

双腕による組立姿勢探索

○森山 龍太 (大阪大学), 万 偉偉 (大阪大学), 原田 研介 (大阪大学)

1. はじめに

近年, 産業用ロボットの発展に伴い, 製造業の現場においてロボットによる自動化の試みが盛んである. しかしながら, 作業の多種多様性により, 組立工程の自動化は遅れている. 例えば単腕作業を考えてみよう. 単腕であると, 組み立て作業時に物体を作業領域へ移動する必要があり, 無駄な動作が増える. さらに, 物体を組み立てる際に接触した位置からずれが発生するため, 固定させる器具も必要になる. ここで, 双腕の場合を考えると, 物体同士は把持により固定されているため, ずれが少なく, 作業領域を固定する必要がなくなる. また, 再把持を必要とするとき, 双腕で行うといった新たな手法も試すことができ, より組み立て作業の幅が広がると考えられる.

従来の手法では, 双腕ロボットを扱う際は, 人の監視下の下であったり, 決め打ちの動作が主流である. そのため異なる製品に対して, その都度決め打ちの位置や姿勢を入力する必要があった. しかし, 上記の工程を, 単腕で行っている自動組み立て作業を双腕に拡張するだけではできない. なぜなら, 従来までは組み立て作業領域は固定されたものであるが, 双腕であると, 動作の都度作業領域へ移動することなく, 左右どちらかのアーム上で組み立てることができる. つまり, 作業領域が可変であるため, 重力の影響も加味しながら組み立て作業計画を行わなければならない.

そこで本稿では, 双腕ロボットによる組み立て作業計画における各動作に着眼を置いて, 可変する作業領域に合わせた組み立て計画を行う手法について検討する. その際, 双腕における把持計画, 衝突検出, IK (Inverse Kinematics), や RRT (Rapidly exploring Random Tree) を用いることとする.

2. 提案手法

本提案手法について説明する. 手法は大きく分けて 5 つの段階に分かれる. 1 つ目は, 先行研究 [2] で行われた把持計画をもとに, 物体ごとの 3D モデルに対して把持プランをデータベース化 [3] する段階である. 2 つ目は, 組み立て位置と姿勢を探索する段階である. 3 つ目は, 把持プランデータベース, および組み立て位置と姿勢探索を踏まえたアームの手先の衝突検出を行う段階である. 4 つ目は, 上記 3 つの工程で取得したデータを用いた把持姿勢と組み立て姿勢の IK 計算のアルゴリズムの 2 つの手法を行う段階である. 5 つ目は, RRT を用いた双腕の組み立て計画の段階である. 本節では, これら 5 つのプロセスを順を追って説明していく.

2.1 把持計画

把持に用いられるハンドにはいくつか種類が存在する. 本研究では, その中の 'Robotiq[4]' と呼ばれるハンドを使用する. まず, 物体の把持を行う上で, 3D オブ

ジェクトの形状を取得する. 把持位置の探索は, 先行研究 [1] を用いている. 手法としては, 物体に対して, 表面の法線ベクトルを抽出し, ランダムにベクトルを選び, 閾値で固定された範囲内の法線ベクトルを抽出し一つの面とする. 次に, 一定以上離れた法線ベクトルを選び上記を繰り返すことで, 物体の平行面を重複を含めて割り出すことができる. 抽出された平行面と対となる反対側の平行面をハンドの把持位置とし, 回転を加える. ここでは把持の安定性を高めるために, 物体の境界面付近の把持は排除している. 最後に, 物体とハンドが衝突している姿勢を除外することで, 把持プランを作成し, それらをデータベースに保存する.

