

## 第5章

# 協調荷物搬送問題での協調手順の探索

複数ロボットで複数のサブタスクからなるタスクを実行するには、サブタスクの実行順序を決定する必要がある。これは一般的なプランニングに共通する問題である。述語論理などのシンボルを用いたプランニングでは、シンボルを用いてサブタスク同士の相互関係や制約条件を記述する。しかしロボットや環境の3次元モデルを利用したプランニングでは、制約条件を明示的に記述しなくとも、モデルと物理的条件を考慮することで自動的に制約条件が導き出せる。これは例えば2台以上のロボットが同時に同じ場所に存在できないことや遠く離れたロボットに荷物を受け渡すことはできないなどの物理的制約である。

前章では、提案手法を用いて協調荷物搬送問題を解決するアルゴリズムを示した。このアルゴリズムではロボット同士の荷物受け渡しのためにサブゴールを自動的に決定する。しかし前章のアルゴリズムでは、入力として協調手順（荷物の受け渡し順序）が与えられると仮定した。ロボットの操作者にとって、この受け渡し順序を指定することは煩雑である。

本章では、ロボットや環境の3次元モデルを利用し、受け渡し順序を自動的に探索するアルゴリズムを実現する。さらにロボットが他のロボットに荷物を受け渡せるかどうかをも自動的に探索するアルゴリズムも示す。これらアルゴリズムには経路プランニングアルゴリズムを統合しているため、必要とする情報は少なくすむ。これにより、ロボット操作をより少ない情報で実現するための基礎となる。シミュレーション実験を行い、これらアルゴリズムの有効性を示す。

### 5.1 受け渡し手順の探索問題

#### 5.1.1 ロボットの作業環境

本章で想定する環境は以下のとおりである。環境中にロボットが複数存在し、ロボットはそれぞれ荷物をスタート位置からゴール位置まで運ぶことができる。ロボット同士があ

る程度まで近づくと荷物を受け渡すことができる。環境中には壁があり、ロボットは乗り越えることができない。ただし、壁には低くなっている箇所（以下では、「窓」と呼ぶ）がある。窓の場所を使って、ロボットは荷物を別の部屋のロボットに受け渡すことができる。環境とロボットの3次元モデル、初期位置、荷物のゴール位置は既知である。ただしどの場所に窓があるかという情報は明示的には与えないが、経路プランニング中に環境のモデルと衝突チェックをすることで自動的に認識される。

これは前章で扱ったタスクとほぼ同じである。ただし、前章ではロボットからロボットへの荷物受け渡し手順をあらかじめ指定していた。本章では、この手順の探索を目的とする。

### 5.1.2 タスク

荷物をスタート位置からゴール位置まで運ぶような荷物受け渡しの手順を探索する。スタート位置で荷物を持っているロボットと、荷物を置くべきゴール位置は指定する。ただし、ゴール位置まで荷物を運ぶべきロボットは指定せず、自動的に探索する。荷物を持ったロボットはほかのロボットに荷物を受け渡すか、もしくは荷物のゴール位置にいるならば荷物を置くことができる。簡単のため、スタートからゴールまで荷物を受け渡す中で、同じロボットに2回以上受け渡さないとする。各ロボットは障害物やほかのロボットに衝突しない限り自由に移動できる（ホロノミック環境である）。

このような条件を満たす経路は、各ロボットをノードとする有向グラフで表現できる。このグラフは最初に荷物を持つロボットを始点とし、ゴール位置に荷物を置くロボットを終点とする。また、設定した条件（同じロボットに2回以上受け渡さない）から、この経路はループを持たない。

### 5.1.3 提案アルゴリズムの利点

以下で示すアルゴリズムを用いることで得られる結果を例示する。図5.1(a),5.1(b)のような環境を考える。この二つの環境は似ているが、図5.1(b)では中央の「窓(window)」の場所で荷物を受け渡す必要がある。

環境が異なるだけでほかは同じ情報を与えた場合に、提案アルゴリズムで得られる結果が、それぞれ図5.1(c), 5.1(d)である。図5.1(c)では、ロボット#1が直接ゴールまで荷物を運ぶ。図5.1(d)では窓があるので、荷物をいったんロボット#1からロボット#2に受け渡し、ロボット#2がゴールまで運ぶという経路が自動的に生成される。

