B が与えられ、 $0 \in \text{int conv}A$ とすると、 J_2 -関数は幾何学的問題と等価である：

$$J_2(A, B) = \min_{\sigma_B \text{ conv}B \cap \text{conv}A \neq \emptyset, 0 \leq \sigma_B \leq 1} 1 - \sigma_B \quad (12)$$

$0 \in \text{int conv}A$ と $b \sim \text{conv}A +$ のとき、問題(12)の正しさはセクション III-C の幾何学的解析から直接見ることができる。したがって、 $0 \in \text{int conv}A$, $b \in \text{conv}A +$ とし、図4(a)に示すような矩形 A 、点 b を用いて、証明を直感的に説明する。 $0 \in \text{int conv}A$, $\forall b \in \text{conv}A +$, $\exists 0 \leq \sigma_B \leq 1$ なので、 $\sigma_B b \cap \text{conv}A \neq \emptyset$ 。そして、 $b \sim \text{conv}A$ ならば、 σ_B がゼロに近づくと、 $\sigma_B b$ は最初 $\text{conv}A$ の境界と交差する。また、 σ_A が小さくなれば、 σ_B が小さくなり、 $\sigma_B b$ が $\sigma_A \text{ conv}A$ の境界と交差するようになる。その結果、最適値 J_2 は1に等しい(点 b と物体 A の距離 J_1 の観点からも同じ結果が得られる)。ここで、シュリンク σ_B と J_1 距離だけが J_2 -関数の目的値に影響する。 $b \sim \text{conv}A$ とすると、 J_1 距離は σ_B がゼロに近づくとつれて小さくなり、 J_2 -関数は両者のバランスをとる。 J_2 -関数におけるこれら2つの因子の有意性を比較するために、これらを独立に考える。ここで、 σ_B^* は $\max \sigma_B b \cap \text{conv}A \neq \emptyset$ の最適解を表し、 $a_1 = \sigma_B^* b$ とする。(10)の最適値を c' 、 $a_2 = S_c' \cap \text{conv}A$, S_c' と交差する $\text{conv}A$ からの表面の法線を n 、線 ba_2 と法線が n の部分空間の交点を a_3 とする。 a_1 , a_2 , a_3 と n の表記を図4(a)に示す：

$$c' = \|b - a_2\|, 1 - \sigma_B' = \frac{\|b - a_1\|}{\|b\|} = \frac{\|b - a_2\|}{\|b - a_3\|} < 1. \quad (13)$$

$1 - \sigma_B'$ よりも c' が大きく、 σ_B が J_2 -関数において主導的な役割を果たすことがわかる。したがって、 $\sigma_B^* = \sigma_B^* = 1 - J_2^*(A, b)$ となり、問題(12)が得られる。

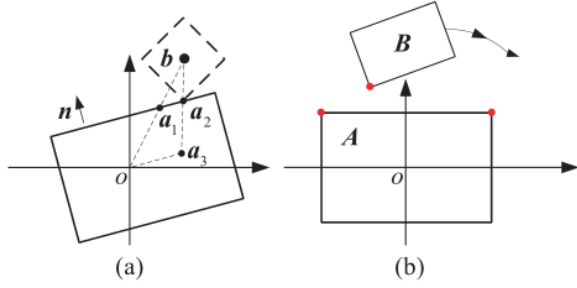


図4 (a) 縮小 σ Bと l_1 距離の有意比較。(b) アクティブポイント(赤い点)は、瞬間の衝突挙動を特徴付けることができる。

幾何学的問題(12)によれば、 $0 \in \text{int conv } A$ の場合、MJ線形計画法は次のようになる：

$$J_2(A, B) = \min 1 - \sum_{j=1}^{n_2} y_j \quad (14)$$

$$\text{s.t. } Ax - By = 0, \quad (14a)$$

$$\sum_{i=1}^{n_1} x_i = 1, \quad \sum_{j=1}^{n_2} y_j \leq 1, \quad (14b)$$

$$x \geq 0, y \geq 0 \quad (14c)$$

そして、そのコンパクトな形

$$J_2(A, B) = \min 1 + c^T p \quad (15)$$

$$\text{s.t. } Qp = b, \quad 1 + c^T p \geq 0, \quad p \geq 0$$

where

$$p = \begin{bmatrix} x \\ y \end{bmatrix}, \quad c = \begin{bmatrix} 0_{n_1 \times 1} \\ -1_{n_2 \times 1} \end{bmatrix}, \quad b = \begin{bmatrix} 0_{m \times 1} \\ 1 \end{bmatrix},$$

$$Q = \begin{bmatrix} A & -B \\ 1_{1 \times n_1} & 0_{1 \times n_2} \end{bmatrix}.$$

MJ関数は $m + 2$ 変数を削減し、計算速度を向上させる。最適値は $0 \leq l_2 \leq 1$ であり、 $l_2 = 1$ はBが無限大の場合に対応する。

E. アクティブポイントに基づく修正 J_2 -関数

実際、比較的移動する2つの多面体間の衝突は、常に面/頂点、頂点/面、辺/辺の最も近い成分間で発生する[39]ので、衝突回避は最も近い成分と衝突のない成分と等価である。この事実によれば、最も近い成分が決定できれば、衝突回避問題に対して、各物体の瞬時幾何学的表現を簡略化することができる。この小節では、与えられた2つの物体が衝突していないため、最も近い成分の概念が意味を持つという仮定を置く。図4(b)は、この仮定を固定化した2つの比較的動く平面多角形A、Bを示している。そして、瞬時に、衝突を避けるために、各ポリゴンを赤い点で特徴付けることができる。本論文では、これらの点を活性点と呼び、 J_2 -関数の最適解 x^* , y^* の非零要素に対応する：

$$A_a = \{a_i \mid x_i^* \neq 0\}, \quad B_a = \{b_j \mid y_j^* \neq 0\} \quad (16)$$

Using A_a and B_a , an APMJ function can be obtained:

$$\begin{cases} J_2(A_a, B_a) = 1 - \sum_{j=1}^{n_{a2}} y_j \\ A_a x - B_a y = 0, \quad \sum_{i=1}^{n_{a1}} x_i = 1, \\ x \geq 0, \quad y \geq 0 \end{cases} \quad (17)$$

ここで n_{a1} と n_{a2} は活点数である。

APMJ関数は最適化問題ではなく、単なる制約付き連立一次方程式であり、点集合 A_a と B_a のカーディナリティは非常に低いことに注意。したがって、APMJ関数は計算時間を劇的に短縮する。しかし、2つの物体が任意に与えられた場合、その間の活性点は透明ではない。また、ロボットの動作計画問題では、ロボットの動作軌跡はまずわからないが、アクティブポイントはおろか、ロボットの動作軌跡はわからない。その結果、APMJ関数は直線的に使用することができず、最初にアクティブポイントを選択するためにいくつかの技術を利用する必要がある。セクションIVでは、2段階のBOMPアルゴリズムを提案する。初期段階でのMJ関数は近似的な最適軌道を見つけ、軌道に沿った活性点を選び出してAPMJ関数に入力し、最終段階のアルゴリズムを構築する。関連する理論、および2段階のアルゴリズムの詳細については、以下のセクションで説明する。

