二つのマニピュレータのためのスケジューリング†

伊 藤 高 廣*・進 藤 章*

Motion Planning in a Multimanipulator Cooperation System

Takahiro Iro* and Akira SHINDO*

This paper describes a method of motion planning for a dualhand assembly system. Two major possibilities of two-arm robots are as follows:

- (1) coordination between the two arms (for example, assembling bolts and nuts);
- (2) being able to perform two different tasks at the same time. However, there is an increasing degree of teaching complexity when two manipulators must be coordinated in contrast to working independently. Therefore, a plan generator is required to be developed for two-arm robots. In this paper, a knowledge-based approach is introduced to actualize multimanipulator coordination planning.

A data model which is used for generating plans is described. Assembly parts are divided into two categories, "plan" and "world". In describing this structure, a "frames" system is used. "Plan" describes the simple assembly structure in an assembly plan. According to this structure, the sequence of parts-transfer is generated. "World" consists of parts on an actual assembly cell. In these models, the three-dimensional information attached to them such as, the coordinates of points, are represented.

After the sequence of parts-transfer is obtained, it has to be translated into the motions of each manipulator. However, there sometimes is a deadlock problem. This deadlock occurs when each manipulator arrives at the shared space at the same time. To prevent collision, each must wait until the other completes its task and therefore, may be in a complete standstill.

To avoid this deadlock, priority is set to each manipulator in every unit of motion. The assembly is well performed when this pri-

ority is adjusted according to each motion. This is called this "adaptive priority method", which is tested by computer simulation. To get the best priority combination, an optimal-search method is employed. Using this method, the sequence with the shortest total performance time can be selected.

Key Words: multimanipulator coordination, motion planning, frame, deadlock, priority

1. ま え が き

二つのマニピュレータによって(1)ボルトにナットをねじ込む作業のように二つの対象物を同時に扱う協調作業の自動化, (2)それぞれ独立に作業を行う並列動作による作業時間の短縮が可能となる. しかし二つのマニピュレータによる作業内容は一つの場合に比べ複雑になるため人間が教示する負担が増大する. ここでは組み立て作業を対象として, 教示の簡易化を目的とした作業手順の生成, 特に振り分けに重点を置き, その実現法を検討する.

二つのマニピュレータの動作手順生成は過去二, 三の例があるにすぎない. しかも生成した手順の評価は不十分である. Maimon ら¹⁾は治具を共有する二つのマニピュレータのための動作手順生成を行った. 取り扱う部品, 治具までの距離を基に, 「巡回セールスマン問題の解法」²⁾を応用して手順の最適化を行っている. しかし各マニピュレータごとの作業ルートを最適化するととどまり, 全体の作業自体が最短時間で実行できるとは限らなかった. また, 生成が行われているのは手先の動作命令についてであり, 二つのマニピュレータの特質が出るはずの作業の順番は人間が明示的に与えている. 隅田ら³⁾は階層的問題解決法によって二つのマニピュレータの動作手順生成を行った. しかし手順評価については今後の課題としている.

本システムでは, 各マニピュレータの動作を組み立て目標物の設計情報から生成する手法と, その結果の

† 第30回自動制御連合講演会で発表 (1987・10)

* NTT LSI 研究所 厚木市森の里若宮 3-1

* LSI Laboratories, NTT Co., Atsugi

(Received August 21, 1989)

(Revised November 14, 1989)

評価に焦点をあてる．特に多数のマニピュレータで作業を行う場合に特徴的な作業の振り分けすなわち協調動作と並列動作の分割，それぞれの動作における各マニピュレータの作業順序について考える．このため本論文では状況に応じた優先度を各マニピュレータに設定することを提案する．この結果，生成した手順の作業時間最小化が可能となる．

2. 部品移動手順の生成

2.1 手順生成の階層

組立手順の生成を行うためにまず複数マニピュレータによる組立システムを人間による作業からの類推で階層化した．Fig. 1 に概念図を示す．階層は上位から

- (1) plan translator……部品移動手順の生成
- (2) motion distributor……動作の振り分け
- (3) manipulator……動作の実行