このように環境の情報を変更するだけで、物理的制約を考慮し自動的に適切な経路が生成される。これが提案アルゴリズムの利点である。

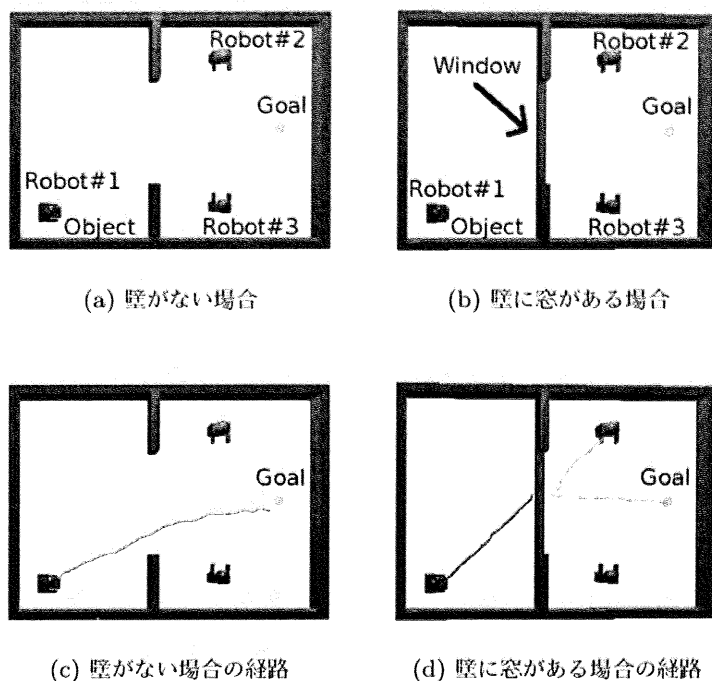


図 5.1 環境が異なる場合に得られる経路の違い

### 5.1.4 タスクの解の性質

このタスクでは解である受け渡し手順の最適性を追求しないものとする。そのため上記の条件を満たす解の一つでも見つければ良い。例えば、ロボットによる受け渡しの総時間が最小となる最適な受け渡し手順を探索することは難しい。なぜなら、どのロボットから荷物を受け渡されるかによってロボットの移動距離が変わるからである。例えば、ロボット A → ロボット B → ロボット C と受け渡す場合と、ロボット D → ロボット B → ロボット C と受け渡す場合では、ロボット B → ロボット C の移動距離はそれぞれ違う場合がある。これはそれぞれの経路でロボット B が荷物を受け取る時に、同じ場所で受け取るとは限らないからである。

また、たとえ最短時間となる経路を求められたとしても、その経路を実行する時間が最短とは限らない。なぜならば、あるロボットが受け渡しを行っているときに他のロボットは次の受け渡し位置まであらかじめ移動できる。それによって、経路で見た場合に最短時間でなくとも、実行時間は短いことがある。このため、最適解を求めるにはすべての手順に対して、すべての経路をプランする以外にない。これは計算量が非常に多い。そのため本研究では最適解は追究しない。

## 5.2 受け渡し手順の探索アルゴリズム

### 5.2.1 探索の方針

最初に荷物を持っているロボットを開始点として、次に受け渡すロボットを探索し、コストをもとに最良優先探索を行う。コストには、各ロボットからゴール位置までの直線距離を用いる。一度受け渡したロボットには二度と受け渡さないで、ループのない順序を探索すればよい。このため動的な環境では解が見つからない可能性がある。

提案アルゴリズムではスタート位置とゴール位置から双方向探索を行うことで探索効率を挙げている。ゴール位置からの逆向き探索では、ゴール位置に到達できるロボットを探索の開始点とする。そのときコストには、ロボットからスタート位置までの直線距離を用いる。スタート位置からの探索とゴール位置からの探索のそれぞれで得られた手順を連結できるとき、連結した手順が求める解となる。ただし、連結して得られる手順にはループを含む場合（同じロボットが2回以上現れる場合）があり、それは解から除く。

この問題を解くには、荷物をゴールに置くロボットを決定する必要がある。それには各ロボットがゴール位置に到達できるかを検査すればよい。この検査はゴール位置が違えば結果も異なるため、新たな探索のたびに検査する必要がある。

上記の探索は、受け渡し関係（すなわち、あるロボットから別のロボットまで受け渡しできるかどうかの関係）が既知の場合と未知の場合に分けられる。受け渡し関係が未知の場合には、手順の探索中に、受け渡し関係を3次元モデルを用いて検査する。

以下の節では受け渡し関係が既知の場合と未知の場合の2通りのアルゴリズムを示す。アルゴリズム中では経路プランニングを利用する。経路プランニングを利用する関数は  $\text{IsReachableToGoal}(r)$  と  $\text{IsPassable}(r_1, r_2)$  である。前者はあるロボット  $r$  がゴール位置に到達できるかどうか判定する関数であり、後者はあるロボット  $r_1$  から別のあるロボット  $r_2$  へ荷物を受け渡せるか判定する関数である。これら関数の実装については5.3章で示す。

### 5.2.2 タスクの計算量

探索中に利用する経路プランニングの計算量を一定と仮定すると、このタスクの計算量は受け渡し手順の探索数で測ることができる。

受け渡しの組み合わせが未知の場合には、受け渡し手順の総数はロボットの順列を数え上げと同じになる。双方向探索を行う場合には、探索すべき経路長は最短経路長  $l$  の半分程度になる。ロボット数を  $n_r$  とすると、 $n_r - 1$  個（最初のロボットは指定されるため  $n_r - 1$  となる）の中から  $\frac{l}{2}$  個取り出す順列の数となるので、その数は  $\frac{(n_r - 1)!}{(n_r - 1 - l/2)!}$  となる。さらにスタート側とゴール側から同時に探索するので探索ノード数は2倍となる。す

なわち計算量はおよそ

$$O\left(2 \times \frac{(n_r - 1)!}{(n_r - 1 - l/2)!}\right) \quad (5.1)$$

となる。

受け渡しの組み合わせが分かっているならば、その受け渡しが行えるペアの総数を  $n_{\text{pair}}$  とすると、経路の平均分岐数  $\alpha = n_{\text{pair}}/n_r$  が計算できる。最短パス長が  $l$  であれば、探索の計算量はおよそ  $O(\alpha^l)$  となる（ただしこれにはループも含まれる）。双方向探索を行う場合には、半分の経路長で、2 倍のノード数を探索するので、

$$O(2\alpha^{l/2}) \quad (5.2)$$

程度の計算量と推定できる。

### 5.2.3 受け渡し関係が既知の場合

アルゴリズムの説明を簡単にするために、以下では通常の最良優先探索を用いたアルゴリズムを示す。しかし後述の実験では、探索性能を向上させるため、初期状態とゴール状態の両方から最良優先探索を行う双方向探索を用いた。

