論　文

双腕マニピュレータによる自己組織化マニピュレータの組立作業

正員 福田敏男（名古屋大）
非会員 薛国慶（名古屋大）
非会員 新井史人（名古屋大）
非会員 浅間一（理研）

Assembly Work of Self-Organizing Manipulator Using Dual Manipulator

Toshio Fukuda, Member, Guoqing Xue, Non-member, Fumihito Arai, Non-member (Nagoya University), Hajime Asama, Non-member (The Institute of Physical and Chemical Research)

This paper deals with a kind of cellular robotic system (CEBOT-Cellular Robot), called Self Organizing Manipulator System (SOMS). In this system the configuration of the manipulator can be changed according to the given task. A new type of cells is designed, which the signal bus and power lines are made inside the cell. We use two manipulators to assemble, disassemble, and reassemble cellular manipulator by cooperation of the manipulators. When a task is given, the system analyzes the task and generates the structure of the cellular manipulator. The assembly work of the cellular manipulator is divided into modular subtasks. The system generates the macro command for each subtask based on the database. The collision avoidance is considered for planning the cooperation work by using the collision database which was calculated before hand. The simulation results and the experiments illustrate validity of the proposed system.

キーワード：セルラーマニピュレータ，プランニング，共同作業，衝突回避

1. まえがき

従来のロボットマニピュレータではリンクの長さなどその構造は固定されているため，マニピュレータの動作範囲や自由度は初めから決められており，それぞれのマニピュレータを適用できる作業は限られていた。

こうした問題を解決しようとするものとして，最近知的生産システム（IMS）を念頭に置いたCEBOT（Cellular Robotics）の研究が行われてきた[1]-[4]。その基本概念はシステムをモジュラー構造とし，作業目的に応じてシステムを組替え再構成することにより，一つのシステムで多様な作業にフレキシブルに対応することが可能となる。

これまで試作したCEBOTのセル（MARK I, MARK II, MARK III）では，移動，通信機構を含めて多くの機能をセルにもたせ，完全自律型のセルを目標としているが，物理的，技術上の制限のため実用的なマニピュレータを構成するにはまだ困難である。そこで，本研究ではCEBOTの基本概念に基づいて，その一形態である自己組織化マニピュレータシステムを提案する。本システムで用いたセルは，分離状態での通信機能と移動機能は搭載していないため，セルをより小型化することが可能になった。セルの結合は他のマニピュレータを用いた組立作業により実現する。単台のマニピュレータでは複雑な作業に対して効率は良くないので，本論文では，双腕マニピュレータによるセルの組立作業の計画を行い，作業時間の短縮と作業範囲の拡大を図る。

同じ作業空間のなかに2台以上のマニピュレータが

あるとき，マニピュレータどうしの干渉を考慮する必要がある。ロボットマニピュレータの衝突回避方法は多数提案されているが[5]~[12]，本研究ではリアルタイムの制御性を重視し，干渉データベースを用いて干渉チェックをしながら，複数マニピュレータによる共同作業をプランニングする方法を提案する。それを実験により検証し，その有効性を示す。更に，構成されたセルラマニピュレータの作業実験も行い，このシステムの産業応用への可能性について考察した。

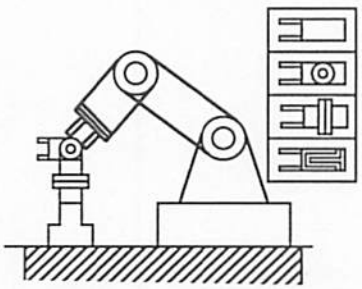

図 1　自己組織化マニピュレータの概念
Fig. 1. The concept of self-organizing manipulator.

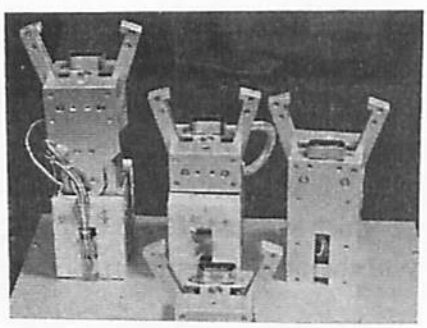

図 2　マニピュレータセル
Fig. 2. Manipulator cells.

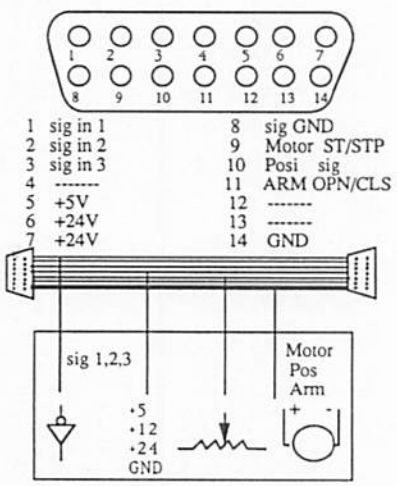

図 3　セルの信号バスと電源線の配置
Fig. 3. The signal and power lines of cell.