である．

plan translator は，組立目標の設計情報に基づき部品の組立順序を決定し，並列動作／協調動作の分割を表現した部品移動手順を生成する．これは部品を主体とした表現であり，マニピュレータとは独立な表現である．motion distributorは，部品とそれを扱うマニピュレータの対応付けを行い，動作手順を生成して各マニピュレータに分配する．manipulator は，動作命令を各マニピュレータ固有の命令に展開し，動作する．本論文が扱うのはこのうちの上位二階層である．

2.2 組立対象

組立対象物の前提として，その場に固定するものではなく組立後移動させるものを取り扱う．以下の説明を容易にするため，組立対象物として Fig. 2 に示す玩具ブロックを例としてとりあげる．このアーチの組立には，

- (1) 土台となる二つのブロックを各マニピュレー

Fig. 1 Hierarchical structure of a coordination model

Fig. 2 Schematic of a product (1)

タがそれぞれ独立に目標位置まで移動させる並列動作

(2) ボルト締めの際，他方のマニピュレータがブロックを押さえて補助する協調動作
という，二つの要素が含まれている．

2.3 手順生成の方法

ふつう設計者は，組立不可能となる事態を避けるために，組立手順を確認しながら図面を書く．この設計者の思考過程が設計図データに反映されていれば，これをもとに設計者が考えた部品移動手順を再現できる．しかも潜在的に設計者の意図が含まれるので複雑な手続きを必要としない．この方法は，設計者が設計図（組立図）の中に，組立の前後関係を表わす部品間の接続情報を付加することで実現できる．

現在一般に使われている CAD では各部品の位置データ，形状データは詳細に記述されるが，組立を考慮した部品間の接続関係のデータはもっていない^{4)~6)}．設計者は組立手順を考慮しながら各部品を集めて組立図を作成したりまた逆に組立図より部品図を作成するが，この作業と同時に部品の接続関係を入力することは比較的容易に可能である．この接続関係をたどれば，組立に必要な部品移動手順を生成することができ．具体的には Fig. 2 に示すアーチの場合，部品間の接続順序は Fig. 3 の矢印で示すようになるであろう．この矢印は設計者の頭の中にある情報であり，現在の CAD システムの中では表現されていない．ここでとりあげる組立手順は，まずこの矢印を情報化し，この矢印の方向をさぐることにより作られる．この方法は一般的に用いることができ，たとえば Fig. 4 のシャコ万力⁷⁾の場合，部品間の関係の一例として，

Fig. 3 Schematic of assembly plan (1)

Fig. 4 Schematic of a product (2)

Fig. 5 Schematic of assembly plan (2)

①本体に②送りネジがねじ込まれ、そこへ③押エ金と④ハンドル棒が接続しさらに⑤ツバが付いていると考えることができる。するとこれは Fig. 5 のように表現され、この矢印を逆にたどり組立手順を得ることができる。

上記の方法を計算機で実行するためには組立目標、部品、動作、スケジュール結果を表現する枠組みが必要である。以下で、この枠組みの設定法、その内容の表現法を述べる。

2.4 組立のフレームによる記述

動作手順自動生成を行おうとすると、多くの環境知識を管理する必要がある。人間が行う作業を機械に代行させる場合、その環境情報管理方法も人間の知識構造に近いものを用いるほうが作りやすく保守しやすい。この意味で静的な階層構造を表現するフレーム⁸⁾が環境知識表現には適していると考え採用した。本システムでは4種類のフレーム(plan, world, motion, gantt)を用いて、それぞれ設計情報を表わす部品、組立に使われる部品、マニピュレータの動作、スケジュール結果を表現する。

Fig. 6 に部品を記述するフレーム間の階層関係を示す。部品は二つの概念(plan, world)より表現される。以下に各部を説明する。

plan は組立図すなわち Fig. 2 で示した組立目標物を表わし、world は実際に組立作業場所にある部品の集合すなわち部品図を表わす。通常、組立システムでは部品の名前は1回定義されるだけである。その場合、同じ種類の部品を何回も使用するとき、組立現場

