

空間的制約を用いた障害物回避動作計画\*

1N-5

向井 理朗<sup>†</sup>, 大和田 勇人<sup>‡</sup>, 溝口 文雄<sup>†</sup>

東京理科大学 理工学部<sup>†</sup>

1 はじめに

産業用ロボットの制御を行なう場合、作業環境内に障害物等がある場合にはその制御のためのオペレータの負担は非常に大きなものとなる。このようなオペレータの負担を軽減するため、障害物回避を中心としたロボットの動作計画が用いられる。本稿では Block-World での組み立て作業について、Motion Planning により途中経路を導出し、実際にロボットを制御する。

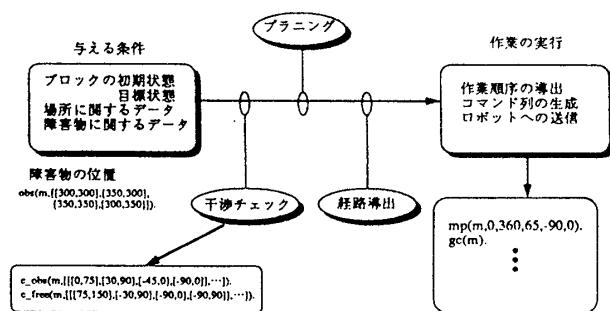


図 1: 本研究の枠組

本稿では対象とするロボットを自由度4の多関節型マニピュレータとし、Tomas Lozano-Pérez[3]の方法を改良した簡易コンフィギュレーション空間上で経路導出し、ロボットを直接制御する。実際の組み立て作業を行なう際には障害物の存在等により空間的な制約が存在する。そこで本稿では接近・離脱方向の考慮、複数のマニピュレータの使用により空間的な制約を解決する方法を示す。試作したシステムではブロックや障害物の位置やブロックの状態を与えることにより、ブロックの組み立てを実行する際の途中経路を導出し、出力としてはロボット制御コマンドの形式で獲得する。

2 空間的な制約の考慮

本稿で対象としている空間的な制約とは障害物の存在により作業に支障をきたす制約である。使用可能なロボットが限定されている場合、作業環境によっては実行が不可能なタスクが数多く存在する。

例えば、大きな障害物が存在する場合には、マニピュレータの構造や障害物の位置、形状によってマニピュレータの作業空間に空間的な制約が生じる。このように1台のマニピュレータでは処理できない作業について2台のマニピュレータを用いることにより、マニピュレータ同士がお互いの空間的な制約をカバーすることにより、ある程度の空間的な制約を解消することができる。

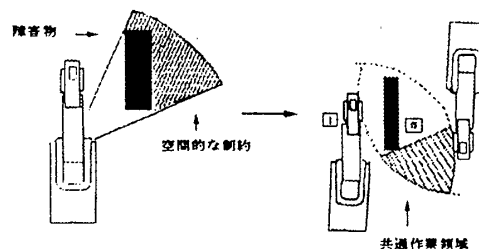


図 2: 空間的な制約

そこで1台では初期状態・目標状態が空間的な制約により不可能な場合あるいは1台では経路が複雑となり効率が悪くなる場合に2台のロボットによる協調作業を行なう。

3 本研究のアプローチ

従来のコンフィギュレーション空間法 [3] では物体の姿勢を一意に定めるパラメータによって張られるパラメータ空間上で自由空間・障害物空間を定義し、この上で経路を導出する。干渉チェックの際に計算量が関節の数に依存して増大するという問題点を持つ。

障害物回避問題としては与えられたタスクに対し、必要最小限の自由空間が存在すればよいので、次のように自由空間・障害物空間を定義する。

**自由空間** マニピュレータが障害物と干渉をおこさない関節角度領域。

**障害物空間** 干渉をおこす関節角度の組合せが含まれている関節角度領域。

自由空間・障害物空間はそれぞれ関節角度の組合せで表現される。

本稿で対象としているロボットは水平5軸マニピュレータである。このうち1つはリストロールのための関節であるので、Motion Planning という観点からは自由度4のマニピュレータとすることができる。そこで、マニピュレータの形状を定めるパラメータとして関節の回転角度 [Waist, Shoulder, Elbow, Wrist] を用いる。

自由空間を導出する際に障害物とロボットの干渉チェックが必要である。干渉チェックではロボットのすべての姿勢について障害物と衝突するかどうかチェックしなければならないため、従来の方法では非効率である。そこで本稿では制約論理プログラミングを用いた簡易コンフィギュレーション空間を用いる。