IV. BOMPモデルの解法

(15) A. J_2 -関数に基づくBOMPモデル

凸多面体ロボットBが静的な凸多面体障害物Aの周りを初期状態から最終状態まで移動すると考える。そして、目的が最適な滑らかな無衝突軌道を生成することであると。 (c)と(d)の性質 J_2 -関数、新しい制約 $J_2(A, B) \geq \delta > 0$ 、MJ関数が制約(4d)と(4e)を置き換えると、厳密な J_2 -関数に基づくBOMPモデルが得られる：

$$\min \beta g(x_i, x_f, t_f) + (1 - \beta) \int_0^{t_f} L(x, u) dt \quad (18)$$

$$\text{s.t. } \dot{x} = f(x, u) \quad (18a)$$

$$x(0) = x_i, \quad x(t_f) = x_f \quad (18b)$$

$$x \in X, \quad u \in U \quad (18c)$$

$$J_2(A, B) \geq \delta > 0 \quad (18d)$$

$$J_2(A, B) = \min 1 + c^T p \quad (18e)$$

$$\text{s.t. } Qp = b, \quad 1 + c^T p \geq 0, \quad p \geq 0.$$

ここで、 A と B は障害物座標系で記述された点集合である。したがって、 A は定数であり、 B はロボットの状態 x の関数である。より正確には、 B , Q , p は時間 t の軌跡関数である。したがって、制約条件 $J_2(A, B) \geq \delta > 0$ はロボットの運動が衝突しないことを示す。

B. Model Solution

BOMPモデルの問題点を考慮すると、全体的な2レベル最適制御問題を従来の1レベル最適制御問題に還元することは有用である。線形計画法[40]の最適条件に従い、制約条件(18d)に気づくと、制約条件 $1 + c^T p \geq 0$ はアクティブではない。

つまり、下位最適化問題(18e)では、制約 $p \geq 0$ に対応するラグランジュ乗数 $\lambda \in \mathbb{R}^{n+1}$ $\lambda \geq 0$ と、制約 $Qp = b$ に対応する乗数 $v \in \mathbb{R}^{m+1}$ が存在する。 λ と v が時間 t の軌跡関数である場合、下問題の連続MAKKT形式は次のようになる:

$$\begin{cases} Qp = b, & p \geq 0 \\ \|c - \lambda + Q^T v\| \leq \sqrt{\epsilon}, & \lambda^T p \leq \epsilon, & \lambda \geq 0 \end{cases} \quad (19a)$$

$$(19b)$$

ここで、(19a)は実行可能条件、(19b)は平衡条件と相補条件、 $\epsilon > 0$ は近似緩和である。これらの制約を静める点 p を ϵ -MAKKT点(または ϵ 近似最適点)と呼び、 $\epsilon = 0$ のとき、実際にはKKT点(または最適点)であることに注意する。

BOMPモデルでは、下位最適化問題の最適値 $J_2(A, B)$ も制限されるため、最適値と近似最適値の関係も考慮する必要がある。すなわち、ある ϵ -MAKKT点では、目的値 $1 + c^T p$ は少なくとも $J_2(A, B)$ より大きい。その結果、(18)の変換された単一レベル最適運動計画問題は以下のように定式化できる:

$$\min \beta g(x_i, x_f, t_f) + (1 - \beta) \int_0^{t_f} L(x, u) dt \quad (20)$$

$$\text{s.t. } \dot{x} = f(x, u) \quad (20a)$$

$$x(0) = x_i, \quad x(t_f) = x_f \quad (20b)$$

$$x \in X, \quad u \in U \quad (20c)$$

$$1 + c^T p \geq \epsilon + \delta \quad (20d)$$

$$Qp = b, \quad p \geq 0, \quad \lambda \geq 0 \quad (20e)$$

$$\|c - \lambda + Q^T v\| \leq \sqrt{\epsilon}, \quad \lambda^T p \leq \epsilon \quad (20f)$$

MAKKTと厳密なKKT最適化条件の関係は以下の通りである: 点列 $\{p_k\}$ がマンガリア・フロビッツ制約条件も満たす点 p^* に収束し、各点 p_k が ϵ_k -MAKKT点であり、 $\epsilon_k \rightarrow 0$ $as^k \rightarrow \infty$ ならば、点 p^* はKKT点である。制約条件(18d)と(18e)を満たすロボット状態とは、その構成が自由空間の δ -interior内にあること、すなわち $q \in \text{int } \delta(C_{\text{free}})$ [12] を意味する。同様に、制約条件(20d)-(20f)を満たす状態 $x \in X$ を定義すると、その構成は $\epsilon - \delta$ -interior構成となる。そして、 C の $\epsilon - \delta$ -内部は $\epsilon - \text{int } \delta(C)$ と表され、全ての $\epsilon - \delta$ -interior配置の集合として定義される。MAKKTと厳密なKKT最適化条件の関係から、 $\epsilon - \text{int } \delta(C)$ は ϵ がゼロに収束するにつれて $\text{int } \delta(C_{\text{free}})$ に収束する(図5参照)。

したがって、問題(18)を解くためには、問題(20)に減少列 ϵ を適用し、近似最適軌道列を解いて正確な最適軌道に収束させる必要がある。が大きいとき、問題(20)は ϵ の最適軌道を解くのが簡単である。次に ϵ を減少させ、 ϵ がゼロに収束するまで新しい軌道を解くための初期推測として前の軌道を選ぶ。しかし、 ϵ がゼロに近づくにつれて、MAKKT条件における非凸性と組合せ性が強調され、問題が解きにくくなるか、収束しなくなる。

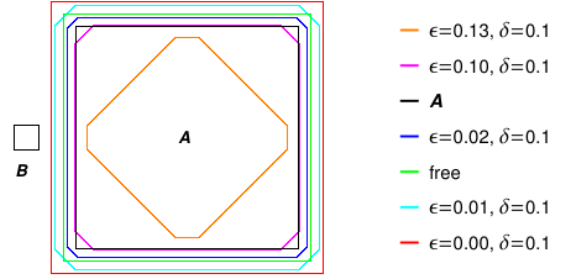


図5. 正方形Bは向きを固定して自由に平行移動できるため、その配置は中心点の座標と配置空間 C が平面であることで記述できる。正方形Aは障害物であり、座標系はその中心を原点とする。ラベル "A" は障害物、"free" は障害物のない配置空間 C_{free} の境界、" $\epsilon = 0.00, \delta = 0.1$ " は $\text{int } \delta(C_{\text{free}})$ 、他のラベルは $\epsilon - \text{int } \delta(C)$ を表す。このように、 $\epsilon - \text{int } \delta(C)$ は ϵ がゼロに収束すると $\text{int } \delta(C_{\text{free}})$ に収束する。

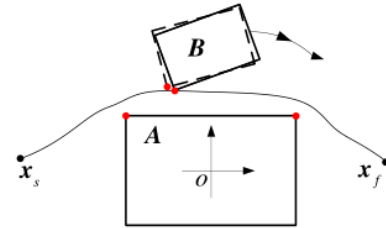


図6. APMJ関数が機能する理由。実線は物体Bが x_s から x_f に移動するための近似的な最適軌道であり、小さな変動は2つの物体間の活性点を変えない。

このような状況下では、新しい収束技術を求める必要がある。APMJ関数はこの問題を解決し、なぜうまくいくのかは以下の通りである。

ϵ がゼロに近づくとき、まず衝突のない近似最適軌道を解くことになり、図6に示すように、厳密最適解はその小さな変形となり、実線と破線のボックスはそれぞれ近似軌道と瞬間最適位置となる。このように、小さな変動は物体間の活性点を変えないので、近似軌道からの活性点は衝突挙動を正確に特徴付けることができる。この軌道に沿った活性点を選び、APMJ関数(17)を構成すると、次のような運動計画モデルが得られ、正確な最適軌道を解くことができる。

$$\begin{aligned} \min \quad & \beta g(x_i, x_f, t_f) + (1 - \beta) \int_0^{t_f} L(x, u) dt \\ \text{s.t.} \quad & \dot{x} = f(x, u) \\ & x(0) = x_i, \quad x(t_f) = x_f \\ & x \in X, \quad u \in U \\ & 1 + c^T p \geq \delta \\ & Qp = b, \quad p \geq 0 \end{aligned} \quad (21)$$