Fig. 6 Model for assembly components

にある部品の名前に応じて設計情報の名前を変えなければならない。しかし部品を上記 plan と world に分けることにより、同じ設計情報 plan をたとえ別の現場 world でも活用することができるようになる。動作の種類は、扱う部品によって決定される。たとえば対象部品が block であれば move (部品を取りに行き、目標位置へ運ぶ) という動作であり、bolt であれば screw (部品を取りに行き、目標位置へねじ込む) と hold (接続先部品と一緒に回ってしまわないように別の腕で押さえる) という協調動作である。この対応は world の各部品フレーム(block, plate, bolt)で部品の種類を追加登録するときに記入しておくことで対処している。

設計図内の部品を表現した plan フレームの一例として、Fig. 7 に plate-a の内容を示す。図中 Out-of, To, Func の3スロットが手順生成に必要な部品間の接続関係を示すものである。Out-of スロットの値は接続元の部品である。この例では Fig. 3 で矢印の根本にある部品すなわち plate-a に接続する(bolt-a,

Frame name is plate-a

Prototype "plan"	Frame-Fields	From: world
Slot	Data	
World	middle-plate	
Out-Of	(bolt-a bolt-b)	
To	(block-a block-b)	
Func	((-30 0 60 0 0 0) (30 0 60 0 0 0))	
Goal	(-30 0 60 0 90 0)	

Fig. 7 Plate-a frame in "plan"

bolt-b) が入っている。To スロットの値は接続先の部品である。この例の場合は Fig. 3 で矢印の先にある部品、すなわち plate-a から接続する (block-a, block-b) が入っている。Func スロットには To スロットで示す部品への移動ベクトルを入れておく。Fig. 7 の例で図内の数字は plate-a の座標原点から block-a, block-b の座標原点へのベクトルであり、左から xyz 方向の距離 (単位 mm), x, y, z 軸回りの角度 (単位: 度) を示している。本システムでは作業空間、各部品の座標系を Fig. 8, Fig. 9 のように定め、各部品、マニピュレータの位置は作業空間座標で表わす。

部品移動の目標位置は Goal スロット (Fig. 7) に格納する。完成品のどれか一つの部品について作業空間内における組み立て目標位置を与えておけば、あとはシステムが To, Out-of, Func 各スロットの接続情報からすべての部品について目標位置を算出し、各部品の Goal スロットに記入する。

部品間の接続関係を表わす上記 Out-of, To, Func の各スロットの値は現在キーボードから入力している。しかし将来 CAD と生産設備を統合化して、以下に示す機能を付加することにより CAD での簡単な入力が可能となる。

現在の機械 CAD では、部品をソリッドモデルとして扱い、その部品の代表点を指定し、位置を入力することができる。また NC 工作機械用のデータを作成することができる。しかし組立まで考慮した部品間の接続

Fig. 8 Coordinate system of an assembly cell

Fig. 9 Coordinate system of assembly parts

((block-a block-b) plate-a (bolt-a bolt-b))

Fig. 10 Sequence list

情報は、モデルに含まれていない。接続情報入力のためには、ある部品がどの部品に接続しているのかをマウスやキー入力で指定すると、指定した部品の名前が To スロット、Out-of スロットの値として部品フレームの中に記入され、各部品代表点間の距離、部品座標同士の回転角が部品モデルの位置データから計算され Func スロットに記入される機構を付け加えればよい。

上記フレームのスロット値を利用した手順生成の方法はつぎのとおりである。

(1) 土台となる部品の探索

Plan-translator (Fig. 1) は組立目標を構成する部品の中から接続先のない部品を捜す。具体的には Fig. 6 の plan の下の階層にあるすべてのフレームのうち Fig. 7 で示す To スロットに部品名がないものを捜し、みついたら部品名を出力する。

(2) 接続元の探索

(1) でみつけた部品をもとに、接続関係を逆にたどる。すなわち部品の Out-of スロットにある部品名を出力し、さらに出力した部品の Out-of スロットの部品を出力し、これを部品がなくなるまで繰り返す。

この出力結果が Fig. 10 に示す部品移動手順である。これは外側の括弧内の左の部品から順に組み立てればよいことを意味し、最も内側の括弧内の部品は並列に処理してよいことを表わす。また bolt のように他の部品を押えながら作業する必要がある部品フレームの中には協調動作を発生させる指示をあらかじめ記入することとしたので、部品名が協調動作も表わしている。この部品移動手順をもとに Motion-distributor (Fig. 1) がマニピュレータに動作割付を行う。その方法を以下に述べる。

