

空間的制約を考慮した協調ロボット動作計画*

3N-6

向井 理朗†, 大和田 勇人†, 溝口 文雄†

東京理科大学 理工学部‡

1 はじめに

マニピュレータの制御を考える場合、知覚からプランニング、障害物を回避するための運動制御を一貫して行なえることが望ましい。本稿では、ブロックワールドを対象とし、初期状態・目標状態を与えることにより、組み立て順序の生成および衝突回避経路の導出・実行を一貫して行なうことを目標としている。

本稿では空間的な制約を次のように定義する。

空間的な制約：

障害物の存在によりマニピュレータの作業空間に存在する作業不可能空間

自由度の少ないシンプルなマニピュレータの場合、障害物の近くにブロックをおくという作業を行なう問題のときには空間的な制約が強過ぎて1台のマニピュレータの作業としては組み立て問題は行なうことができない。そこで本稿ではこの空間的な制約に対し、2台のマニピュレータによる協調処理というアプローチにより制約を緩和し、問題解決を行なう方法を示す。対象とするマニピュレータはX-Y平面に対し自由度1、Y-Z平面に対し自由度3、グリップの自由度1のMovemasterである。

2 協調作業によるアプローチ

本稿で対象としているのは次のようなブロックワールドである。図1に示すような初期状態から目標状態への組み立て問題を行なう。この時の入力としては初期状態・目標状態のブロックの積みかた、初期状態・目標状態の座標であり、出力として組み立て順およびマニピュレータのコマンドを導出する。

本稿の場合、組み立ての際の障害となるのは作業空間内に存在する障害物および移動に関係のないブロックであり、これらにより空間的な制約が発生する。

この空間的な制約を解消するため、2台のマニピュレー

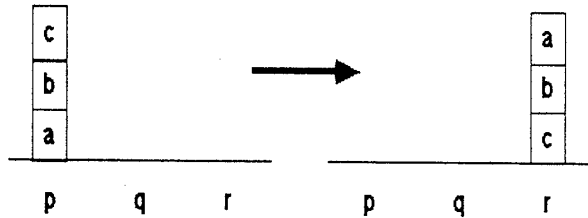


図1: 問題例

タによる協調作業により問題を解決する。図2のように、

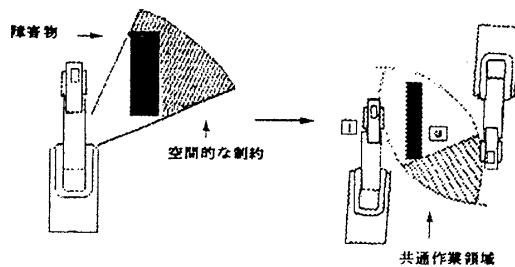


図2: 空間的制約の解消

構造がシンプルなマニピュレータの場合、障害物が存在すると作業空間内に動作不可能空間が生成される。そこで、2台のマニピュレータを用いることにより、この空間的な制約を解消する。

協調処理の際の前提条件としてマニピュレータ同士の衝突に関しては、プラン生成・経路探索の際に同期をとることにより衝突を回避させる。以下その手順を示す。

1. 共通作業領域 (place(q)) の導出
高さ 0
(Movemaster 1 の動作可能領域) \cap (Movemaster 2 の動作可能領域)
2. 組み立てプランに従い、サブゴールまでの経路導出を行なう。
3. 組み立てプランを1つの単位として Movemaster 1, Movemaster 2 の同期をとりながら、プランを実行する。

*The Cooperative Motion Planning for Solving Spatial Constraints

†Toshirou MUKAI, Hayato OHWADA, Fumio MIZOGUCHI

‡Faculty of Sci. and Tech., Science University of Tokyo

3 組み立て順序の生成

初期状態 (place(p))・目標状態 (place(r)) のブロックの積み方および配置座標 (X-Y) をもとに共通作業領域 (place(q)) を導出する。

図 1 の例の場合、次のような連言で初期状態、目標状態が記述される。

前提条件:

block(a), block(b), block(c), place(p), place(q), place(r).

初期状態:

clear(q), clear(r), clear(a), on(a, b, 3), on(b, c, 2), on(c, p, 1).

目標状態:

clear(p), clear(q), clear(c), on(c, b, 3), on(b, a, 2), on(a, r, 1).

この初期状態に対し、目標状態に到達するまで追加リスト・削除リストを繰り返すことにより組み立て手順を導出する。図 1 の場合には

Plan = [pickup(a, b, 3), putdown(a, r, 1), pickup(b, c, 2),
putdown(b, a, 2), pickup(c, p, 1), putdown(c, b, 3)]

というプランが導出される。導出されたプランは place(X) の N 段目から place(Y) の M 段目に移動させるという形に変換される。

2 台のマニピュレータはそれぞれ初期状態の位置から共通作業領域、共通作業領域から目標状態の間の動作を行なう。ここで導出されたプランには初期状態から目標状態へ直接移動させることが考えられるが、この場合には共通作業領域に置き直すことにより作業を分割し、それぞれの経路を導出する。

4 障害物回避経路の導出

ここでは導出された組み立て順に従って組み立てを行なう際の経路を導出する。組み立て動作の際に、障害となるのは作業環境内の障害物のほか、動作に直接は関係のないブロックがあげられる。

以下、動作可能な自由空間 (関節角度の組合せ) を導出する手順を示す。

1. Waist 部分の可動範囲を 4 つに分け、自由空間となる部分を導出する。障害物空間となる部分については次を行なう。
2. Shoulder, Elbow, Wrist の可動範囲を 2 つに分け、それぞれの組合せ (8 通り) について干渉チェックを行ない、自由空間を導出する。
3. 自由空間内に初期状態、目標状態があれば、干渉チェックを終了し経路導出を行なう。
4. 自由空間内に初期状態、目標状態のどちらかがなければ、2 を繰り返す。

ここで得られた自由空間内において初期状態・目標状態を結ぶ経路を導出する。導出されたプランは RS-232C ケーブルをとおして直接実行が可能である [2]。導出されたプランの一部を図 3 に示す。

干渉チェック・経路探索の際には制約論理プログラミングを用いた方法 [1][2] を用いることにより、経路導出の効率化をはかっている。

マニピュレータ 1	マニピュレータ 2
mj(0, -80, 20, 70, 0)	
← place(p)	
gc ← グリップクローズ	
mj(0, 80, -20, -70, 0)	
⋮	
mj(5, -36, 70, 20, 0)	
← place(q)	
go ← グリップオープン	
mj(-5, 36, -70, -20, 0)	mj(18.75, 0, 0, 0, 0)
⋮	mj(15.25, -99, 90, 115, 0)
mj(-18.75, 60, 0, 0)	← place(q)
← ホームポジション	gc ← グリップクローズ
mj(0, -80, 20, 70, 0)	mj(0, 0, 25, -25, 0)
⋮	⋮
← place(p)	mj(1.875, 15, 0, 0, 0)
gc	← place(r)
mj(0, 80, -20, -70, 0)	
⋮	⋮

図 3: 導出されたコマンド (一部)

5 まとめ

本稿では障害物が存在する作業空間内において組み立て順序の生成から障害物回避経路の導出・実行を行なう方法を示した。

協調作業によるアプローチにより障害物により発生する空間的な制約を緩和するとともに、より複雑な作業への拡張が可能である。また、ここでは組み立て順序の生成のため Strips のアルゴリズムを採用しているが、非線形プランナとの融合も可能であり、より複雑な問題への拡張が期待される。

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