ここで、 Q, p, c の表記は(20)の表記と区別する必要がある。このモデルでは、

Q , p , c はカーディナリティが非常に低い活性点に対応する。その上、乗数 λ 、 v 、相補制約と平衡制約が消えるので、このモデルはより解きやすい。

C. BOMPアルゴリズム

以上の理論と技術を組み合わせて、ロボットの軌道生成と最適化ミッションのための2段階のBOMPアルゴリズムフレームワークを定式化する。アルゴリズムの詳細をアルゴリズム1に示す。ステップ1~10は初期段階、ステップ11は最終段階に相当する。初期段階では、MJ関数を用いて運動計画問題(20)を構成し、衝突のない近似最適軌道を生成し、軌道に沿った活性点を選択する。 ϵ -MAKKT点と正確なKKT点の関係から、初期段階では減少列 ϵ を採用し、一連の近似最適軌道を解く。最終段階では、APMJ関数に基づくモデル(21)を用いて最適な軌道を把握する。PSOC法は問題(20)と(21)の両方を解き、2つのステージの離散ノード数はそれぞれ15と30である。本論文の長さを制限するため、PSOC法の詳細は省略し、読者は参考文献[29]を参照することができる。離散大規模自然言語処理の解法には、バージョン3.12.9のIPOPT [41]アルゴリズムが使用されている。IPOPTアルゴリズムは、収束許容誤差 $1e$ 、MA86線形ソルバーなどの特殊なオプションで設計されており、アルゴリズムの早期終了を避けるために、許容できる終了条件は望ましいものよりも厳しい。IPOPTアルゴリズムのオプションを表Iに示す。最後に、BOMPアルゴリズムで使用される小さなトリックであるステップ9が、衝突回避制約の数を劇的に減らすことを指摘する。つまり、ロボットが障害物から非常に離れている場合、その瞬間に衝突回避制約を考慮する必要はない。小さなトリックは正しい臨界衝突回避制約を選び出し、この制約を本論文ではアクティブ制約と呼ぶ。

アルゴリズム1 BOMPアルゴリズム

入力: δ 、 ϵ 列 $\{0.1, 0.01, 0.001, \dots\}$ 、初期軌道推測 τ_0 、PSOC法における離散ノード数 K 、能動制約閾値 J_a 。

Output: the optimal trajectory τ^* .

- 1: Let $k = 0$, $K = 15$.
- 2: **loop**
- 3: Using ϵ_k to construct the problem (20).
- 4: Solve problem (20) by the PSOC method with the initial guess τ_k to get a new trajectory τ_{k+1} .
- 5: **if** the trajectory τ_{k+1} is collision free **then**
- 6: Let $\tau = \tau_{k+1}$ and break.
- 7: Let $k \leftarrow k + 1$.
- 8: Let $K = 30$ and get the interpolated trajectory τ .
- 9: Pick out the active constraints along the trajectory τ according to the rule $1 + c^T p \leq J_a$.
- 10: Pick out the active points for the active constraints.
- 11: Construct problem (21) using the active points and solve it by the PSOC method with the initial guess τ to get the optimal trajectory τ^* .

TABLE I. The IPOPT algorithm options.

Parameter	Description	Setting
tol	望ましい収束許容度	$1e^{-12}$
a_tol	許容される収束許容誤差	$1e^{-16}$
cv_tol	の望ましい/許容される閾値	$1e^{-12}$
acv_tol	constraint violation	
c_tol	の望ましい/許容される閾値	$1e^{-4}$
ac_tol	相補性条件(complementary conditions)	
d_tol	Desired/Acceptable threshold for the dual	1
ad_tol	infeasibility	
solver	ステップ計算に使用される線形ソルバー	MA86

V. BOMPモデルの検証

本節では、自律駐車問題におけるBOMPアルゴリズムのシミュレーションを示す。面積基準(AC)ベースのモデル[22]と円近似法に対するBOMPモデルの計算量と精度の利点を実証された。そして、Turtlebot3で実際の実験を行った。C++コードはLinuxシステムでプログラムされ、シミュレーションはIntel Core i7-7700K CPUと8GB RAMで行われ、4.20GHzで動作する。

A. 自律駐車におけるシミュレーション

本節では、車両運動学モデル(1)と最適化性能指標(3)を用いる。駐車プロセスは低速であり、軌道曲率とその微分を束縛するため[22]、以下の機械的・物理的制約が考慮される。

$$|v| \leq v_{\max}, \quad |\alpha| \leq \alpha_{\max}, \quad |\omega| \leq \omega_{\max} \quad (22)$$

そして、初期状態制約と最終状態制約は次のように扱われる:

$$\begin{cases} x(0) = x_0, y(0) = y_0, \theta(0) = \theta_0 \\ v(0) = 0, v(t_f) = 0, |\theta(t_f)| \leq \epsilon_\theta \\ |(A_y(t_f) + B_y(t_f) - SW)/2| \leq \epsilon_p \\ |(C_y(t_f) + D_y(t_f) - SW)/2| \leq \epsilon_p \\ |(A_x(t_f) + D_x(t_f) - SL)/2| \leq \epsilon_p \\ |(B_x(t_f) + C_x(t_f) - SL)/2| \leq \epsilon_p \end{cases} \quad (23)$$

ここで、 (x_{000}^y, θ) は車両の初期配置、 (A_{xy}^A) は $\{W\}$ で記述した車両コーナー点Aの座標である、 (B_x, B_y) , (C_x, C_y) , (D_x, D_y) はそれぞれ、駐車スポット中心軸に対する車両中心軸の位置偏差と姿勢偏差を表す。したがって、制約条件(1)、(22)、(23)は、BOMPモデルの上位制約を形成する。このアプリケーションでは、BOMPアルゴリズムの一般性、ロバスト性、利点を反映するために、各シナリオで4つのシナリオと3つのケースを設計している。図7は、4つのシナリオと各シナリオにおける1つのケースを示している。赤い車は障害物、黒い車は初期位置、破線車は理想的な駐車場所を示す。4つのシナリオはすべて、日常生活で容易に遭遇する不規則に配置された障害物に対応し、各シナリオにおけるケース間の差は、車両の初期構成のみである(表II参照、略号 S, O, C はそれぞれシナリオ、障害物、ケースを表す)。

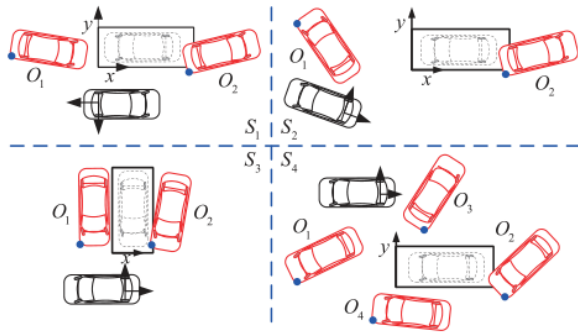


図7. シミュレーションに使用した4つのシナリオ。赤い車は障害物、黒い車は初期位置を示し、破線車は理想的な駐車を示し、それぞれの障害物は車両と同じ大きさである。最初の2つのシナリオは、並列駐車、3つ目のシナリオは、垂直駐車を示している。最後のシナリオは日常生活では普通ではないが、アルゴリズムの能力を示すために設計されたものである。

簡単のため、各障害物は車両と同じ大きさであり、衝突回避は平面矩形のペアが衝突のない状態であることと等価である。障害物座標系 (x, y, θ) のワールド座標系に対する変換パラメータ (x, y, θ) を用いて、各障害物をコンパクトに表現する。これらの変換パラメータ (x, y, θ) を表IIに示す。