2. 自己組織化マニピュレータ

自己組織化マニピュレータとは，作業対象や作業の内容に応じて，ロボットマニピュレータの形状を自律的機能を有する要素（以下，セルと呼ぶ）によって再構成できるようなマニピュレータシステムである。マニピュレータはモジュール構造になっており，各モジュール要素であるセルどうしのいろいろな結合によって，多様な作業に対応することが可能となる。図1に自己組織化マニピュレータの概念を示す。

今回試作したマニピュレータセルは，セルにコネクタをもたせ，電源コードと信号バスをセルの中に収納することにより，セルの組立作業の進行と同時に電源コードと信号バスが，ベースセルからエンドエフェクタセルまで自動的につながっていくような構造をしている。図2に試作した新しいマニピュレータセルの写真を示す。図3にセルの信号バスと電源線の配置を示す。

3. システム構成

本システムは双腕マニピュレータ（Move_Master 五自由度　RBBBR 関節型），ベース，回転，延長，屈曲など4種類のセル，双腕マニピュレータ制御用パーソナルコンピュータ2台と，プランニング用ワークステーション(Sun-sparc) 1台から構成されている。コンピュータ間の通信はソケットベースド通信により行う。図4は本システムのモデルを示す。本システムでは与えられたタスクに対して，マニピュレータの構造を生成し，衝突回避を考慮した双腕マニピュレータの協調作業計画を行う。このときの作業シーケンスはタスクモジュールを用いてモジュール化し，マニピュ

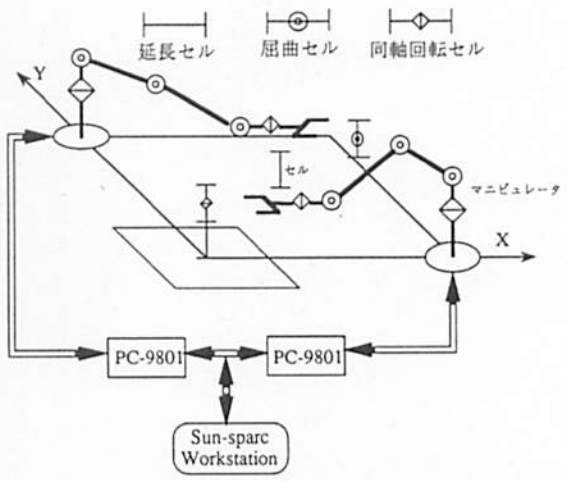

図 4　システムのモデル
Fig. 4. System model.

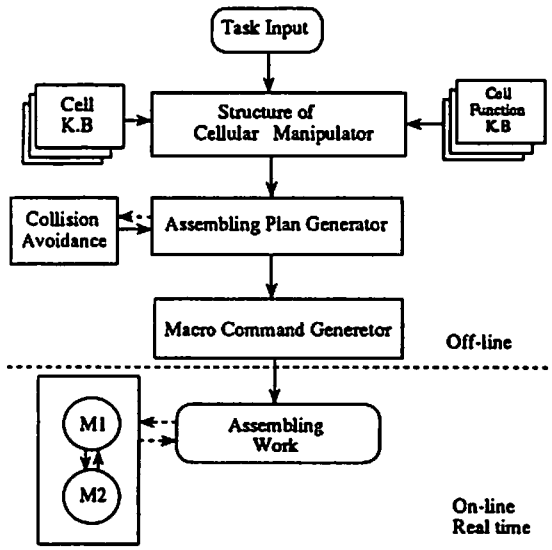

図 5　システムの構造
Fig. 5.　System structure.

レータが実行できるマクロコマンドレベルまで自動展開する。これに基づき，双腕マニピュレータを駆動し，タスクに適するセルラマニピュレータを組立てる。ここで述べるシステム全体の構造を図5に示す。モジュール構造のマニピュレータの構造の自動生成に関しては既に幾つかの研究により行われている[3]。本研究では，セルの組立作業のプランニングについて検討する。

4. 双腕マニピュレータ協調作業による組立タスクプランニング

〈4・1〉 衝突回避に関する基本的考え　　従来よりロボットマニピュレータの障害物回避方法としては，主に障害物の置く場所と形によりコンフィギュレーション空間を計算し，最優パスを探す自由空間方法[5]や，障害物表面で最大となり，目標点で最小となるポテンシャル場を作業空間全体に形成し，そのポテンシャル場から得られる力を用いて目標点までの経路を生成するポテンシャル方法[6][7]などが知られている。しかし，これらの方法はかなりの計算時間を要するため，コンフィギュレーション空間，またはポテンシャル場が常に変わるような双腕マニピュレータの実時間制御には適さず，移動障害物の衝突回避に対する研究は少ない[8]~[12]。従って，マニピュレータのような複雑，しかも常に移動している障害物に対しては，演算時間はかなりかかるため，まだ実用段階にはなっていない。本研究は，あえてパスプランニングを避けて，逆の手順からこの問題を解決しようと試してみたものである。以下はこの手法のポイントを述べる。