3. 動作手順の生成と最適化

3.1 アダプティブプライオリティ方式の提案

Motion-distributor は、各マニピュレータに対し先に導いた部品移動手順 (Fig. 10) の各部品を割り当て、マニピュレータの動作を決定していく。しかし、動作主が複数の場合には作業の分担、あるいは協力が必要となる。そこでこの問題に対処するため、各マニピュレータへ新たに優先度を設定する。しかし優先度が固定されていると、同等の機能をもつ二つのマニピュレータの柔軟性が生かしきれない。そこで動作ごとに優先度を変化させ、状況に応じた役割分担ができ

ようにする。優先度の割当が決まれば各マニピュレータの待ち時間が決定され、全体の作業時間も決まる。全体の作業時間を最小にする動作手順すなわち最適手順は、優先度の組合せを適切に設定することによって得ることができる。これをここではアダプティブプライオリティ方式と呼ぶことにする。

優先度の設定方法としては、可能な優先度の組合せを各マニピュレータにまず与え、そのなかからこの優先度を基に決定した動作が可能かどうか、予想実行時間が最も短くなるのはどれかという評価基準で選ぶこととする。

3.2 ガントチャートの作成

並行して行われる作業のスケジューリングには一般に、ガントチャート⁹⁾が使われる。本システムにおいてもこれを用い、変化する優先度を新たに導入する。Fig. 2 で示したアーチの組立において、二つのブロックの上に板をのせるところまでをガントチャートで表現したものが Fig. 11 である。図で横軸は時間を示し、縦軸でマニピュレータ、さらに手順の比較を示す。太線で示した動作は優先度が高いことを示す。最初マニピュレータ A のほうが優先度が高く、これが block-a を取りに行き (go), 目標位置へ運ぶ (bring)。この間マニピュレータ B は block-b の用意してあるところまで行き (go) 待機している (wait)。マニピュレータ A の動作が終わるとマニピュレータ B は優先度が高くなり、これが block-b を目標位置まで移動させ (bring) 初期位置に戻る (return)。このときマニピュレータ A は plate-a を用意してあるところへ取りにいき (go), マニピュレータ B が初期位置に戻る間に目標位置へ運ぶ (bring)。

本システムではガントチャートもフレームで表現する。人がガントチャートを作成する場合、装置の数だけ平行線を引き、その上に相互の関連を考慮しながら作業内容を記述し作業時間ごとに区切りをつけていく。同様のことをここでは、フレームを作りスロット

Fig. 11 A priority set for motion distributor (ex. 1)

Frame name is g101

Slot	Data
Priority	((manipulator-a 1) (manipulator-b 0))
Motion (basic)	((manipulator-a move block-a (manipulator-b move block-b))
Motion (detailed)	((manipulator-a move0 block-1) (manipulator-b move2 block-2))
Motion time	14.2[s]
Total time	14.2[s]

Fig. 12 A frame for one movement

値を書き込むことで行う。具体的には一つの作業を一つのフレームで表わし、優先度スロット、動作内容スロット、動作時間スロットに値を入れ、続いて行われる作業はその下の階層のフレームとして表現する。このガントチャートを構成する作業フレームの例が Fig. 12 である。このフレームは最初の二つのブロックの移動を表わしている。各スロットの値の決め方を以下に述べる。

3.3 並列動作におけるスケジューリング

初めに、優先度を各マニピュレータに対して設定し、Fig. 12 の Priority スロットに記入する。マニピュレータが2台なので、(マニピュレータ A, マニピュレータ B) という組合せに対して優先度は (1, 0), (0, 1) (数字の1が優先度が高いとする) の2通りがある。そこで作業フレームもこれに対応して二つの候補が作られる。Fig. 12 は (1, 0) のほうを示している。

二番目に、部品とマニピュレータとの対応付けを行う。部品は、Fig. 10 の部品移動手順ですでに指定されている。まずこの部品移動手順左端カッコ内で、並列移動が可能なが示されている block-a, block-b について扱う。マニピュレータ A が block-a を運び、マニピュレータ B が block-b を運ぶことは、Fig. 13 に示すように組立位置に仮想的な壁を設け、それぞれのマニピュレータ側の空間に存在する部品を割り当てることで決定される。