駐車場、車両、アルゴリズム1の入力に対する指定されたシミュレーションパラメータは表IIIに示すとおりであり、車両サイズパラメータ、機械的および物理的制約パラメータは[22]に由来する。アルゴリズム1の初期軌道推測 τ_0 については、非常に簡単に指定できる。任意の駐車完了時刻を推測すると、 τ_0 は、任意の t に対して、 $x(t) = x_0$ 、 $y(t) = y_0$ 、 $\theta(t) = \theta_0$ 、 $\alpha(t) = 0$ 、 $v(t) = 0$ 、 $\omega(t) = 0$ 、 $p(t) = 0$ 、 $\lambda(t) = 0$ 、 $v(t) = 0$ で構成される。最後に、 J_2 -関数に用いる座標系は車両の中心に確立されていること、すなわち、障害物を車両の中心座標系に変換するために逆座標変換が適用されていることを指摘する。これらの条件の下で、BOMPアルゴリズムはこれら全ての 4×3 問題を同じシーケンス $\varepsilon \in \{0.1, 0.01\}$ で解く。すなわち、初期段階で2回、最終段階で1回の反復を経て、BOMPアルゴリズムは自律駐車問題全体を解決する。 ε 列の値はアルゴリズム-12収束許容誤差 $1e$ よりはるかに大きいので、初期段階の2つの問題は非常に解きやすい。計算結果を表IVと図に示す。8-11。

次に、BOMPアルゴリズムとACアルゴリズムの計算複雑度の比較を行う。平面上の矩形の点 b と4つの角点 A, B, C, D が与えられたとき、本論文で使用するAC衝突回避制約は以下の通りである：

$$\frac{S_{\Delta bAB} + S_{\Delta bBC} + S_{\Delta bCD} + S_{\Delta bDA}}{S_{\square ABCD}} \geq 1.025 \quad (24)$$

ここで Δ と \square はそれぞれ三角形と長方形の面積を表す。そして、一対の平面矩形の衝突回避問題に対して、8つの非線形制約(24)が存在する。

表II. 各障害物の世界座標系と車両初期配置 (x_0, y_0, θ_0) に対する変換パラメータ (x, y, θ) 。ここでの角度単位は度数であるが、他の場所ではラジアンである。

S_1	O_1	$(-5.6, 0.6, -10.0)$	C_1	$(-6.0, -2.75, 0.0)$
			C_2	$(0.0, -2.75, 0.0)$
	O_2	$(5.8, -0.5, 13.0)$	C_3	$(0.0, -2.75, 180.0)$
S_2	O_1	$(-7.0, 2.8, -55.0)$	C_1	$(-6.0, -2.0, 0.0)$
			C_2	$(-8.0, 0.0, -30.0)$
	O_2	$(5.8, -0.5, 13.0)$	C_3	$(-4.0, -3.0, -36.0)$
S_3	O_1	$(-2.2, 0.4, 0.0)$	C_1	$(-6.0, -2.0, 0.0)$
			C_2	$(1.0, -2.0, 0.0)$
	O_2	$(2.3, 0.3, -5.0)$	C_3	$(-3.0, -2.0, 180.0)$
S_4	O_1	$(-6.3, 0.3, 20.0)$	C_1	$(-5.0, 7.5, 0.0)$
	O_2	$(6.8, -1.0, 40.0)$	C_2	$(-5.0, 5.0, -10.0)$
	O_3	$(2.0, 3.6, 60.0)$		
	O_4	$(-1.5, -2.2, -3.0)$	C_3	$(-1.0, 6.0, 0.0)$

表III. 駐車場、車両、アルゴリズム1の入力に対する指定されたパラメータ。

Parameter	Description	Setting
SL	Parking spot length	6.00 m
SW	Parking spot width	2.50 m
l	Wheel base length	2.800 m
l_1	フロントオーバーハング長	0.960 m
l_2	リアオーバーハング長	0.929 m
W	Vehicle width	1.942 m
v_{\max}	速度の境界	2.00 m/s
α_{\max}	ステアリング角の境界	0.714 rad
ω_{\max}	角速度のパウンド	1.00 rad/s
ϵ_p	ターミナルポジションエラーの境界	0.10 m
ϵ_θ	端子角度誤差の境界	0.17 rad
δ	The safety pseudodistance of the J_2 -function	0.05
J_a	アクティブ制約のしきい値	0.30

そして、(18d)と(18e)をこれらの制約に置き換えることで、ACに基づく運動計画問題が得られ、ACアルゴリズムはアルゴリズム2に示される。これに対応して、ステップ1-2を初期段階、ステップ3を最終段階と呼ぶ。ACは距離を明示的に定義していないため、能動的制約のフィルタリング処理はアルゴリズム2には現れない。22]では、IPOPTオプションと計算時間が示されていないため、表Iのアルゴリズムオプションが使用されている。そして、初期推測 τ_0 は、 $x(t) = x_0$ 、 $y(t) = y_0$ 、 $\theta(t) = \theta_0$ 、 $\alpha(t) = 0$ 、 $v(t) = 0$ 、 $\omega(t) = 0$ と同様に選択される。しかし、ACアルゴリズムは自律駐車問題全体を解決せず、初期段階では失敗する。結果を表IVと図8に示す。最適に近い初期推測は常に複雑な問題の解決を容易にするという概念によれば、以下ようになる。そして、能動的制約のフィルタリングプロセスの影響を示すために、BOMPアルゴリズムの最終段階(ステップ11)の衝突回避制約を、ACベースの手法に置き換えるだけである。すなわち、初期段階のMJ関数は、衝突のない近似最適軌道 τ^* を生成し、アクティブ制約を選び、その後、AC法は、

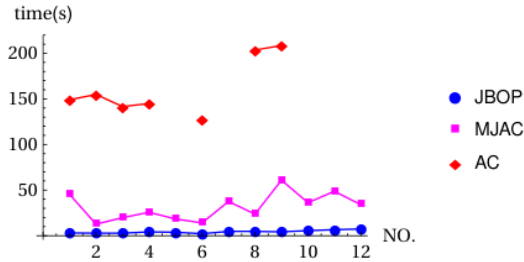


図8. 図8. 12のシミュレーションケースの解法に使用したBOMP, MJAC, ACアルゴリズムの総計算時間。ACアルゴリズムに欠落している値は、アルゴリズムがこれらの問題を解決できないことを意味する。

アルゴリズム2 ACアルゴリズム

入力: 入力:初期軌道推測値 τ_0 、PSOC法における離散ノード数 K 。

出力: 最適軌道 τ^* 。

- 1: Let $K = 15$, using the area criterion constraint (24) to construct the motion planning problem and solve it by the PSOC method with the initial guess τ_0 to get a new trajectory τ .
- 2: Let $K = 30$ and get the interpolated trajectory τ .
- 3: Construct the new problem and solve it by the PSOC method with the initial guess τ to get the optimal trajectory τ^* .