説明のため、スタート位置で荷物を持っているロボットを  $r_s$ 、ゴール位置を表現するためのインデックスを  $r_g$  とする。環境中にあるロボットの集合を  $R$  で表現する。環境中のロボットは  $r_i$  ( $i = 1, 2, \dots, n$ ) と表現する。受け渡し関係が既知なので、ロボット  $r_i$  から受け渡せるロボットの集合を返すルーチンとして  $\text{PassableRobots}(r_i)$  を定義する。

探索中の経路は  $r_s \rightarrow r_1 \rightarrow r_2 \rightarrow \dots \rightarrow r_i$  のように表現できる。ゴールに達したときの経路は  $r_s \rightarrow \dots \rightarrow r_g$  となる。

経路の先頭ロボットを返す関数を  $\text{FirstRobot}(\text{path})$  とし、二番目のロボットを返す関数を  $\text{SecondRobot}(\text{path})$ 、最後から二番目のロボットを返す関数を  $\text{SecondLastRobot}(\text{path})$ 、そして最終ロボットを返す関数を  $\text{LastRobot}(\text{path})$  とする。例えば、 $\text{path} = r_i \rightarrow r_j \rightarrow \dots \rightarrow r_k \rightarrow r_l$  のとき、 $\text{FirstRobot}(\text{path})$ 、 $\text{SecondRobot}(\text{path})$ 、 $\text{SecondLastRobot}(\text{path})$ 、 $\text{LastRobot}(\text{path})$  はそれぞれ  $r_i$ 、 $r_j$ 、 $r_k$ 、 $r_l$  である。

探索中の経路を保持するために、優先順位キューを用いる。優先順位キューはコスト関数に基づいてもっともコストの小さい経路を返す。スタート位置から探索を行うときのキュー  $\text{queue}_s$  では、経路中最後のロボットからゴール位置までの直線距離をコスト関数とする。

受け渡し順序探索アルゴリズムは以下のように実現される。

1. データを初期化する。（優先順位キュー  $\text{queue}_s$  を空にする。すべてのロボット  $r_i$  ( $r_i \in \text{PassableRobots}(r_s)$ ) について、キュー  $\text{queue}_s$  に初期経路 " $r_s \rightarrow r_i$ " を入れる。）
2. キュー  $\text{queue}_s$  から最小コストである経路を取り出し、 $\text{path}_s^c$  とする。その経路  $\text{path}_s^c$  の最後のロボットを  $r_c$  とする ( $r_c \leftarrow \text{LastRobot}(\text{path}_s^c)$ )。

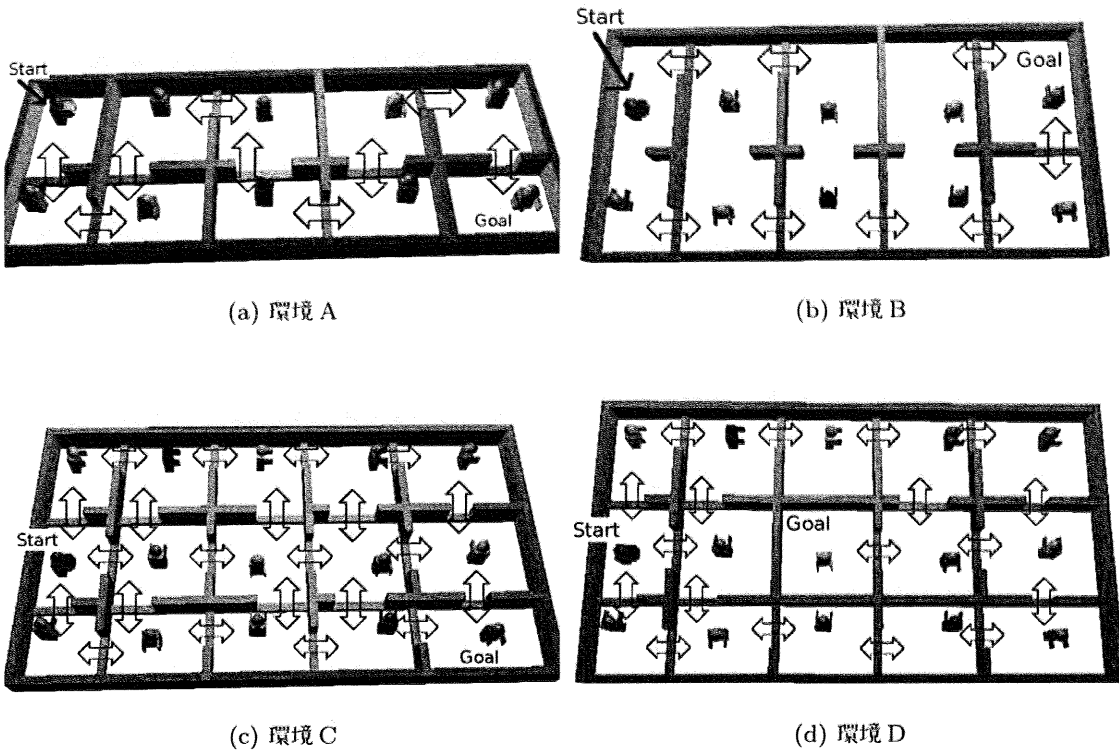


図 5.2 実験に用いた環境モデル (矢印は「窓」の部分を示している.)