衝突回避は複数マニピュレータを扱うとき重要な問題の一つである。本システムでは計算を簡単にするため、あらかじめ定められた協調動作以外では共通作業空間に両マニピュレータが同時に入らないようにした。この結果 Fig. 11 のガントチャートで、block-a, block-b の移動では両マニピュレータが同時に move という動作に取り掛かったものの、優先度の低いマニピュレータ B はマニピュレータ A の作業が終わるまで

Fig. 13 Space organization

待つこととなった。現状でも、衝突回避の計算に簡単で高速なものが使用できれば、共通作業空間に同時に部品を運ぶ動作も、ネジ締めのような協調作業と同様の可能な組合せの一つとして設定し、作業の効率化を図ることが可能である。

この Fig. 11 の例では、マニピュレータ A が二つの作業をするので、優先度の組合せは 2^2 で 4 通りとなる。Fig. 11 の優先度組合せは作業ができる場合であるが、Fig. 14 では block-b を運んだ後、運ぶものがないマニピュレータ B の優先度が高いままなのでマニピュレータ A は組立位置へ入ることができず作業が進行不可能となる。よってこの動作手順は捨てることとする。他の優先度組合せについても同様に判断していく。

三番目に設計図部品 (plan) と同型の部品を作業空間内部部品 (world) から選び対応付けし、motion (detailed) スロットに記入する。このとき motion (basic) スロットの動作と優先度から、相手の動作終了待まで含めたより詳細な動作が決定される。この決め方とさらに細かい動作の内容は、move, screw, hold などの motion フレームにあらかじめ記述して

Manipulator	priority	Sequence of motion	Time
A	0 0	go wait move block-a	
B	1 1	go bring move block-b	

Fig. 14 A priority set for motion distributor (ex. 2)

ある。部品の移動距離とマニピュレータの動作速度から作業の動作時間が算出され、これを motion, time スロットに記入する。さらにこれを足し合せてここまでの合計時間を出し、total time スロットに記入する。以上を実行可能なすべての場合について目標物が組み立つまで続ける。

3.4 協調動作におけるスケジューリング

二つのマニピュレータによる組立では、同じくねじを締めるという仕事でも、前に行った仕事の分担、残っている部品の状況、つぎに行う仕事に対する拘束条件によって各マニピュレータの分担のしかた (技術的順序) が変わらう。むしろ状況に応じて仕事の分担を変えうることでフレキシビリティが発揮できる。このように自由度が大きいため定式化しにくい。むしろシミュレーションによって動作時間などの評価値を求め、その中から最適なものを選ぶほうが現実的である。このスケジューリングの解を効率よく求めるために、optimal-search²⁾と逐次枝刈り法を用いる。すなわち、つぎの作業フレームを作成する候補の中から、開始からの作業時間 (total time) の最も短いものを選んで、その結果組立が完成し、かつ他の候補よりまだ作業時間が短ければそれ以上他の候補の計算はしなくてすむというものである。必ず最適解をみつけることができ、極端に作業時間の短い解が存在すれば大幅に探索を減らすことができる。しかしどの候補も同じような作業時間であれば、全数探索になる場合もある。

探索数を減らす方法としてさらに、逐次枝刈りを行う。Fig. 15 に組み立て例のガントチャートを示す。太線が優先度の高い作業を表わし、縦の細線が優先度を設定する作業間の区切りを示す。Fig. 2 の組立て例では、マニピュレータ A が八つの動作を行うので 8 回優先度の設定を行う。そこで $2^8 = 256$ 通りの場合が考えられるが、二つのマニピュレータの協力が必要とする作業 (たとえばネジ締め) では協力する側、される側の 2 通り動作を選べるのでさらに場合が多くな

Fig. 15 The optimal sequence of movements

る。しかしこの中には優先度の設定の仕方によっては実行不可能な動作があり、これはつぎの二つの条件で排除される。

(1) 優先度の高いほうのマニピュレータでも、Fig. 10 の部品移動手順により後で移動すべきことが表現されている部品を前もって扱うことはできない(この条件に反すると、つぎの組立の前提となる部品がないため作業が続けられなくなる)。