表IV. BOMP, MJAC, ACアルゴリズムの計算結果比較。IとFはそれぞれ初期段階と最終段階の*計算時間、Tは総計算時間、 v^* は最適値である。MJACアルゴリズムは、BOMPアルゴリズムと同じ計算時間を初期段階で持つ。記号×は、ACアルゴリズムが対応する問題を解くことができないこと、より正確には、初期段階ですべて失敗することを示す。

		BOMP				MJAC				AC			
		I (s)	F (s)	T (s)	v^*	I (s)	F (s)	T (s)	v^*	I (s)	F (s)	T (s)	v^*
S_1	C_1	2.82	0.30	3.12	17.14	*	42.46	45.28	16.52	86.43	62.61	149.04	16.98
	C_2	2.45	0.27	2.72	13.07	*	10.20	12.65	12.76	96.34	58.50	154.84	14.64
	C_3	2.41	0.36	2.77	17.33	*	17.40	19.81	17.18	83.15	58.47	141.62	17.58
S_2	C_1	3.85	0.30	4.15	15.73	*	21.86	25.71	15.41	94.01	50.95	144.96	15.26
	C_2	3.55	0.27	3.82	19.40	*	14.81	18.36	19.13	×	×	×	×
	C_3	1.56	0.62	2.18	14.88	*	12.54	14.10	14.88	80.82	46.84	127.66	14.88
S_3	C_1	4.25	0.49	4.74	17.36	*	33.22	37.47	16.65	×	×	×	×
	C_2	4.36	0.32	4.68	14.55	*	19.34	23.70	13.73	132.64	71.00	203.64	13.57
	C_3	3.82	0.44	4.26	19.65	*	57.08	60.90	18.11	123.78	84.73	208.51	18.46
S_4	C_1	5.05	0.45	5.50	16.58	*	31.18	36.23	16.39	×	×	×	×
	C_2	6.03	0.59	6.62	17.81	*	42.68	48.11	14.64	×	×	×	×
	C_3	6.95	0.35	7.30	18.35	*	26.92	33.87	15.89	×	×	×	×

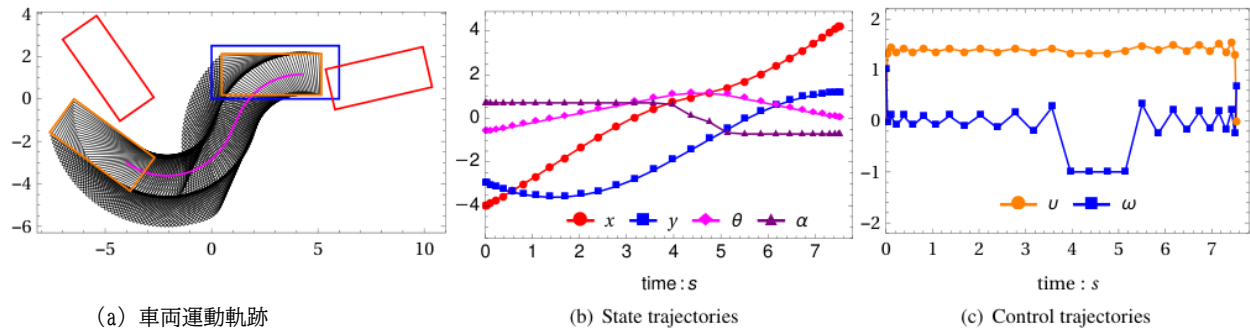


図9. シナリオ2ケース3のBOMPアルゴリズムによる最適化結果。(a)において、オレンジ色の長方形は車両の初期位置と最終位置を表し、マゼンタ色の曲線は車両後輪軸の中心の運動軌跡を示す。(b)の状態軌道は非常に滑らかであるが、(c)の制御軌道は振動している。他のすべての問題における状態軌道と制御軌道は同じ特徴を持つ。

初期推測として τ を用いて最適軌道をfinds outする。このアルゴリズムをMJACアルゴリズムと呼ぶ。このような状況下、MJACアルゴリズムはこれら4×3の問題をすべて解き、その結果を表IVと図8に示す。

MJACアルゴリズムとACアルゴリズムの成功率は、良い初期推測がACアルゴリズムの難易度を下げていることを示している。MJACアルゴリズムとACアルゴリズムの最終段階での計算時間は、アクティブ制約フィルタリング処理が計算速度を数倍高速化することを示している。表IVから、12個の問題に対して最適な無衝突軌道を求めるBOMPアルゴリズムの最短時間は2.18秒、

最長時間は7.30秒であることがわかる。MJACアルゴリズムでは、12.65秒と60.90秒である。一方、解かれた問題のACアルゴリズムでは、127.66秒と208.51秒である。BOMPアルゴリズムが計算速度において有意に優れている理由は以下の通りである：

- 関数衝突回避は線形計画法であるが、ACに基づく衝突回避法では、点間の距離、次に三角形の面積を計算する必要があり、非線形性が高い。線形計画法は最もよく研究されている問題であり、その解を求めることは、

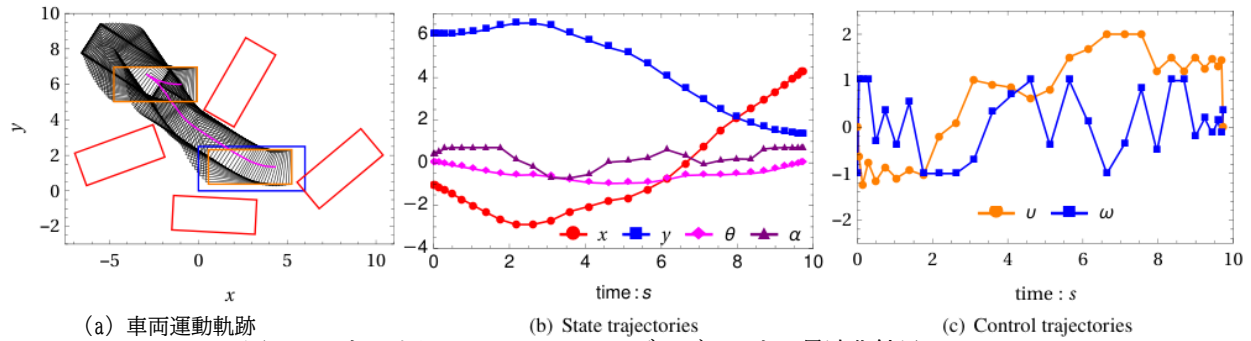


図10. シナリオ4ケース3のBOMPアルゴリズムによる最適化結果。

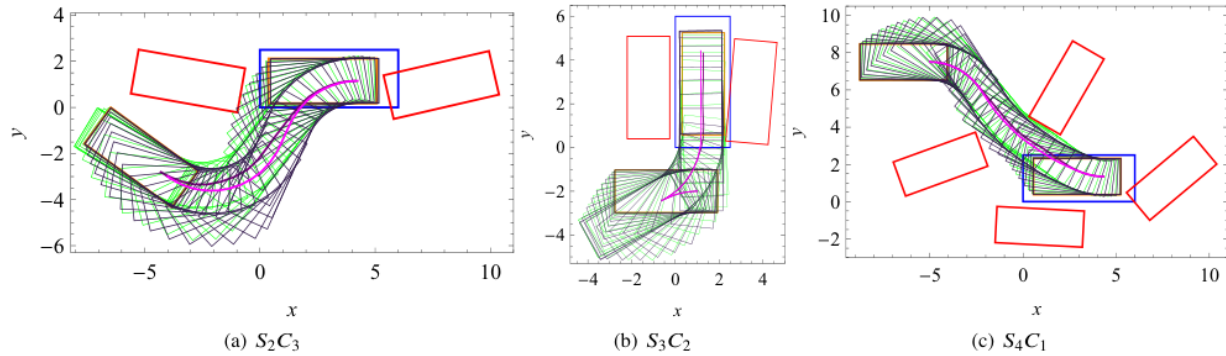


図11. BOMPアルゴリズムにおける2つのステージ間の軌跡結果の比較。緑と紫の軌跡は初期段階の結果であり、黒とマゼンタの軌跡は最終段階の結果である。この結果から、最終的な厳密最適軌道は、前回の近似最適軌道のわずかな変化であり、APMJ関数の収束戦略が適切であることが示された。