3. ロボット  $r_c$  が物体のゴール位置に到達できるかを検査する ( $\text{IsReachableToGoal}(r_c)$ ).  
もし  $\text{IsReachableToGoal}(r_c) = \text{true}$  ならば, 経路  $\text{path}_s^c$  を結果として返し, 終了する.
4. まだ物体を受け取ったことのないすべてのロボット  $r_j$  ( $r_j \in \text{PassableRobots}(r_c)$ ) について新たな経路  $\text{path}_s^j$  を生成する. 経路  $\text{path}_s^j$  とは経路  $\text{path}_s^c$  の最後にロボット  $r_c$  からロボット  $r_j$  に受け渡す手順を追加したものである. すなわち,  $\text{path}_s^j = \text{concatenate}(\text{path}_s^c, "r_c \rightarrow r_j")$  である. そして, 経路  $\text{path}_s^j$  をキュー  $\text{queue}_s$  に入れる.
5. ステップ 2. に戻る.

シミュレーションのため, 4つの異なる環境 (図 5.2 参照) を用いた. これらの環境には多くの壁と「窓」(図中に両矢印で示した) がある. ロボットはそれぞれ別々の部屋に配置されている. 特に, 問題 B では, ロボット 2 台が配置された部屋がある. これらのロボットは同じ部屋内で互いに受け渡し可能である. しかし, それはタスク実行において非効率的な受け渡しである. このような無駄を含まない適切な経路を見つけなければならない.

表 5.1 は問題ごとに探索時に訪問したノード数 (探索ノード数) と成功率を示している. この探索ノード数は双方向探索によるものである. 表中の理論計算量は式 (5.2) で求めた

表 5.1 探索ノード数 (受け渡し関係が既知の場合)

問題番号	A	B	C	D
平均分岐数 $\alpha$	1.8	4.2	3	2.46
最短経路長 $l$	9	4	5	6
理論計算量	28.2	35.3	31.2	29.8
探索ノード数	18.3	10.0	12.0	26.4
探索成功率	100.0%	100.0%	100.0%	100.0%

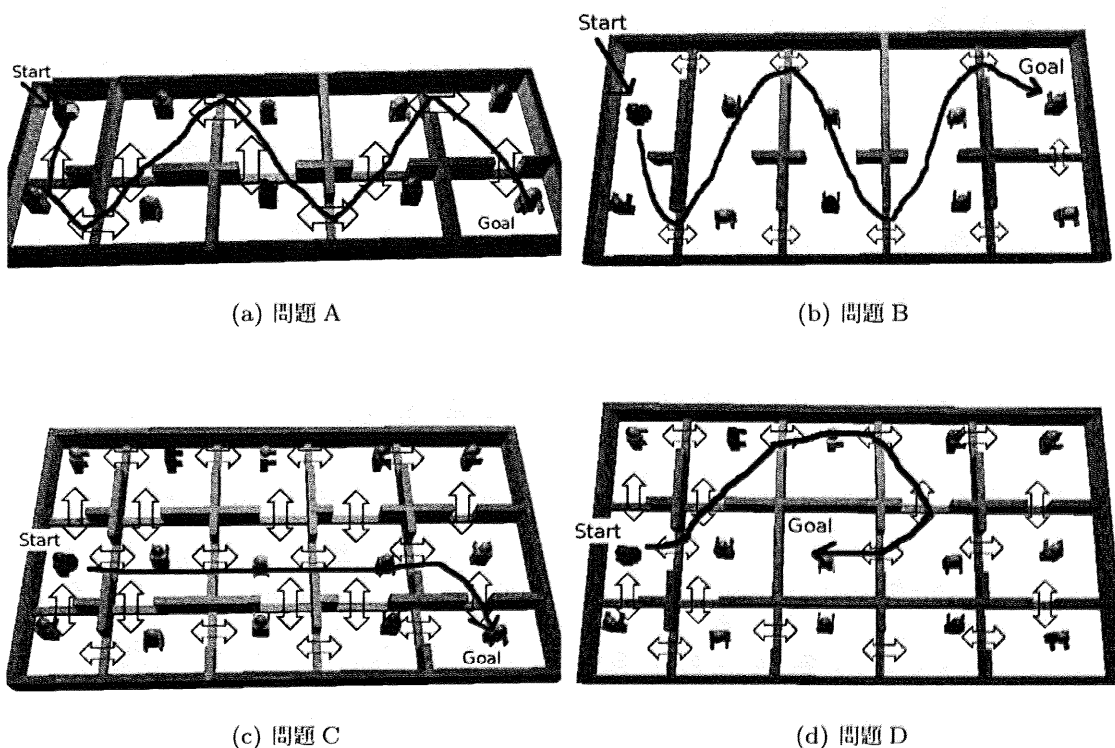


図 5.3 得られた手順の例

値である。これによると、環境 C や D のように複雑な環境でも理論値以下の探索ノード数で経路を得られている。問題 B では、同じ部屋に配置されている二つのロボットは互いに荷物を受け渡すことが可能である。提案アルゴリズムは、無駄な受け渡しを行わず、受け渡しの回数が最小 (4 回) となる手順を見つけた。これは提案アルゴリズムが有効な探索を行っていることを示している。