（i）　最優パスを探すのではなく，障害物がないとして，各マニピュレータの経路を各タスクモジュールに基づき生成する。そして，干渉データベースを用いて衝突をチェックし，タスクの前後条件を満たし，プランニングにより衝突しないよう，各マニピュレータの待ち時間リストを作る。

（ii）　双腕マニピュレータのベース位置を動かさない限り，干渉する姿勢の組合せと干渉しない姿勢の組合せは決まる。従って，マニピュレータの姿勢を適当に量子化して，あらゆる姿勢の組合せに対して，あらかじめ衝突するかどうかを計算して干渉データベースに入れておく。プランニング時に両マニピュレータの姿勢さえわかれば，データベースを調べることにより干渉しているかどうかがわかる。

（iii）　従来の方法と比較し，本研究で提案する方法は実行に先立ち，前もって作業計画を行うことにより実際の作業動作を高速に行うことができる利点がある。

〈4・2〉 協調作業システム

（1）　協調作業システムの流れ　　協調作業を行うマニピュレータの干渉チェックは，干渉データベースに基づきオフラインで実行される。

作業の計画は以下の流れで進行する。

（i）　組立タスクからサブタスクに展開する。両マニピュレータにサブタスクを配布する

（ii）　データベースに基づき，マクロコマンドを生成する。セグメントタイムごとに両マニピュレータのポジションリストを計算する。

（iii）　双腕マニピュレータの干渉データベースを計算する。

（iv）　干渉データベースに基づき，サブタスク前後条件を満足させ，プランニングにより，各マクロコマンドを実行する前の待ち時間を計算する。次に，マクロコマンドリストに待ち時間リストを加えて，実行コマンドリストを作る。これに基づきマニピュレータを動かす。

（2）　サブタスクの展開と配布　　組立ての順番から，セルのデータベースとセル間の関係によりサブタスクの順番を決める。更に，知識ベースによって両マニピュレータにサブタスクを配布する。そのモデルを図6に示す。

サブタスクはモジュール化している。今回使っているモジュールは以下のように3種類あり，この3種類のサブタスクですべての組立タスクを表示する。

A．　move Cell * to Cell * *
B．　move Cell # to *S_place* #
C．　move Cell # to Cell * *

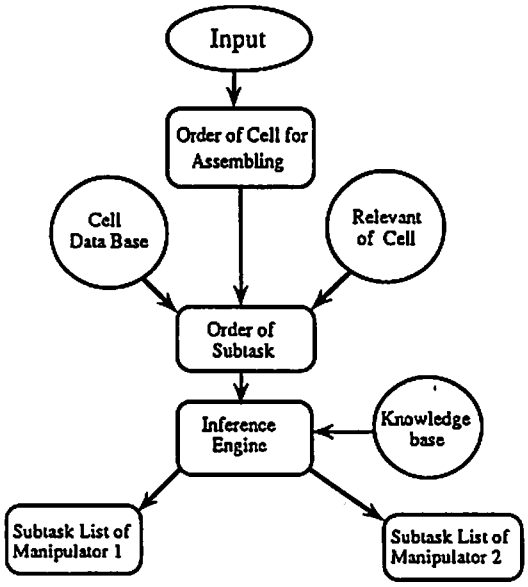

図 6　サブタスクへの展開
Fig. 6.　Expend to subtask.

A は基本サブタスクである。一つのセル＊を他のセル＊＊の上に載せる作業である。

B はセルの置く場所の相互関係により一つのセルを動かさなければ，とりたいセルがとれないことがあるため，邪魔なセルを安全地域（S_place）に退避させる作業である。