非線形制約のある問題よりも本質的にはるかに高速である。

- 関数 f_2 -関数の変数により、BOMPアルゴリズムは初期段階で暗黙のうちに最も近い成分のみを考慮するようになる。しかし、ACに基づく衝突回避法は、一対の矩形間の8点を同時に明示的に考慮する必要があり、その最適化方向を決定することは困難である。
- BOMPアルゴリズムは、最適解に徐々に近づいていく解のシーケンスを見つけ、問題を解く難易度を大幅に下げる。一方、AC法は、複雑な問題で破綻しがちな解を見つけるためにグローバルステップを使用する。

本論文の長さに制限され、最適化された車両運動と、BOMPアルゴリズムによって最適化された状態変数と制御変数の軌道は、図に部分的にプロットされている。9-11. その結果、BOMPアルゴリズムが駐車動作計画問題を効率的に処理できることが実証された。その上、BOMPアルゴリズムは、熟練したドライバーのような複数の操作を必要とする、より複雑なシナリオで自律的に車両を駐車することができる。一方、端末の方位偏差と位置偏差を考慮するため、駐車スポットの中心軸と同様に平行かつ接近して停止する。これらの4×3シミュレーションはすべて、 ϵ_i のようなアルゴリズムオプションを調整することなく行われており、BOMPモデルとBOMPアルゴリズムの頑健性、一般性、統一性を示していることを強調しておく。

しかし、我々のアプリケーションにおける車両運動学、機械的および物理的制約は[22]のものと同様であるため、制御変数の滑らかさの問題と終端ステアリング角の問題もBOMPアルゴリズムに存在する。これらの問題は、モデリングプロセスに内在するものであり、克服するために、いくつかのユニークな、しかし複雑ではないテクニックが必要である。とはいえ、本稿ではBOMPモデルとその解決策に焦点を当てているため、これらの問題は解決に留まっている。

B. ポリゴンVS円近似

前節のシナリオ3における1つの運動計画問題を用いて、ポリゴン近似法と円近似法の異なる影響を説明する。計画タスクは、車両を初期配置(1.0, -2.0, 0.0)から最終配置(1.0, 10.0, $\pi/2.0$)へ移動させることである。図12(a)はBOMPアルゴリズムの結果であり、車両と障害物はやはり矩形で表現されている。図12(b)に円近似法の結果を示す。車両と障害物は、それぞれ半径1.3mの3つの重なり合った円(赤い円)で近似される。この結果から、円近似により物体が膨張し、車両が狭い通路を通過できなくなることがわかる。物体を正確に近似するためには、さらに多くの異なる半径の円が必要であり、これによって計算時間が劇的に増加する。矩形の幅と長さの比がさらに小さくなれば、円近似の欠点はますます深刻になる。

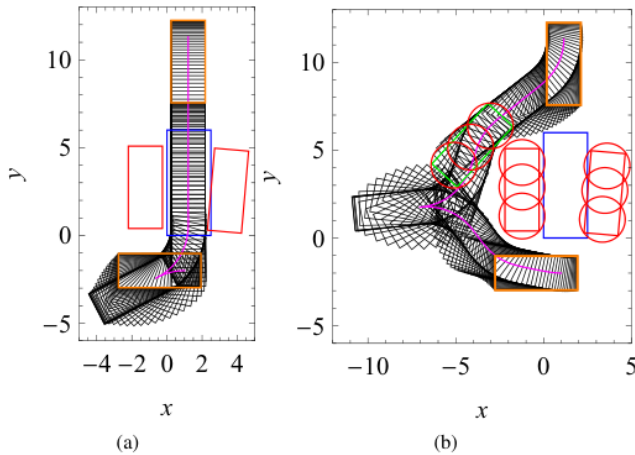


図12. BOMPアルゴリズムと円近似法でそれぞれ得られた車両運動。(b)の緑色の長方形は瞬間の車両を表し、重なり合う3つの円は車両と障害物の近似を表す。

BOMPモデルは、多面体近似法や多角形近似法の近似精度が高く、オーバーラップ率が低いこと、 $\frac{1}{2}$ 関数の単純さ、APMJ関数の高速計算により、多くの領域で運動計画法を補償することができる。

C. 物理実験

移動ロボット(TurtleBot3)を用いた物理実験を行い、BOMPモデルの正しさと有効性を検証した(図13参照)。実験を簡略化するため、ロボットのダイナミクスは考慮せず、ロボットは非常に低速で移動した。SLAM技術を用いると、環境ジオメトリを得ることができ、凸分解[42]によりポリゴンの情報を得ることができる。次に、ロボットの初期位置が与えられると、BOMPアルゴリズムによってグローバル制御コマンド(線速度と角速度)が計算される。本実験では、時間最適軌道を求め、図14に示すようなグローバル制御変数の軌道を求めた。ロボットが制御コマンドを実行しているとき、動的な不確実性とドリフトにより駆動誤差が蓄積される。この誤差が許容誤差を超えると、再計画プロセスが発動する。ロボットのオドメトリ、ローカライゼーション、制御フレームワーク、誤差累積計算、リプランニングサイクルの詳細は、今後の研究課題として紹介する予定である。この実験のビデオは<http://www.hust.edu.cn>。

VI. 結論と展望

本論文では、障害物環境におけるロボットの軌道生成と最適化のための一般的で統一的なBOMPモデルを提案し、複雑な高精度ロボットの運動計画問題に対する実用的なアプローチとして最適制御法を提案した。上位最適制御はロボットの非線形ダイナミクス用に設計され、下位 $\frac{1}{2}$ -関数はジオメトリ無衝突制約用である。複雑な自律駐車シナリオでのシミュレーションとTurtlebot3での実験により、



図13. 本実験の環境設定。非常に小さな領域(1.5m×1.5m)に7つの障害物があり、すべての実験の目的は、Turtlebot3を0.5m×0.5mの白い正方形の中心で停止させることである。

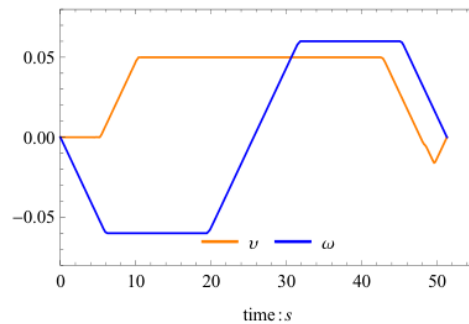


図14. ロボットの初期位置が与えられたとき、BOMPアルゴリズムによって計算されたグローバル制御変数の軌跡。

BOMPモデルの計算の優位性と効率性を実証。本稿のハイライトは以下の点にある。

- 1) BOMPモデルは衝突を避けるために $\frac{1}{2}$ 関数線形計画法を利用する。BOMPモデルは $\frac{1}{2}$ 関数の単純さにより、ロボットの運動計画問題を非常に高速かつ高精度に解くことができる。
- 2) BOMPモデルは、ユーザが指定した制約を上位または下位の制約に組み込むことができるオープンなフレームワークである。そして、BOMPモデルは次元に依存しない。したがって、このモデルはマニピュレータ、車両、ヒューマノイドロボットなどに活用できる。
- 3) BOMPアルゴリズムは、MAKKT理論の収束特性をフルに活用し、MJ関数とAPMJ関数を別々のステージで適用する。この収束戦略により、BOMPアルゴリズムは非常に高速になる。