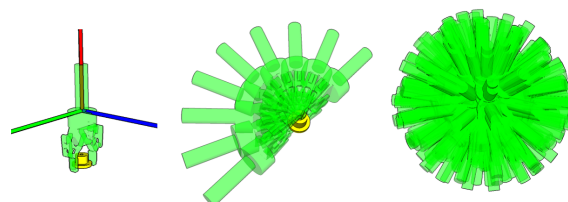


図1 物体の把持姿勢

2.2 組み立て位置と姿勢の探索

組み立てる際に必要な情報は, 2 つの物体の形状と, それらの組み立て姿勢である. つまり, 基準となる物体を決め, 基準からの相対的な位置と回転座標を入力することで, 物体の組み立てモデルを作成する. 次に必要な情報は組み立てられた物体の絶対位置と姿勢である. 絶対位置は, Nextage の胸元手前に固定しているが, 回転は制限していない. 回転の決め方は Icosphere を用いている. Icosphere でサンプリングされた球の位置をベクトルとし, 回転座標に変換することで物体の回転が得られる. 基準となる物体は回転座標を加え, もう一つの物体は, 相対的な回転座標と icosphere で得られた回転座標を加える. 以上の動作をすべてのサンプリングポイントで行うことで, 2 つの物体の組み立て位置と姿勢が得られる.

また, 実際のシミュレーションで行う際には, 左右のアームで物体を持つため, 組み立て姿勢が左右反対であるとロボットの構造上ほぼ IK が存在しない. そこで, 組み立て位置探索の際に物体がロボットの中心位置から把持するアームと同方向に存在する制約をつけることで, 余分な組み立て姿勢を除外することができ計算を早くしている.

2.3 把持と組み立て姿勢における手先の衝突検出

衝突において障害物となるものは, 作業領域とロボット自身である. 物体はテーブルの上に乗っていると考える. [把持計画] でデータベース化した把持の中にはテーブルに衝突するものが多く存在する. 衝突の検出法とし

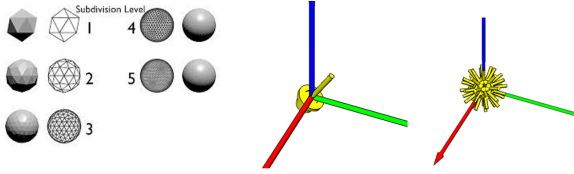


図2 物体の組み立姿勢

ては,ballによるあたり判定は大きすぎるため,ほぼすべての把持が作業領域と衝突となってしまふ. そのため衝突判定は,Mesh モデルで行わないといけない. しかし,ロボット全身の Mesh モデルによる衝突検出は IK 探索などの時間も含まれ,さらに,Mesh モデル自体の衝突検出にも時間がかかる. 改善手法として,アームの手先だけの Mesh モデルを用いて IK を計算せず,衝突のみにこだわることで判定する.

また,組み立て位置での衝突検出も行う. 一方のアーム姿勢に対して,もう一方のアームの中には衝突しないものがある. そのため,組み立ての際,他方の物体との衝突は除外するが,アーム同士の衝突は除外しない. つまり,基準となるアームの把持に対して,その他のアームの把持すべてが衝突することはまれであり,すべての把持の組み合わせに対して衝突検出を行う必要がない. そのため,組み立て位置のアームの衝突を検出する前処理として,右手の座標が大きく左にずれない制約(左右対称)をかけることで,余分な計算を省くこととする.

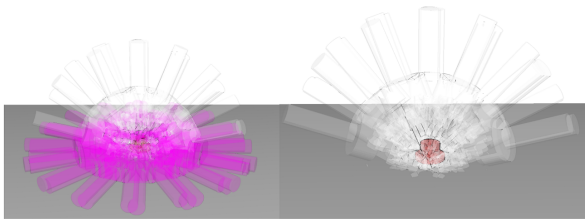


図3 把持における衝突(紫)

2.4 把持姿勢と組み立て姿勢の IK 計算のアルゴリズムの2つの手法

取得したいデータは,双腕の把持位置と姿勢,組み立て位置,姿勢,および直前の~が必要である. これらの探索のために,組み立て姿勢における IK の計算と衝突検出が必要であり,その手法を2つ以下に示す.

[手法1]

手法1では,組み立て姿勢の候補を選び,把持の IK と衝突検出を同時に行い,すべての姿勢が見つかり次第終了する. 詳しいアルゴリズムは以下に示す.