得られた受け渡し手順と前章で提案した経路プランニングアルゴリズムを用いることでロボットの経路プランが得られる。図 5.3 には得られた手順から求めた経路プランの例を示した。

表 5.2 探索ノード数 (受け渡し関係が未知の場合)

問題番号	A	B	C	D
ロボット数 $n_r$	10	10	15	15
最短経路長 $l$	9	4	5	6
理論計算量	30240.0	144.0	364.0	4368.0
探索ノード数	73.4	31.1	53.7	83.6
探索の成功率	100.0%	100.0%	95.0%	100.0%

### 5.2.4 受け渡し関係が未知の場合

受け渡し関係が未知の場合には、受け渡し手順探索を探索すると同時に、IsPassable ルーチンを使ってロボット同士の受け渡しが可能かどうかをチェックする。もし受け渡しが可能ならば、その受け渡しを使ってさらに深く探索を行う。

探索のアルゴリズムは以下ようになる。これは通常の最良優先探索のアルゴリズムであるが、後述の実験では探索効率を高めるために双方向探索による実装を用いた。

1. データを初期化する。(キュー  $queue_s$  を空にする。すべてのロボット  $r_i$  ( $r_i \in R$ ) について、キュー  $queue_g$  に初期経路 " $r_s \rightarrow r_i$ " を入れる。)
2. キュー  $queue_s$  から最小コストの経路を一つ取り出し、 $path_s^c$  とする。その経路  $path_s^c$  の最後のロボットを  $r_c$  とし、最後から 2 番目のロボットを  $r_{last2}$  とする ( $r_c \leftarrow \text{LastRobot}(path_s^c)$ ,  $r_{last2} \leftarrow \text{SecondLastRobot}(path_s^c)$ )。
3. ロボット  $r_{last2}$  が物体をロボット  $r_c$  に受け渡せるか进行检查する ( $\text{IsPassable}(r_{last2}, r_c)$ )。もし、 $\text{IsPassable}(r_{last2}, r_c) = \text{false}$  ならば、その経路を破棄し、ステップ 2. に戻る。
4. ロボット  $r_c$  が物体のゴール位置に到達できるかどうか进行检查する ( $\text{IsReachableToGoal}(r_c)$ )。もし、 $\text{IsReachableToGoal}(r_c) = \text{true}$  ならば、経路  $path_s^c$  を結果として返し、終了する。
5. まだ物体を受け取ったことのないすべてのロボット  $r_j$  について、経路  $path_s^j$  を生成し、キュー  $queue_s$  に入れる。経路  $path_s^j$  とは、 $path_s^c$  の最後にロボット  $r_c$  からロボット  $r_j$  までの受け渡しを追加した経路である。すなわち、 $path_s^j = \text{concatenate}(path_s^c, "r_c \rightarrow r_j")$  である。
6. ステップ 2. に戻る。

IsPassable ルーチンによる検査は最小コストの経路が取り出された直後に実行される。これは IsPassable ルーチンの計算量が多いために、必要な時にだけ呼び出している。そのため、検査すべき経路をキューに入れる前 (ステップ 5.) にはこの検査を実行しない。

表 5.2 はそれぞれの問題での探索ノード数と成功率を示している。この探索ノード数は



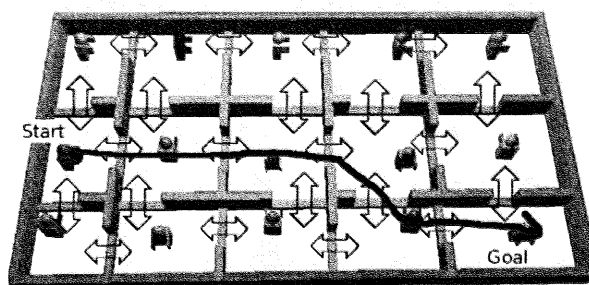


図 5.4 環境 C で得られた異なる手順の例

双方向探索によるものである。理論計算量は式 (5.1) から求めた。表から分かるように、探索ノード数は理論計算量よりも小さい。特に経路長  $l$  が長いほどその差は大きい。理論計算量はロボット同士の単純な順列の数であり、何の前提知識も使わず探索する場合の計算量である。ロボット間には受け渡しできるロボットとそうでないロボットという差がある。この実験結果から、提案手法が環境モデルから得られるそのような物理的条件を利用して、適切なロボットを選択し探索できていることを示している。

提案アルゴリズムはコスト計算が確率的探索に依存しているため、本節のアルゴリズムは前節の結果とは異なる手順が得られることがある。得られた異なる手順の例を図 5.4 に示した。手順は異なるが手順の長さは同じであり、どちらも受け渡しタスクは達成できる。

### 5.3 RRT による実装

RRT を用いた経路プランニングでは、環境モデルとロボットモデルだけを手がかりとして経路を探索できることから、アルゴリズムが非常に簡単となる。

RRT の探索では、ランダムサンプリングを用いているため、スタート状態とゴール状態をつなぐ経路の存在が確率的にしか判定できない。つまりプランニングを何度か繰り返して失敗しても、それが経路がないためなのか、それともさらに繰り返せば経路が得られるのかは分からない。