C は S_place(安全地域）にある Cell #をあるセルの上に載せる作業である。

ここで推論エンジン（Inference Engine）は以下の基準に基づき判断する。

（i）　どちらのマニピュレータがとりやすいか（近いかどうか，方向が向いているかどうか)。

（ii）　マニピュレータが実行しているサブタスクが次のサブタスクとむだなくつながるかどうか。

（3）　マクロコマンドの生成　　サブタスクが生成されたあと，モジュール化したサブタスクをデータベースに基づき，マクロコマンドに展開する。

例えば，サブタスク A に対して以下のように展開する。

（i）　マニピュレータの先端（以下，先端と略記）を Cell ＊置く場所の真上に移動する。

（ii）　先端をマニピュレータのハンド方向へ距離 D だけ移動する。

（iii）　先端を垂直（Z 軸）方向へ高さ $H1$ だけ移動する。

（iv）　先端をハンドの反対の方向へ距離 D だけ戻す。

（v）　ハンドを閉じる。

（vi）　先端を組立作業位置に移動する。

（vii）　先端を垂直方向へ高さ $H2$ だけ下ろす（合体）。

（viii）　ハンドを開ける。

（ix）　先端をハンドの方向へ距離 D だけ移動する。

（x）　先端を垂直方向へ高さ $H3$ だけ移動する。

ただし，パラメータ $D, H1, H2, H3$ はセルの大きさと構造によって違うので，セルのデータベースに基づき計算する。

サブタスク B，C も同様にマクロコマンドレベルまで展開する。

一つのマクロコマンドの実行時間は

$$T_c = T_s(v) + T_i(v) * L + T_e(v) + T\alpha \cdots (1)$$

T_s：起動時間，v：マニピュレータのスピードレベル，T_i：単位距離移動するのにかかる時間，L：起点と終点間の距離，T_e：制動時間，$T\alpha$：予期しなかった Timeover 分

と仮定している。ここで，$T\alpha$ というのは予想できない時間，例えば，はめ合い失敗するときの繰返し実行の時間など，オフラインプランニングするときは，この成分を考慮しない。これに基づいて両マニピュレータのセグメントタイムごとに位置リストが得られる。従って，両マニピュレータの位置は予想できることになる。

（4）　干渉データベース計算方法　　今回用いた 5 自由度関節型ロボット（RBBBR）について，干渉データベースの計算方法を述べる。

ここで述べる干渉データベースとは，両マニピュレータどうしの干渉状況を記憶するデータベースである。

図 2 からわかるように，今回用いた RBBBR 型ロボットではその幾何学的構造上から，マニピュレータの各リンクの中心線は一つの紙面と垂直する平面内にある。すなわち，上から見れば双腕マニピュレータの中心線は 2 本の線分上に存在する。従って，上から見てこの 2 線分が交差しなければ，少なくとも両マニピュレータはぶつからないと考えられる。

そこで図 7 のように，作業空間の X-Y 平面を $n * n$ 等分し，マニピュレータ 1 の手先がその n^2 個の領域の中の任意の領域にあるとき，マニピュレータ 2 の手先が各領域にあるときマニピュレータどうしの干渉状況をすべて計算する。一つのマニピュレータを 2 本の平行線で表すと，もしある領域で 4 本の線分がそれぞれ交差しなければ干渉は生じないため，干渉データベースに 0 と記入する。そうでなければ 1 と記入する。

上記線分が交差する場合は，すなわち干渉データベ

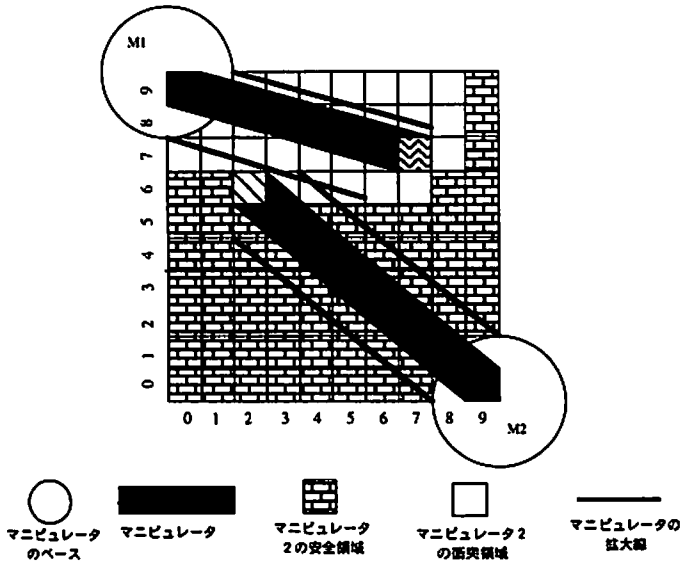

図 7　干渉データベースの一例 (n=10)
Fig. 7. An Example of collision database (n=10).

ースに 1 と書いてある場合，干渉する可能性があると考えられるため，これらの領域について Z 軸方向のチェックを行う。図 8 で示すように，まず両マニピュレータの中心線で構成する平面にマニピュレータを写像する。こうするとマニピュレータをある折線で表現できる。マニピュレータには太さがあるため，この平面に上下二折線で表す。

各リンクの位置は手先の位置と関節の可動範囲で決められており，折線の位置も唯一に決まるものとする。両折線の各リンクがそれぞれ交点をもつかどうかをチェックすることにより，マニピュレータが干渉するかどうかがわかる。

マニピュレータどうしの干渉チェックは具体的に以下の流れで進行する。

〔ステップ 1〕　X-Y 平面の干渉データベースを計算する。

#1.　作業空間の X, Y 軸をそれぞれ n 等分し，作業空間を n 個の領域に分ける。

#2.　マニピュレータ 1 の先端が領域 0 から領域 99 にあるとき，マニピュレータ 2 の先端が領域 0 から領域 99 まで逐次に変更し#3 をする。

#3.　両マニピュレータの先端位置に従って 4 本の線分を決め，それぞれ交差するかどうかを計算する。

#4.　交差するならば，干渉データベースに
$c\_base(P1, P2, 1)$　と記入する。そうでなければ
$c\_base(P1, P2, 0)$　と記入する。
$P1, P2$ はそれぞれマニピュレータ 1 とマニピュレータ 2 が存在する領域の番号。