- 1:組み立て姿勢の選択
- 2:右アームの把持位置の候補の選択
- 3:把持位置の IK が存在しなければ2へ
- 4:プレ把持位置の IK or 衝突が無理なら2へ
- 5:組み立て位置については3と4と同様
- 6:左アームについても同様に行う
- 7:すべての把持位置でも無理なら1へ

[手法2]

手法2では,組み立て姿勢の候補を選びすべての把持位置に対して衝突検出のみを行い衝突がないものをリスト化する. 次に左右のアームの IK を2重ループで探索し見つかれば次第終了する. 詳しいアルゴリズムは以下に示す.

- 1:組み立て姿勢の選択
- 2:把持位置の候補の選択
- 3:把持位置の衝突が無理なら2
- 4:プレ把持位置の衝突が無理なら2へ
- 5:組み立て位置については3と4と同様
- 6:左アームについても同様に行う
- 7:左右の把持と組み立て位置と姿勢のリストを作成し2へ
- 8:リストを参照しながら左右の IK を計算し見つからなければ1へ

表1 手法1・2の比較

	手法1	手法2
IK の存在が既知である組み立て姿勢の計算時間	1.1445936 秒 (10 回平均)	6.3096538 秒 (10 回平均)
10 個目の組み立て姿勢で IK が発見できる計算時間	150.17278133 秒 (10 回平均)	88.95728258 秒 (10 回平均)
1 つの組み立て姿勢でのアームの依存関係	右アームの候補は最大一つであり右アーム完全依存型	左右のアームそれぞれの候補は複数個存在

2.5 RRT を用いた双腕の組み立て計画

RRT の原理を簡単化するために2次元で示す. 組み立て計画は, start と goal の2点と障害物を定義した平面内で行う. まず,ランダムな定義域内のサンプリングされたポイントと,現在伸びている木の最短距離にある点までの直線を考え,定義されたノルムより大きければノルムまでの線分とし,それ以下ならその線分を使う. 次に線分内で衝突検出を行い,衝突がなければ木を伸ばす. 任意の確率(本研究では3割)で goal をサンプリングポイントとし,衝突を検出する. 以上の処理を start と goal がつながるまで繰り返す. しかし,得られた path は不規則であるため,滑らかな path にするために,何らかの補正が必要となる. 手法としては,得られたパスの任意の2点を選び,2点の線分に衝突がなければ間の path を消去し置き換えることとする. 以上の処理を任意の回数(本研究では500回)行う.

以上の操作を双腕で行う場合,アームは片腕が6自由度であるため,12次元のRRTを解く問題となる. 原理は2次元と同じであるが,定義された平面はアーム

のジョイントリミットとなり, start と goal は 3 次元空間何の位置・姿勢を IK でジョイント (12 自由度) に置き換えたものである。この際, 左右のアームは同時に計画されるため, 右アーム側に障害物が多く存在し, 左アームには, 障害物がない場合, 左アームは右アームが障害物をよけるまでその場で待機や微小運動を行う。一方, 片腕ずつ RRT を行い単位時間ごとに衝突を検出すると再計算や複雑になり, 計算時間が大幅に増えると考えられる。そこで, RRT を 12 次元とし, 衝突を双腕で検出するよう構築した。

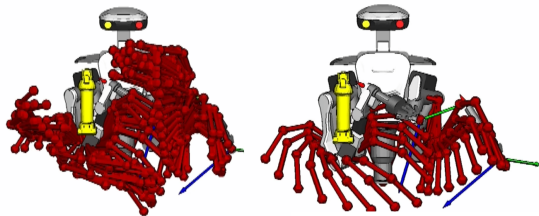


図 4 最適化の適応前後の RRT

3. 結果

3.1 手法 1・手法 2 の比較

手法 1 のメリットとしては,[表 1] に示したように計算時間が早いことが挙げられる。右アームの計算を IK と並行して行っているため, 右アームの計算が終了した時点で左アームの探索に移るためである。しかし, デメリットとしては, 組み立て姿勢の候補 1 つに対して, 右アームの IK は一つのみになってしまうことが挙げられる。そのため, 右手完全依存となってしまう組み立て姿勢の多様性が失われてしまうことが想定される。