そこで、本研究ではヒューリスティクスをもちいて探索打ち切りを決定する。探索を一定回数繰り返して得られた最短距離を終了判定の指標として用いる。一定回数のうちにゴール位置までの距離が閾値以下にならなければ、そのような経路は存在しないとみなし、探索を打ち切る。閾値よりも小さくなれば、さらに長時間かけて探索する。

以下では、IsReachableToGoal 関数と IsPassable 関数の RRT を用いた実装を示す。

#### 5.3.1 IsReachableToGoal の実装

IsReachableToGoal ルーチンでは指定されたロボットがゴール位置まで直接荷物を運べるかどうかを検査する。ほかのロボットに荷物を受け渡してはいけない。これを検査す

るには、そのロボットの現在状態からゴール状態（ゴール位置に荷物を置くようなロボット状態）までの経路を探索すればよい。これは TestReachable ルーチン（第2.2.4節）を用いて実現できる。RRT を用いた双方向探索手法であるため、前記したヒューリスティクスで探索を打ち切る。探索が成功すれば、そのロボットがゴールまで荷物を運ぶ経路が存在する。

### 5.3.2 IsPassable の実装

IsPassable( $r_1, r_2$ ) ルーチンでは、ロボット  $r_1$  からロボット  $r_2$  へ荷物の受け渡しが可能かどうかを判定する。ロボット  $r_1$  は荷物を持っており、ロボット  $r_2$  は荷物を受け取る。荷物を受け渡す経路が存在すれば、IsPassable は true を返し、経路が存在しなければ false を返す。

これには4.2.2節で行ったように GenerateSubgoal( $r_1, r_2$ ) でサブゴールを生成してみればよい。このルーチンでサブゴールを生成できれば、ロボット  $r_1$  からロボット  $r_2$  へ荷物を受け渡せる可能性がある。このルーチンでは二体のロボット  $r_1, r_2$  以外を無視して経路プランニングしている。そのため、実際には他のロボットが障害となり荷物を受け渡せない場合がある。これを確認するには、受け渡し手順がすべて確定してから、すべてのロボットの経路を矛盾なくプランできるか試みれば良い。これは前章で示したアルゴリズムを適用することで確認できる。

### 5.3.3 コスト見積もり

手順探索アルゴリズムの初期化部分で、手順のコスト見積もりのための準備を行う。このコスト見積もりにも RRT を利用する。それには各ロボットの初期位置を始点とする RRT から、スタート位置とゴール位置に向けて一定回数だけランダムに枝を伸ばす。伸ばした枝のうち、最もスタート位置、またはゴール位置に近いノードでそれぞれに対する最短距離を計算する。この最短距離を、各ロボットからスタート位置またはゴール位置までのコストとして使う。

## 5.4 考察

### 5.4.1 実行時間

参考までに、実装したプログラムの実行時間を表5.3に示した。実験はそれぞれの環境に対して20回行い、平均値を示した。探索は最大10分で打ち切りとし、10分間で手順の得られなかったものは除いて平均をとった。実験に用いた計算機はLinuxオペレーティングシステム、Athlon XP 3200+ (2.2GHz) である。

受け渡しペアが既知の場合のほうが当然実行時間は短い。受け渡しペアが未知の場合には、計算時間と理論計算量(式(5.1))にやや相関がみられる。もちろん、ゴール位置やロ

表 5.3 実行時間の比較

問題番号	A	B	C	D
理論計算量 (式 (5.1))	30240.0	144.0	364.0	4368.0
理論計算量 (式 (5.2))	28.2	35.3	31.2	29.8
アルゴリズム A (受け渡し関係が既知の場合)				
実行時間 [秒]	8.50	11.51	6.05	14.76
探索成功率	100.0%	100.0%	100.0%	100.0%
アルゴリズム B (受け渡し関係が未知の場合)				
実行時間 [秒]	129.15	48.58	58.04	148.23
探索成功率	95.0%	100.0%	95.0%	95.0%

ポットの配置にも依存しているため完全に理論値で説明できるわけではない。

探索中に得た受け渡しペアをキャッシュして再利用することで、探索効率は受け渡し検査なしの結果に近づくはずである。

#### 5.4.2 関連研究

Okada らは、環境中の物体を操作しながら、ヒューマノイドロボットを目的地まで移動させるプランニングの研究を行った [27]。その中では、目的地となる場所まで Environment Manipulation Task Graph を生成し、サブタスクに分けている。これはあらかじめ与えられた移動可能な障害物の場所をサブタスクの区切りとして用いている。経路探索には A\* アルゴリズムを用いた。彼らは A\* アルゴリズムに用いるコスト関数を変えることによって、総負荷 (workload) 最小や距離最小の解を求めた。本研究で扱うタスクでは、複数のロボットが同時に動く。そのため全探索以外では最適解を求めるのが困難であり、彼らのような最適化手法が使えない。これについては 5.1 節で述べた。

Gravot らはシンボルでのプランニングと幾何モデルでのプランニングを組み合わせたシステム aSyMov を提案した [16]。しかし複数のロボットが存在する場合でも同時に 1 台のロボットしか行動できるようになっておらず、実用面では非効率的な結果しか得られない。また、彼らが用いた経路プランニングは PRM [36] を用いているため、多数のロボットが同時に行動する環境では探索性能が急激に劣化すると考えられる。提案手法では多数のロボットが行動する環境でも経路プランが得られることを実験的に示した。

### 5.5 本章のまとめ

本章では、ロボットの荷物受け渡しタスクの受け渡し手順を探索するアルゴリズムを示した。第 2 章で提案した経路プランニングアルゴリズムを用いることで、少ない情報をも