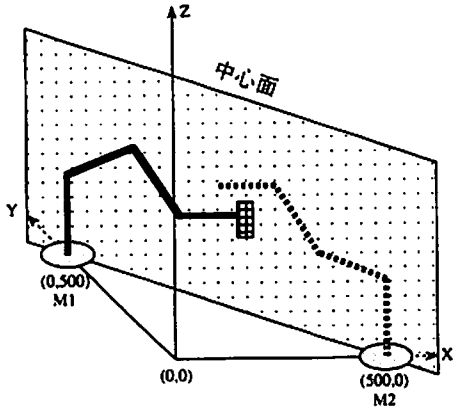

図 8　Z 軸方向でのチェック
Fig. 8. Check in the direction of Z axis.

〔ステップ 2〕　両マニピュレータ座標データ
$X1, Y1, Z1$, alpha 1, beta 1
$X2, Y2, Z2$, alpha 2, beta 2
から $P1, P2$ を計算する。
X, Y, Z はマニピュレータの手先の直角座標。
alpha, beta は手先の姿勢に関する角度。

〔ステップ 3〕　$P1, P2$ で干渉データベースを調べる。

0 であれば，干渉しないと考えられる。この状態の干渉チェックが終り，1 であれば干渉する可能性があるので，〔ステップ 4〕に進行する。

〔ステップ 4〕　両マニピュレータの関節点を決め，ベース軸で構成する平面に写像する。

〔ステップ 5〕　この平面で 4 本折線の位置を決め，それぞれ交差するかどうかを計算する（図 8 参照)。マニピュレータがセルをもっているときは，手先にもう一つリンクを付ける。

〔ステップ 6〕　交差する折線があれば，干渉する。なければ，干渉しないと考えられ，干渉チェックを終了する。

以上により求めた X-Y 平面内での二次元の干渉データベースは，本論文の方法では $n^2 * n^2$ の行列に収容できる。メモリーに余裕がある場合は Z 軸方向の干渉結果もデータベースに入れることにより，あとの計算がより速くなる。しかし，本システムにおいて，マニピュレータどうしの干渉は二次元の干渉が生じたときのみ発生する可能性があるため，今回は二次元の結果のみを干渉データベースに入れておき，干渉チェックのときデータベースから干渉の可能性ありと検出された点だけ Z 軸方向で干渉チェックを行う。このようにすることにより，干渉チェックの高速性を保ちながらメモリーのむだ使い少なくすることができる。

また先端にもう一つ屈曲自由度があるとき（6 自由

度）は，先端にこの自由度で動ける範囲を一つの球とし，この球をマニピュレータの一部とすることにより一つの自由度を消して，上述の方法が使えるようになる。

〈4・3〉 コマンドの待ち時間の決定方法　双腕マニピュレータをスムーズに協調作業させるには，サブタスクの前後条件と衝突しない条件を満たさなければならない。

マニピュレータ1と2の時間リストはそれぞれ以下のように表す。

マニピュレータ1：

$$Tw_{11}^{1}\rightarrow Ct_{11}^{1}\rightarrow Tw_{12}^{1}\rightarrow Ct_{12}^{1}\rightarrow\cdots Tw_{1n_1}^{1}\rightarrow Ct_{1n_1}^{1}$$
$$\cdots Tw_{21}^{1}\rightarrow$$
$$Ct_{21}^{1}\rightarrow\cdots Ct_{2n_2}^{1}\cdots Ct_{m1}^{1}\rightarrow Tw_{m2}^{1}\cdots Tw_{mn_m}^{1}\rightarrow Ct_{mn_m}^{1}$$

マニピュレータ2：

$$Tw_{11}^{2}\rightarrow Ct_{11}^{2}\cdots\rightarrow\cdots\rightarrow\cdots\rightarrow Tw_{m'n'_{m'}}^{2}\rightarrow Ct_{m'n'_{m'}}^{2}$$

ただし，Ct_{ij}^{k}, Tw_{ij}^{k} において，C_t：コマンドを実行にかかる時間，Tw：C_t と対応するコマンドを実行する前の待ち時間，k：マニピュレータの番号，i：サブタスクの番号，j：コマンドの番号

である。

作業シーケンスの条件例（マニピュレータ1の r 番目のコマンドが同2の r' 番目のコマンドが完了した後で実行しなければならない）

$$\sum_{i=1,j=1}^{i=mr,j=n_r}(Ct_{ij}^{1}+Tw_{ij}^{1})>\sum_{i=1,j=1}^{i=mr',j=n'r}(Ct_{ij}^{2}+Tw_{ij}^{2})$$