手法 2 のメリットとしては, 手法 1 と対となる関係であり, 右手依存が緩和されることが挙げられる。理由は, 手法 2 では, 衝突検出を先に行っており, その際に作成されたリストの候補から自由に IK を探索できるためである。しかし, 2 重ループにより, 外ループの依存性を担保している。デメリットは, すべての把持姿勢の衝突を検出し, リスト化しているため, 計算時間が大幅に増えることが挙げられる。

上記の考察から, 手法 1 は, 計算時間が早い代わりに候補数が少なくなり, 手法 2 は, 計算時間は遅いが候補数は多いということが想定される。今回の場合, 2 章の [把持と組み立て姿勢における手先の衝突検出] を行い, 得られたデータの候補数が多いため, 手法 1 でも問題はないと考えられる。しかし, 限られた把持姿勢の場合, 解が見つからない可能性も存在すると考えられる。

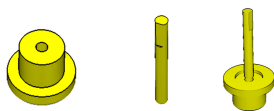


図 5 組立物体

3.2 動作の結果

組み立てる物体は [図 5] に示す 2 つである。2 章で構築した原理に基づき組立作業を行った結果を [図 6] に示す。

入力は 2 つのオブジェクトの種類と位置, オブジェクトの組立相対位置・回転座標である。つまり, 入力は二つのオブジェクトの位置と, どのように組み合わせるのかである。入力は把持や IK などが存在し, 再把持を行わない条件であれば任意である。自動で探索し選ばれた動作計画は, 探索の際に最初に見つけた Path を行うため入力による Path の選択は行うことができない。Path 選ぶことはできないが, うまく制約を与えることで理想に近い Path を動作させることができると考える。

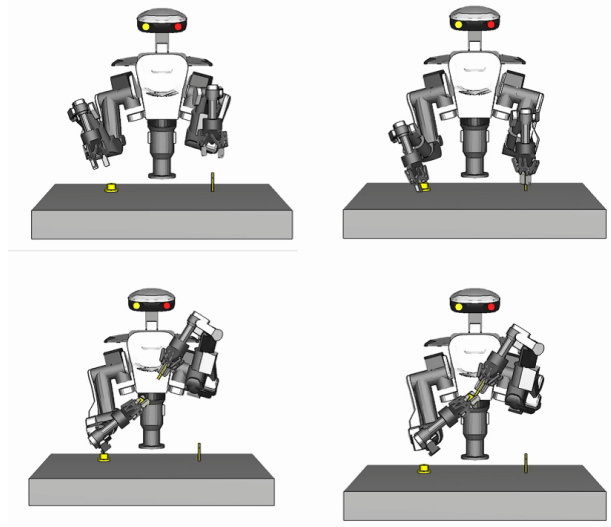


図 6 組立におけるポイント

4. まとめ

本稿では, 双腕ロボットにおける組み立て作業計画の一連動作手順としては, 物体ごとの把持探索のデータベース化, Icospher による組み立て位置探索, アームの手先のみ衝突検出による, 計算時間短縮を行う。および 2 つの手法を用いた把持位置・姿勢や組み立て位置・姿勢の IK 探索を構築した。さらに, 得られた位置・姿勢をもとに 12 次元の RRT を用いて双腕ロボットに組み立て計画を自動で行わせたものである。

参考文献

- [1] LaValle, Steven M.; Kuffner Jr., James. (2001). "Randomized Kinodynamic Planning" (PDF). The international Journal of Robotics Research (IJRR). 20 (5): 378-400
- [2] Weiwei Wan, Kensuke Harada, and Kazuyuki Nagata, Assembly Sequence Planning for Motion Planning, Assembly Automation (AA), 2017
- [3] Weiwei Wan and Kensuke Harada, Regrasp Planning using 10,000s of Grasps, IEEE International Conference on Intelligent Robots and Systems (IROS), 2017
- [4] <https://robotiq.com/products/adaptive-grippers>
- [5] 原田研介 ロボットの動作計画における最適化, 日本ロボット学会誌, Vol.32, No.6, pp.508-511, 2014.