とに探索するアルゴリズムを実現した。さらに、環境のモデルを用いて、受け渡しが可能か検査しながら探索するアルゴリズムも示した。シミュレーション実験を行い、アルゴリズムの有効性を確認した。

## 第6章

# ヒューマノイドロボットによる協調 荷物搬送の実現

前章までは複数ロボットによる協調作業のための、RRTを用いた経路プランニングアルゴリズムを提案し、シミュレーション環境で検証を行った。

本章では、実環境での提案アルゴリズムの運用を想定し、ヒューマノイドロボットでの協調荷物搬送作業を実現する。これまでヒューマノイドロボット同士による協調作業の自律化はほとんど研究されていなかった。実環境でロボットを動かす際には外乱によって「ずれ」が生じる。そこで実環境でのずれの調整を行うために、再プランニングを行う手法を提案する。その実現にはカメラ画像を用いる。再プランニングアルゴリズムの有効性をヒューマノイドロボットで検証する。

本章は次のような構成となっている。6.1節ではヒューマノイドロボットを用いた実環境へのシンプルな適用の結果と問題点を述べる。6.2節では提案する再プランニング手法を示し、6.3節で検証のために構築したシステムについて説明する。6.4節では実験結果を示す。6.5節で考察と関連研究を述べた後、6.6節でまとめる。

### 6.1 実環境へのシンプルな適用

#### 6.1.1 ロボットと実験環境

まず予備実験として、シミュレーションで得られた経路をヒューマノイドロボットにそのまま適用し、実行を試みる（これを以下では実環境への「シンプルな適用」と呼ぶ）。使用したロボットは富士通オートメーションのヒューマノイドロボット“HOAP-1”（図6.1）2台である。HOAP-1は全身で20自由度持っており、全身制御は難しい。今回の実験では、あらかじめ用意した前進、左回転、右回転の3種類の行動だけを使用した。

今回使用したロボットでは頭部にUSBカメラをつけており、外部の環境を画像処理して認識する。このカメラは水平方向（前方を0度とし、-90度～90度の範囲）と垂直方向（前方を0度とし、0度～180度の範囲）に回転することができ、望みの方向の画像を

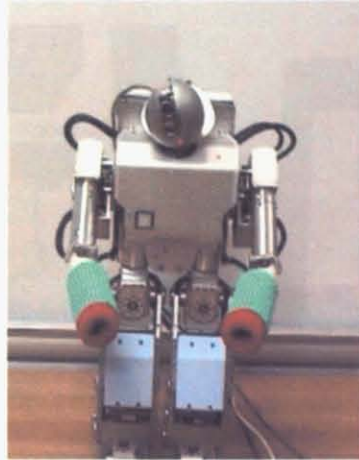
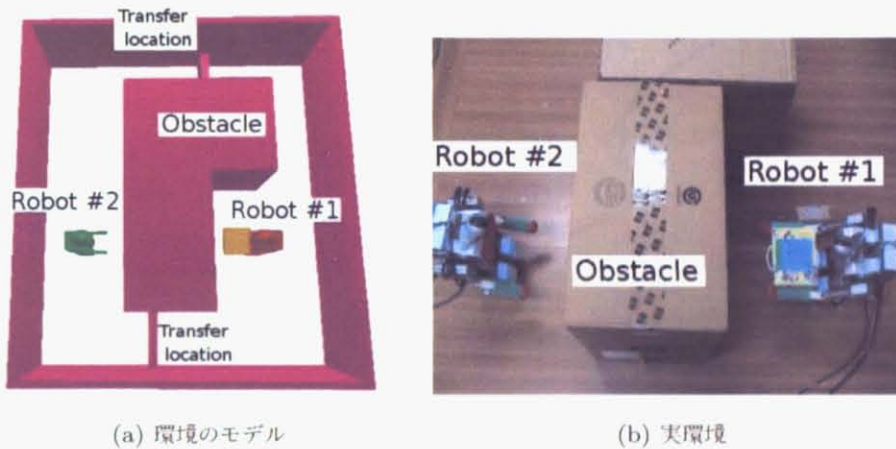


図 6.1 ヒューマノイドロボット HOAP-1. 頭部には単眼の USB カメラが取り付けられている。



(a) 環境のモデル

(b) 実環境

図 6.2 シンプルな適用に用いた環境とモデル

得ることができる（ただし、カメラはロボット頭部に対して 45 度前方に傾斜して取り付けられているため、カメラ垂直方向の 0 度はロボットの前方や下向きに相当する）。

実験環境として第 4 章で用いた複雑な環境を再現するのは困難である。そこで実験には図 6.2 のように単純化した環境を用いた。ここではサブタスクは 1 つだけとし、荷物を持ったロボット (Robot #1) がもう一つのロボット (Robot #2) に荷物を受け渡すまでを行った。

### 6.1.2 経路生成, 行動生成

図 6.3 は提案アルゴリズムで生成された経路プランの例である。生成されるプランは 2 種類に分類できた。一つは環境の手前側に移動する経路 (図 6.3(a)) であり、もう一つは環境の奥側に移動する経路である (図 6.3(b))。このうち図 6.3(a) のような経路はもう一