しかし、多くの側面を深く包括的に研究する必要がある。まず、乱雑で障害物が集中する環境で移動するロボットの計算効率の一つの主要な問題である。今後の研究は、アルゴリズムの初期段階に集中すべきであり、一つの可能な解決策は、サンプリングに基づく方法のような、これまでの研究理論を組み合わせることである。実際のアプリケーションでは、いくつかの簡単なテクニックを使うことができる。駐車場問題を例にとり[23]、駐車場ケースの有限分類を行い、オフラインで軌跡を計算し、データベースを構築する。次に、実際の駐車プロセスにおいて、アルゴリズムを初期化するために、現在の状況に最も近い記録された解を選び出し、最後にオンラインで問題を解く。第二に、移動障害物のある動的環境におけるロボットの運動計画は、最も困難で重要かつ重要な研究分野とみなされている。

BOMPモデルが静的な問題を解決するのは現在のところ、静的な問題しか解決していないのに対し、障害物移動予測・推定理論をBOMPフレームワークに採用する研究はまだ広範囲に及んでいる。第三に、異なるレベルの適切な目的と制約があれば、マルチレベル問題は複数の目的の最適化問題と密接な関係がある。しかし、マルチレベル問題を解くことはバイレベル問題よりもはるかに難しく、関連する理論は提案されていない。これら3つの未解決の問題は、いずれも知的ロボットの開発にとって挑戦的であり、有意義なものであり、今後の研究課題として議論する予定である。さらに、本論文のTurtlebot3でのシミュレーション結果や簡単な実験にもかかわらず、実際の複雑なアプリケーションでの実験や提案アルゴリズムの検証には、広範な研究が必要である。

REFERENCES

- [1] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.
- [2] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, 2016.
- [3] H. Vorobieva, S. Glaser, N. Minoiu-Enache, and S. Mammar, "Automatic parallel parking in tiny spots: path planning and control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 396–410, 2015.
- [4] X. Du and K. K. Tan, "Autonomous reverse parking system based on robust path generation and improved sliding mode control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1225–1237, 2015.
- [5] B. Muller, J. Deutscher, and S. Grodde, "Continuous curvature trajectory design and feedforward control for parking a car," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 3, pp. 541–553, 2007.
- [6] D. Xu, Y. Shi, and Z. Ji, "Model-free adaptive discrete-time integral sliding-mode-constrained-control for autonomous 4wv parking systems," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 1, pp. 834–843, 2018.
- [7] W. Liu, Z. Li, L. Li, and F.-Y. Wang, "Parking like a human: a direct trajectory planning solution," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3388–3397, 2017.
- [8] Y. Tazaki, H. Okuda, and T. Suzuki, "Parking trajectory planning using multiresolution state roadmaps," *IEEE Trans. Intell. Veh.*, vol. 2, no. 4, pp. 298–307, 2017.
- [9] R. V. Cowlagi and P. Tsiotras, "Hierarchical motion planning with dynamical feasibility guarantees for mobile robotic vehicles," *IEEE Trans. Robot.*, vol. 28, no. 2, pp. 379–395, 2012.
- [10] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986.
- [11] M. B. Oetiker, G. P. Baker, and L. Guzzella, "A navigation-field-based semi-autonomous nonholonomic vehicle-parking assistant," *IEEE Trans. Veh. Technol.*, vol. 58, no. 3, pp. 1106–1118, 2009.
- [12] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.
- [13] W. Chi, C. Wang, J. Wang, and M. Q.-H. Meng, "Risk-dtrrt-based optimal motion planning algorithm for mobile robots," *IEEE Trans. Autom. Sci. Eng.*, 2018.
- [14] L. Janson, B. Ichter, and M. Pavone, "Deterministic sampling-based motion planning: Optimality, complexity, and performance," *Int. J. Robot. Res.*, vol. 37, no. 1, pp. 46–61, 2018.
- [15] O. Salzman and D. Halperin, "Asymptotically near-optimal rrt for fast, high-quality motion planning," *IEEE Trans. Robot.*, vol. 32, no. 3, pp. 473–483, June 2016.
- [16] W. Lim, S. Lee, M. Sunwoo, and K. Jo, "Hierarchical trajectory planning of an autonomous car based on the integration of a sampling and an optimization method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 613–626, 2018.
- [17] 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," *Int. J. Robot. Res.*, vol. 32, no. 9–10, pp. 1164–1193, 2013.
- [18] 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," *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1251–1270, 2014.
- [19] J. Park, S. Karumanchi, and K. Iagnemma, "Homotopy-based divide-and-conquer strategy for optimal trajectory planning via mixed-integer programming," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1101–1115, 2015.
- [20] J. Pan, L. Zhang, and D. Manocha, "Collision-free and smooth trajectory computation in cluttered environments," *Int. J. Robot. Res.*, vol. 31, no. 10, pp. 1155–1175, 2012.
- [21] D. R. Robinson, R. T. Mar, K. Estabridis, and G. Hewer, "An efficient algorithm for optimal trajectory generation for heterogeneous multi-agent systems in non-convex environments," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 1215–1222, 2018.
- [22] B. Li and Z. Shao, "A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles," *Knowl. Based Syst.*, vol. 86, pp. 11–20, 2015.
- [23] B. Li, K. Wang, and Z. Shao, "Time-optimal maneuver planning in automatic parallel parking using a simultaneous dynamic optimization approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3263–3274, 2016.
- [24] J. Nilsson, M. Brännström, E. Coelingh, and J. Fredriksson, "Lane change maneuvers for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1087–1096, 2016.
- [25] Y. Rasekhipour, A. Khajepour, S.-K. Chen, and B. Litkouhi, "A potential field-based model predictive path-planning controller for autonomous road vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1255–1267, 2016.
- [26] Z. Ye, K. Li, M. Stapelbroek, R. Savelsberg, M. Günther, and S. Pischinger, "Variable step-size discrete dynamic programming for vehicle speed trajectory optimization," *IEEE Transactions on Intelligent Transportation Systems*, no. 99, pp. 1–9, 2018.
- [27] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010, vol. 19.
- [28] A. E. Bryson, *Applied optimal control: optimization, estimation and control*. Routledge, 2018.
- [29] I. M. Ross and M. Karpenko, "A review of pseudospectral optimal control: From theory to flight," *Annu. Rev. Control*, vol. 36, no. 2, pp. 182–197, 2012.
- [30] Y. Xiong, "Generalized criterion for minimum zone and computer arbitrament," *Acta Metrologica Sinica*, vol. 8, no. 2, pp. 1–8, 1987.
- [31] Y. Xiong and H. Ding, "General criterion and control strategy of collision-free movement for manipulators," *Int. J. Robot. Autom.*, vol. 4, no. 2, pp. 75–80, 1989.
- [32] F. Benita and P. Mehlitz, "Bilevel optimal control with final-state-dependent finite-dimensional lower level," *SIAM J. Optim.*, vol. 26, no. 1, pp. 718–752, 2016.
- [33] A. Sinha, P. Malo, and K. Deb, "Evolutionary algorithm for bilevel optimization using approximations of the lower level optimal solution mapping," *Eur. J. Oper. Res.*, vol. 257, no. 2, pp. 395–411, 2017.
- [34] —, "A review on bilevel optimization: from classical to evolutionary approaches and applications," vol. 22, no. 2, pp. 276–295, 2018.
- [35] P. Hansen, B. Jaumard, and G. Savard, "New branch-and-bound rules for linear bilevel programming," *SIAM J. Sci. Comput.*, vol. 13, no. 5, pp. 1194–1217, 1992.
- [36] L. Vicente, G. Savard, and J. Júdice, "Descent approaches for quadratic bilevel programming," *J. Optim. Theory Appl.*, vol. 81, no. 2, pp. 379–399, 1994.
- [37] S. Albrecht, M. Leibold, and M. Ulbrich, "A bilevel optimization approach to obtain optimal cost functions for human arm movements," *Numer. Algebra Control Optim.*, vol. 2, no. 1, pp. 105–127, 2012.
- [38] J. Dutta, K. Deb, R. Tulshyan, and R. Arora, "Approximate kkt points and a proximity measure for termination," *J. Glob. Optim.*, vol. 56, no. 4, pp. 1463–1499, 2013.
- [39] T. Lozano-Perez, "Spatial planning: A configuration space approach," *IEEE Trans. Comput.*, no. 2, pp. 108–120, 1983.
- [40] S. Wright and J. Nocedal, "Numerical optimization," *Springer Science*, vol. 35, no. 67–68, p. 7, 1999.
- [41] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, 2006.
- [42] K. Mamou and F. Ghorbel, "A simple and efficient approach for 3d mesh approximate convex decomposition," in *Proc. IEEE Int. Conf. Image Proc. (ICIP)*. IEEE, 2009, pp. 3501–3504.