干渉しない条件

$$C\_base\{P_1(T_1), P_2T_2)\}=0 \quad When \quad T_1=T_2$$

2種類の条件のうちのいずれかが満たさない場合，両マニピュレータのうち1台のマニピュレータのコマンドリストに，以下に示すルールに従い待ち時間を入れる。

待ち時間の配分ルール

If Condition=False and

$$T_1^P+T_1^f\leq T_2^P+T_2^f$$

Then

$$Tw_{ij}^{1}=Tw_{ij}^{1}+\text{Segement\_time}$$

Else

$$Tw_{ij}^{2}=Tw_{ij}^{2}+\text{Segement\_time}$$

ここで，

$$T_1^P=\sum_{i=1,j=1}^{i=K,j=NKr}(Ct_{ij}^{1}+Tw_{ij}^{1})$$

$$T_1^f=\sum_{i=K,j=NKr+1}^{i=m,j=n_t}Ct_{ij}^{1}$$

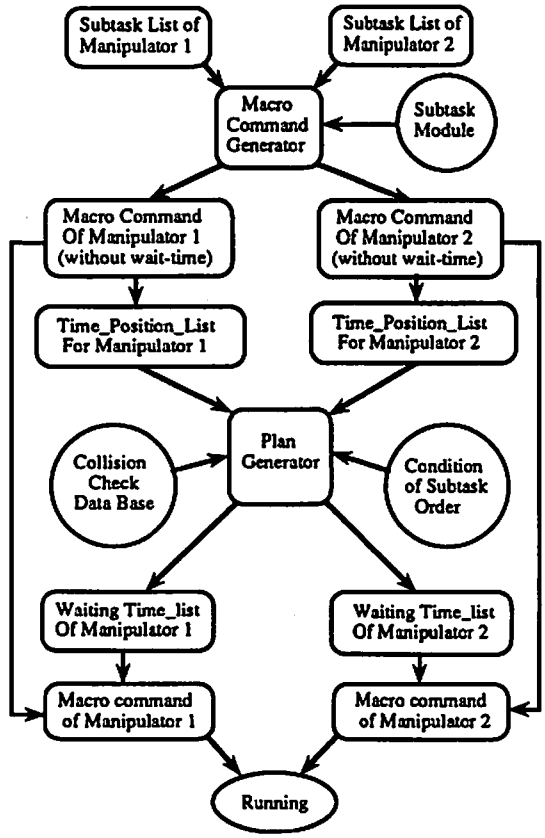

図 9　待ち時間リストの決定

Fig. 9.　Determine the waiting time list.

$$T_2^P=\sum_{i=1,j=1}^{i=K',j=N'Kr}(Ct_{ij}^{2}+Tw_{ij}^{2})$$

$$T_2^f=\sum_{i=K',j=N'Kr+1}^{i=m',j=n't}Ct_{ij}^{2}$$

プランニングの目標は

$$\max\left(\sum_{i=1,j=1}^{i=m,j=n_t}(Ct_{ij}^{1}+Tw_{ij}^{1}),\ \sum_{i=1,j=1}^{i=m',j=n't}(ct_{ij}^{2}+Tw_{ij}^{2})\right) \quad \cdots\cdots\cdots\cdots(2)$$

を最小にすることである。このため，両マニピュレータの作業時間のうち，長いほうの作業時間を短くすることにより両者のバランスをとり，全体の作業時間を短くする。図9にプランニングの流れを示す。

5. プランニングと実験の結果

〈5・1〉 プランニングの結果　以上述べた考えに基づき実験を行った。プランニングの使用言語はK_prolog，マニピュレータを制御するにはC言語を使った。

以下，一つの対象物をStart_pointからEnd_pointまで姿勢制限なしに動かすという極めて単純なタスクに対するプランニング結果を図10に示す。このときのセルの識別番号を表1に示す。

```
Input_task=transfer([0,200,150],[*,*,*],[180,0,170],[*,*,*],pen)
N=3
Cell_structure=[401,201,202]
sub_cmd(Lst)=sub_cmd([[move,201,to_s_place],[move,401,to_base],[move,
201,401],[move,202,201])                                    ..... (1)
cmd_1(L1)=cmd_1([[move,201,to_s_place],[move,201,401]])     .....(2)
cmd_2(L2)=cmd_2([[move,401,to_base],[move,202,201]])        .....(3)
Comm1=[[[mp],[-456.6,-166.2,220,0,0]],[[he],[110]],[[mt],[110,-25]], .....
[[dw],[0,0,120]                                             .....(4)
Comm2=[[[mp],[409.0,258.5,170,0,0]],[[he],[110]], .....     .....(5)
T_list1=[1.3,0.1,0.4, .....,0.7]                            .....(6)
Pos_list1=[[0,0,500],[-456.6,-166.2,220], ... , [-4.3,449.0,380]]  .....(7)
T_list2=[1.4,0.1,0.3, .... , 0.7]                           .....(8)
Post_list2=[[0,0,500],[409.0,258.5,170], .... , [3.1,456.2,520]]   .....(9)
Tim_pos(1,[[[0,500.0,500],[35.1,512.8,471.5], .... , [4.3,51.0,380]]])  .....(10)
Tim_pos(2,[[[500.0,0.0,500],[481.5,54.3,471.5], ... ,[43.8,3.1,520]]])  .....(11)
W_list1=t_base(1,[0,0, .... , 0.5, ...., 4.8, ... , 0,0])  .....(12)
W_list2=t_base(2,[5.4,0, ..... ,0,0])                       .....(13)
Command_1=[[[mp],[-456.6,166.2,220]], ... , [[ti],[5]], ... [[ti],[5], ...
[[dw],[0,0,120]]]                                           ....(14)
Command_2=[[[ti],[54]],....., [[dw], [0,0,120]]]           ....(15)
```