\*The Cooperative Motion Planning for Solving Spatial Constraints

<sup>†</sup>Toshirou MUKAI, Hayato OHWADA, Fumio MIZOGUCHI

<sup>‡</sup>Faculty of Sci. and Tech., Science University of Tokyo

### 3.1 ロボットのモデル化

本稿で対象としているオブジェクトはロボット・障害物・移動対象ブロックであり、これらは凸多面体で表現される。ロボットはそれぞれのアーム部分を直方体で記述し、アームごとに障害物との衝突チェック（干渉チェック）を行なう。また、障害物・ブロック等も直方体で表現する。

```
space(X, Y, Z, Xp, Yp, Zp, L, []): -L = 1.
space(X, Y, Z, Xp, Yp, Zp, Lp, [[X1, Y1, Z1]|P]): -
  Xn = Xp + L * X1, Yn = Yp + L * Y1,
  Zn = Zp + L * Z1,
  L >= 0, 1 >= L, Ln = Lp + L,
  space(X, Y, Z, Xn, Yn, Zn, Ln, P).
```

space/8 は第7引数に与えられた頂点座標のリストをもとに、オブジェクトの領域を表現する。

また、対象物を移動させる場合には少なくとも次のような過程が必要である。

1. ホームポジションから対象物の位置への移動（接近動作）。
2. 対象物をつかむ（グリップ制御）。
3. 目標座標への移動（離脱・接近動作）。
4. 対象物を置く（グリップ制御）。
5. ホームポジションへの移動（離脱動作）。

グリップ制御部分を除けば、物体への接近・離脱動作を考慮に入れた動作制御が必要である。そこで本稿では物体への接近・離脱の際の経路を自動導出する。マニピュレータ (m,s) を

表 1: 接近・離脱動作コマンド

コマンド	
ma(R, I, G, V)	初期状態 I から目標状態 G へ 接近方向 V で移動。
md(R, I, G, V)	初期状態状態 I から離脱方向 V で 目標状態 G へ移動。
mad(R, I, G, V1, V2)	初期状態状態 I から離脱方向 V で 目標状態 G へ接近方向 V で移動。

指定し、初期状態・目標状態、接近・離脱方向を与えることにより、途中経路を導出し、マニピュレータを制御する。

### 3.2 干渉チェック

明らかに自由空間とわかる部分について干渉チェックを行なうことは非効率的である。本稿では関節角度を分割し、動作領域を関節角度領域の組合せで表現し、3次元的に干渉チェックを行なう。ところでアーム部分の移動領域は非線形領域となるが制約論理プログラミングでは非線形制約は扱うことができないので凸多面体に近似し線形制約で表現する。障害物領域、アームの移動領域はそれぞれ obs/4, arm/5 で線形領域化されこれらの連言により衝突を起こすかどうかをチェックする。

```
obs(R, X, Y, Z): -obs_list(R, List),
  space(X, Y, Z, L).
arm(X, Y, Z, L, fail): -space(X, Y, Z, L).
arm(-, -, -, -, true).
? - obs(m, X, Y, Z), arm(X, Y, Z, List, Ans).
Ans = fail.
***yes***
```

障害物との干渉に直接関係するのは Waist 部分の関節である。ほかの関節 (Shoulder, Elbow, Wrist) がどのような値をとっても、減らすため、アーム長が最大の時に衝突をおこす Waist の領域を求め、この部分についてほかの関節を変化させた場合の干渉チェックを行なう。

### 3.3 経路探索

導出された自由空間において初期状態と目標状態を結ぶ経路を探索する。このとき、ロボットに与えるコマンド数を極力少なくするという観点から、探索の深さが少ないものが望ましい。そこで、経路探索のアルゴリズムとして深さ付き縦型探索によって経路を導出する。

深さ N について縦型探索により探索を行ない、目標状態が含まれているセルまで到達すればそれを解とする。深さ N で目標状態に到達しない場合には探索の深さを N+1 として探索を再帰的に行なうことによりプランを導出する。導出されたプランは実行の際に具体化され、ロボットコマンドとして通信することにより直接制御を行なう。

## 4 まとめ

本稿では自由度 6 の多関節型マニピュレータを対象とし、接近・離脱方向を考慮した障害物回避経路を導出する方法について述べた。3次元的に干渉チェックを行なっているため箱の中のオブジェクトを取り出すような問題にも用いることができる。初期状態・目標状態のいずれかが達成不能な場合に他のマニピュレータを用いることにより空間的な制約を解消している。

接近・離脱方向を考慮することにより、途中経路を導出するだけでなく組み立てたオブジェクトを壊すことなく、自動的に接近・離脱を行なうという組み立て重視の動作計画を行なうことができる。このため Strips 等の AI プランナーによる組み立て作業等への応用が可能である。

## 参考文献

- [1] 向井, 大和田, 溝口: 制約論理プログラミングによるロボット動作計画. 情報処理学会第 46 回全国大会, 1993 年 3 月
- [2] 向井, 大和田, 溝口: 空間制約を用いた協調ロボット動作計画. 人工知能学会第 7 回全国大会, 1993 年 7 月
- [3] Tomas Lozano-Pérez: A simple motion planning algorithm for general robot manipulators. AAAI '86
- [4] Koichi Kondo: Motion Planning with Six Degrees of Freedom by Multistrategic Bidirectional Heuristic Free-Space Enumeration. *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, VOL7, NO.3, 1991