APPENDIX

この付録では、障害物の数が増えるにつれて、BOMPモデルの複雑さがスケールすることを示す。一般性を損なわない範囲で、(18)の $x \in \mathbb{R}^n$ 1, $u \in \mathbb{R}^n$ 2, $p \in \mathbb{R}^n$ 3, $Q \in \mathbb{R}^{m \times n}$ 3を仮定する。

次に、変換された単一レベル最適制御モデル(20)において、 $\lambda \in \mathbb{R}^{n_3}$, $v \in \mathbb{R}^m$. PSOC法では、離散ノードは、区間 $[-1, 1]$ にあるLegendre-Gaussコロケーション点またはChebyshev-Gaussコロケーション点として選択される。本論文のLegendre-Gaussコロケーション点を例にとると、区間 $[-1, 1]$ の $K + 1$ 個の離散ノード $[\tau_0, \tau_1, \dots, \tau_K]$ が与えられ、 $x, u, p, Q, f(x, u)$ を次のように離散化する：

$$\begin{aligned} x_k &= x\left(\frac{t_f}{2} + \tau_k \frac{t_f}{2}\right), & u_k &= u\left(\frac{t_f}{2} + \tau_k \frac{t_f}{2}\right) \\ p_k &= p\left(\frac{t_f}{2} + \tau_k \frac{t_f}{2}\right), & Q_k &= Q(x_k), \\ f_k &= f(x_k, u_k) \\ k &= 0, 1, 2, \dots, K \end{aligned} \quad (25)$$

そして、 $X = [x_0, x_1, \dots, x_K]$, $F = [f_0, f_1, \dots, f_K]$ とすると、 $X \in \mathbb{R}^{n \times (K+1)}$, $F \in \mathbb{R}^{n \times (K+1)}$. 各離散時間ノードで満足される制約を制限すると、(20)の大規模なスパース非線形最適化問題が得られる：

$$\min \quad \beta g(x_i, x_f, t_f) + (1 - \beta) \frac{t_f}{2} \sum_{k=0}^K \omega_k L(x_k, u_k) \quad (26)$$

$$\text{s.t.} \quad XD = \frac{t_f}{2} F \quad (26a)$$

$$x_0 = x_i, \quad x_K = x_f \quad (26b)$$

$$x_k \in X, \quad u_k \in U \quad (26c)$$

$$1 + c^T p_k \geq \epsilon + \delta \quad (26d)$$

$$Q_k p_k = b, \quad p_k \geq 0, \quad \lambda_k \geq 0 \quad (26e)$$

$$\|c - \lambda_k + Q_k^T v_k\| \leq \sqrt{\epsilon}, \quad \lambda_k^T p_k \leq \epsilon \quad (26f)$$

$$k = 0, 1, 2, \dots, K$$

ここで、 $D \in \mathbb{R}^{(K+1) \times (K+1)}$ は定数行列、 ω_k は定数スカラーである。

したがって、 $K' = K + 1$ とすると、 $n_1 * K'$ 変数 x_k , $n_2 * K'$ 変数 u_k , $n_3 * K'$ 変数 p_k , $n_3 * K'$ 変数 λ_k , $m * K'$ 変数 v_k と1変数 t_f がある。制約条件(26a)~(26f)に関する制約数を表Vに示す。最適化変数の総数と制約数は、それぞれ $(n_1 + n_2 + 2n_3 + m) * K' + 1$ と $(2n_1 + n_2 + 2n_3 + m + 3) * K' + 2n_1 + 1$ であり、数1は $t_f > 0$ から来る。これらは常に非常に大きいので、変換された非線形最適化問題(26)を解くのは難しい。セクションVの自律駐車アプリケーションでは、4つの状態変数と2つの制御変数がある。また、平面矩形対衝突回避問題のMJ関数では、8つの変数、3つの等式制約、8つの不等式制約がある(もう1つの不等式制約はアクティブではないので省略する)。そこで、自律駐車アプリケーションでは、まず平面矩形対衝突回避問題を考え、運動計画制御問題(20)を構成し、制約条件(20a)~(20f)を離散化して大規模非線形最適化問題を得る、 $n_1 = 4, n_2 = 2, n_3 = 8, m = 3$ 、それぞれ $6K' + 1 + 19K'$ 変数と $10K' + 9 + 22K'$ 制約がある。矩形衝突回避制約の組が n 個ある場合、変数と制約の数はそれぞれ $6K' + 1 + 19nK'$ 個の変数と $10K' + 9 + 22nK'$ 個の制約となる。

すなわち、BOMPモデルは障害物数に対して線形複雑度である。表V. 問題(26)の制約数、ここで $K' = K + 1$ 。

Label	Numbers	Label	Numbers
(26a)	$n_1 * K'$	(26d)	K'
(26b)	$2n_1$	(26e)	$(m + 2n_3) * K'$
(26c)	$(n_1 + n_2) * K'$	(26f)	$2K'$

シナリオ4の適用では、4つの矩形ペアがあり、 $K' = 15$ とすると、変数番号と制約番号はそれぞれ $6 * 15 + 19 * 15 * 4 + 1 = 1231$ と $10 * 15 + 9 + 22 * 15 * 4 = 1479$ である。



Shenglei Shiは、2016年に中国武漢の華中科技大学(HUST)で機械工学の学士号を取得した。現在、博士号取得を目指している。HUSTで学位

His current research interests include motion planning and trajectory optimization in robotics.



Youlun Xiongは、1962年に西安交通大学機械工程学院を卒業し、1966年に西安交通大学機械工程学院を卒業した。1966年から1980年まで、中国武漢の華中科技大学(HUST)機械工程学院に勤務した。1980年から1982年まで、英国シェフィールド大学制御工学科客員研究員。1982年、HUST機械工学科に戻り、教授となる。

1988年から1989年まで、英国サalford大学航空機械工学科客員教授。1995年、中国科学院の学術顧問に選出された。研究テーマは、ロボット工学、精密測定、インテリジェンス製造、製造自動化など。



2001年、武漢理工大学(中国・武漢市)より車両運用工学の学士号、M.S.より取得。武漢の華中科技大学(HUST)で2006年にメカトロニクス工学の博士号を、2010年に博士号を取得した。2011年、HUST制御科学工学科博士研究員。現在、HUSTデジタル製造装置・技術国家重点実験室講師。

His current research interests include flexible electronics manufacturing and roll-to-roll Web transport system control.



1998年、中国武漢の華中科技大学(HUST)で機械工学の博士号を取得。1999年から2003年まで、香港城市大学および香港中文大学に博士研究員として、米国マサチューセッツ州ウースターのウースター工科大学に研究員として在籍。HUSTの長江教授である。中国国家科学基金特別青年奨学生を受賞している。

His research interests include biomechatronic prostheses, rehabilitation robotics, and robot motion planning and control.