図 10　プランニングの結果
Fig. 10.　Result of planning.

表 1　セルの認識番号とその機能
Table 1.　The recognition number and the function of cells.

番　号	機　能
100	移動セル
200	屈曲セル
300	延長セル
400	回転セル
500	スライディングセル
1000	エンドエフェクタセル
2000	分岐セル

このタスクに対して使うセルは三つ，順番は RBB, Subtask は図 10 の(1)という結果が得られた。

(2)，(3)は各マニピュレータのサブタスク列
(4)，(5)はマニピュレータのマクロコマンド列
(6)，(7)はマニピュレータ 1 のコマンド時間リストとコマンド実行する前の位置リスト
(8)，(9)はマニピュレータ 2 のコマンド時間リストとコマンド実行する前の位置リスト
(10)，(11)はマニピュレータセグメント時間ごとの位置リスト
(12)，(13)はマニピュレータの各コマンドを実行する前の待ち時間リスト
(14)，(15)はマニピュレータの待ち時間を含むマクロコマンドのリスト

となっている。

これで双腕マニピュレータの待ち時間を含む実行コマンド列 Command_1，Command_2 を得た。

〈5・2〉 **実験と考察**　プランニングの結果で実際に組立，分解実験を行い，3 個のセルを組立て，アルゴリズムの有効性を示した。実験のとき(1)式中の $T\alpha$(予期せぬタイムオーバ分) が生じる可能性がある。これを考慮して，一つ一つのマニピュレータのコマンド実行終り次第に両パソコン通信により完成信号をお互いに通告され。もし $T\alpha$ が生じれば，次のコマンド前の待ち時間リストに入れる。生じなければそのまま次のコマンドに入る。これによりシミュレーションで得られた結果をリアルタイムで使えることになった。

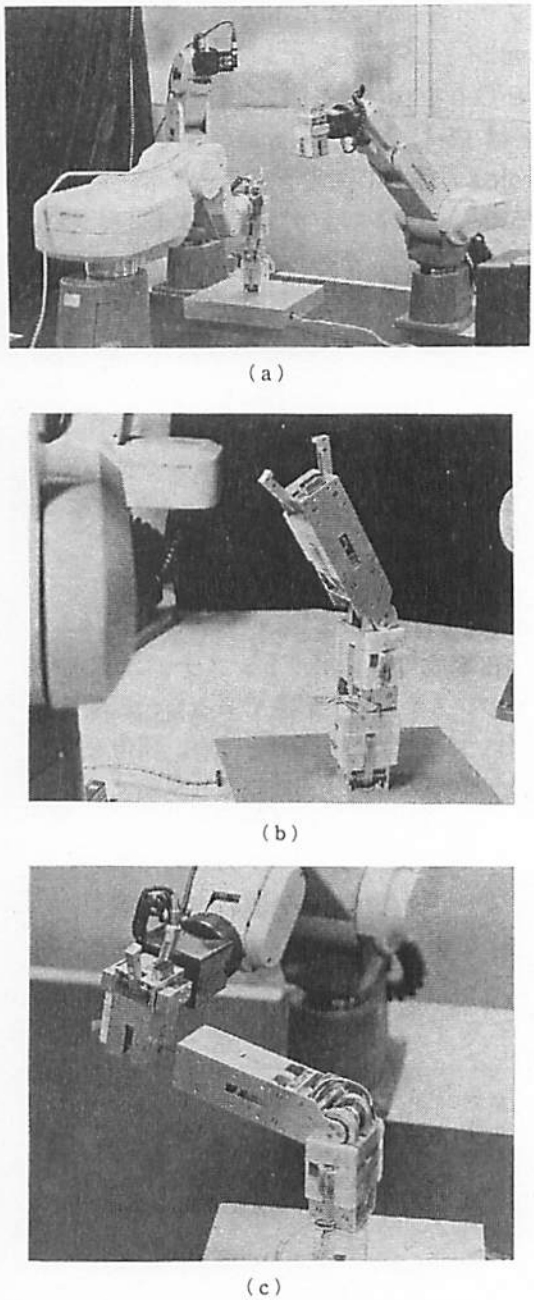

図 11　実　験
Fig. 11.　Experiment.