(2) 相手のマニピュレータが協力を必要とする作業を行っている場合は、それに協力する動作以外はできない。

3.5 計算機実験

以上の条件で枝刈りを行うと、Fig. 2 に示す物体の場合結局 4 通りの実現可能な組立手順が得られる。この結果を Fig. 16 に示す。図中「g 101」とあるのが、Fig. 12 で説明した作業フレームであり、他も同様にガントチャート内の各作業を表わしている。一統きの作業手順としては、g 101-g 202~g 2601, g 101-g 202~g 2202, g 102~g 2801, g 102~g 2202 の 4 通りがあることを表現している。図では左から右へと時間順を示しており、各手順で右端の作業フレーム (g 2601, g 2202, g 2801, g 2202) の total time スロットの値を比較した場合、最短時間のもの(図中 g 101-g 102~g 2601 で、Fig. 15 に示したものと同一)は、最長のものと比べ約 10% 作業時間が短い。ところが、共有作業空間に最初に入ったものが先に作業を行う方法では、最長時間の作業を求めてしまう可能性がある。またこのスケジュールを計算するのに要した時間

は、計算機として ELIS¹⁰⁾、言語には Common LISP を用い、CPU TIME で 27 秒であった。これは作業前のキャリブレーションや、前の作業の実行中に行うことのできる長さである。

組立の部品数が多くなれば、組立手順の場合の数が一層増大し、計算時間が膨大なものになる恐れがある。この問題についてはつぎの二つの対策が考えられる。

(1) 組立のモジュール化

これは完成品は複数個のモジュールからできており、このモジュールも複数のサブモジュールあるいは部品から成り立っているとみなす方法である。このように階層化することにより各モジュール内の部品数は限定することができ、一度にすべての場合を計算するより短時間で処理できる。本システムで使用しているフレームは元々このような階層構造を表現するのに適しており、実現は容易である。

(2) ヒューリスティックな探索の導入

本論文の例では optimal-search を使っているのですが、必ず最適解が見つかる反面どの解も同様のコストの場合に全数探索となってしまう。一方、ゴールまでのコストを推定し、これが最も小さい枝を優先的に延ばしていく best-first-search の場合は、求めた解が最適解である保証はないものの、他の場合をほとんど調べなくても最適解に近いものはみつかることから、計算時間を短縮するのに有望と考える。しかしゴールまでのコストの推定は、残り部品の点数、形態、組合せから、組立作業や部品固有の知識を使って行う必要があり、この定式化が現状では困難である。ただしこれが解決すれば、どの枝を延ばすかの判断部をこのヒューリスティクスで置き換えるだけで、他は本システムのまま実現できる。

4. む す び

二腕マニピュレータの協調動作を事前に生成する手法を提案した。さらに二腕の動作手順を最適化する手法として、アダプティブプライオリティ方式を提案し、計算機シミュレーションによりその有効性を検証した。

本論文執筆に当たり、ご指導、ご討論いただいた LSI 研究所プロセス自動化研究部西村国俊主幹研究員に感謝いたします。

参 考 文 献

- 1) O. Z. Maimon and S. Y. Nof: Coordination of Robots Sharing Assembly Tasks, Trans. ASME, J. Dyn. Syst. Meas. Control, **107**, p. 299 (1985)

Fig. 16 Four possible sequences of motions of job

- 2) 白井, ほか: 人工知能, 岩波講座 情報科学-22, 岩波書店 (1982)
 - 3) 隅田, ほか: 組立てロボット用動作計画システムの試作 (第2報), 情報処理学会第35回全国大会講演論文集, p. 2045 (1987)
 - 4) SDRC 編: よくわかる I-DEAS (1987)
 - 5) 横河ヒューレットパッカード(株): I-DEAS 資料 (1989)
 - 6) 日本 IBM(株), CATIA 講習会資料 (1984)
 - 7) 吉澤, ほか: JIS 機械製図, 森北出版 (1966)
 - 8) P. H. ウィンストン (長尾, ほか訳): 人工知能, 培風館 (1980)
 - 9) 牧野都治: OR 入門, 数学ライブラリー5, 森北出版 (1968)
 - 10) 竹内郁雄: マルチパラダイム言語 TAO① Lisp マシン ELIS, bit, 20-1, p. 25 (1988)
-