分解は組立ての逆作業とし，組立作業中にマニピュレータの各通過点の位置と姿勢をロボットに覚えさせることにより，逆再生の方法で分解も実現できた。

図 11 に実験の様子を示す。(a)図：セルの組立過程，(b)図：でき上がったセルラマニピュレータが作業をしているところ，(c)図：ドライバセルを含む

セルラマニピュレータの作業の一例。

今回シミュレーションと実験で，四つのセル（Base Cellを含む）を組立し，一つのマニピュレータを構成した。Sun_sparc Work station上でK_prolog言語で二次元干渉データベースを計算するには620 s，プランニングするには132 s，組立するには34 sを要した。そのうちデータベースの計算は1回しか要しない。また，双腕マニピュレータを使うためより複雑な組立，分解作業が実行可能になった。

また，構成したセルラマニピュレータの先端に各種エンドエフェクタセルを付けると様々な作業できる。今回はドライバセルとグリッパセルを使って実験を行って〔図11(c)〕，セルラマニピュレータにより作業する可能性を確認した。

6. むすび

動的再構成可能ロボットシステムの実用化を目指し，そのシステムの一形態である自己組織化マニピュレータについて以下の研究を行い，報告を行った。

（1） 干渉データベースによる双腕マニピュレータの衝突回避

（2） 双腕マニピュレータの共同作業によるマニピュレータの組立と分解の作業計画

（3） 実験によるアルゴリズムの有効性の検証

（平成4年9月11日受付）

文　献

（1） 福田・中川：電学論C，**107**，1019（昭62-11）
（2） 小野・福田，他：ロボメカ講演会予稿集，p. 67（平2）
（3） 福田・中川，他：機械学論C，**55**，No. 516（平元-8）
（4） 福田・植山，他：第29回SICE学術講演会（平2）
（5） 音田・長谷川，他：第7回日本ロボット学会学術講演会，p. 755（平元）
（6） 奥富・森：ロボット学誌，**1**，No. 3，226（昭58）
（7） 川村・岡田，他：第7回日本ロボット学会学術講演会，p. 365（平元）
（8） P. tournassoud：IEEE Int. Conf. on Robotics and Automation, p. 1224（1986）
（9） スパルーク・鈴木，他：第7回日本ロボット学会学術講演会，p. 361（平元）
（10） K. Kant：IEEE Int. Conf. on Robotics and Automation, p. 1644（1986）
（11） S. Fortune, G. Wilfong & C. Yap：*ibid.*, p. 1216（1986）
（12） Y. P. Chien, A. J. Koivo & B. H. Lee：*ibid.*, p. 209（1989）
（13） 福田・薛：第8回日本ロボット学会学術講演会，p. 903（平2）

福田敏男（正員）

昭和23年12月12日生。46年早稲田大学理工学部機械工卒業。52年東京大学大学院博士課程修了。この間，48～50年アメリカ・エール大学大学院留学。52年通産省工業技術院機械技術研究所研究院主任研究官を経て，57年東京理科大学工学部機械工学科講師，58年同助教授。54～56年西ドイツ・シュツットガルト大学客員研究員。61年アメリカ・エール大学客員助教授。平成元年4月名古屋大学工学部機械工学部第2学科教授。マイクロロボット，特殊環境下のロボット，自己組織化ロボットの研究に従事。工学博士。IEEE Robotics & Automation Society理事（1988～），IEEE Industrial Electronics Society Vice President（1990～）IEEE Neual Networks Council Secretary（1992～），第1回IEEE International Workshop on Intelligent Robots & System（IROS）の実行委員長（1988）。IJCNN '91-シンガポールのプログラム委員長。

薛　国　慶（非会員）

1963年10月1日生。1985年中国上海交通大学工学部船舶工学科卒業。同年同大学勤務。平成2年名古屋大学工学部機械工学科博士課程入学，4年同前期課程修了，同年後期課程進学。自己組織化ロボット，複数マニピュレータの協調作業の研究に従事。IEEE，日本ロボット学会会員。

新井史人（非会員）

昭和38年生。63年東京理科大学大学院修士課程修了。同年富士フイルム入社。平成元年名古屋大学工学部機械工学科第2学科助手。マイクロマシン，柔軟構造物の振動制御，自己組織化ロボット，画像認識，ヒューマンインタフェイスなどの研究に従事。IEEE，日本機械学会，計測制御学会，日本ロボット学会会員。

浅間　一（非会員）

昭和34年1月18日生。59年東京大学大学院工学系研究科精密機械工学修士課程修了。61年理化学研究所化学工学研究室研究員補，現在同研究員。工学博士。分散知能ロボットシステム，バイオプロセスの知能化技術の開発に従事。IEEE，精密工学会，日本機械学会，化学工学会会員。