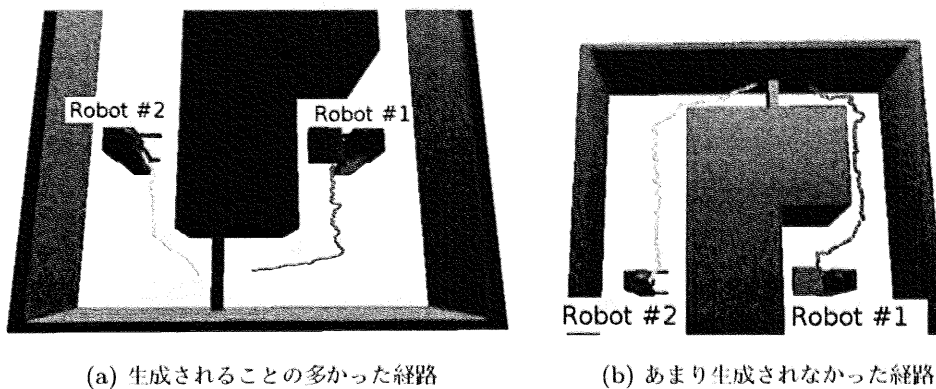


図 6.3 提案アルゴリズムで生成された経路の例

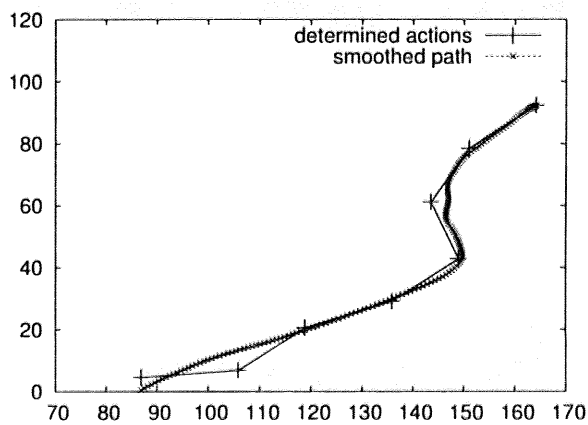


図 6.4 スムース化された経路と経路に沿って決定された行動の軌跡

方のものよりも生成される場合が多かった。なぜならば、図 6.3(a) の経路のほうが単純で距離も短いからである。これは 4.4.1 節で確認したように、提案手法では短い経路のほうが選ばれる確率が高いという性質があるためである。

ロボットで実行するために、このようにプランニングで得られた経路をロボットの決められた行動の系列で表現する。つまり計画された経路を「前進」、「左回転」、「右回転」の 3 種類の行動の系列に変換しなければならない。プランニングアルゴリズムを用いて実際のロボットを動かす手順は以下ようになる。

1. 初期位置からのプランニングを行い、経路を生成する。
2. 経路のスムース化を行う。
3. スムース化した経路に沿う行動を決定する。
4. 行動を実行する。

プランニングではロボットが平面上を自在に行動できるものと仮定しており、またランダ

ムを用いたアルゴリズムであるため、必要以上に経路が細かい動きを含みやすい。そのため得られた経路をスムーズ化する必要がある。

スムーズ化した経路に3種類の行動（前進、左回転、右回転）を当てはめる。図6.4にその例を示した。このためには、スムーズ化した経路を一定間隔でたどりながら、基準となるロボット位置からの前進方向と回転方向のずれを計算する。このずれがある閾値を越えたところで前進または回転行動を出力し、ロボット位置を移動させる。これを経路の最後まで繰り返すことで、スムーズ化した経路から行動系列を得られる。それぞれの行動の閾値はロボットの実際の移動量から求めた。この3種の行動の粒度が大きすぎると、経路の再現性は悪くなる。図6.4に示したように、用いた3種の行動で十分な精度で経路を再現できることを確認した。

最後に、得られた行動系列をロボットに送信して、順に実行する。

### 6.1.3 荷物受け渡し

サブゴールに向けて計画された行動がすべて終了した段階で、荷物受け渡しルーチンを起動する。この荷物受け渡しルーチンは次の3ステップから成っている [56]。

1. 接近ステップ（2体のロボットがお互いに近づく）
2. 位置合わせステップ（2体のロボットがお互いに向き合うように位置と向きを微調整する）
3. 受け渡しステップ（2体で荷物を受け渡す）

ロボットそれぞれの行動はカメラから得られた画像をもとにして決定する。ロボットと荷物には目印となるマーカーを付け、カメラで相手ロボットや荷物を認識し位置を判定している。

この受け渡しルーチンではマスタースレーブ制御が行われている。位置を調整するためにそれぞれのロボットが行動するが、荷物を持ったロボットがマスターとなり、それぞれの行動の整合性を取っている。受け渡しの成功率を上げるために、本実験は井上 [56] による行動決定ルーチンと使用するロボット動作には改良を施して用いた。

受け渡し作業の実行例を図6.5に示した。まずロボット同士が近づきながら向きを合わせる（図6.5(1)-(2)）。そして受け手側ロボットがしゃがみこみ荷物の受け入れ準備をした後、荷物が受け渡される（図6.5(3)-(4)）。その後、受け渡し側ロボットは後退して距離を取り、受け手側ロボットは荷物を持ったまま立ち上がる（図6.5(5)）。これで荷物受け渡し作業は完了である。

### 6.1.4 プランの適用結果

図6.2(a)の環境モデルで得られた経路（図6.3(a)）を実際にロボットで実行した。

図6.6には成功した試行のスナップショットを示した。最初ロボットは環境の手前側に



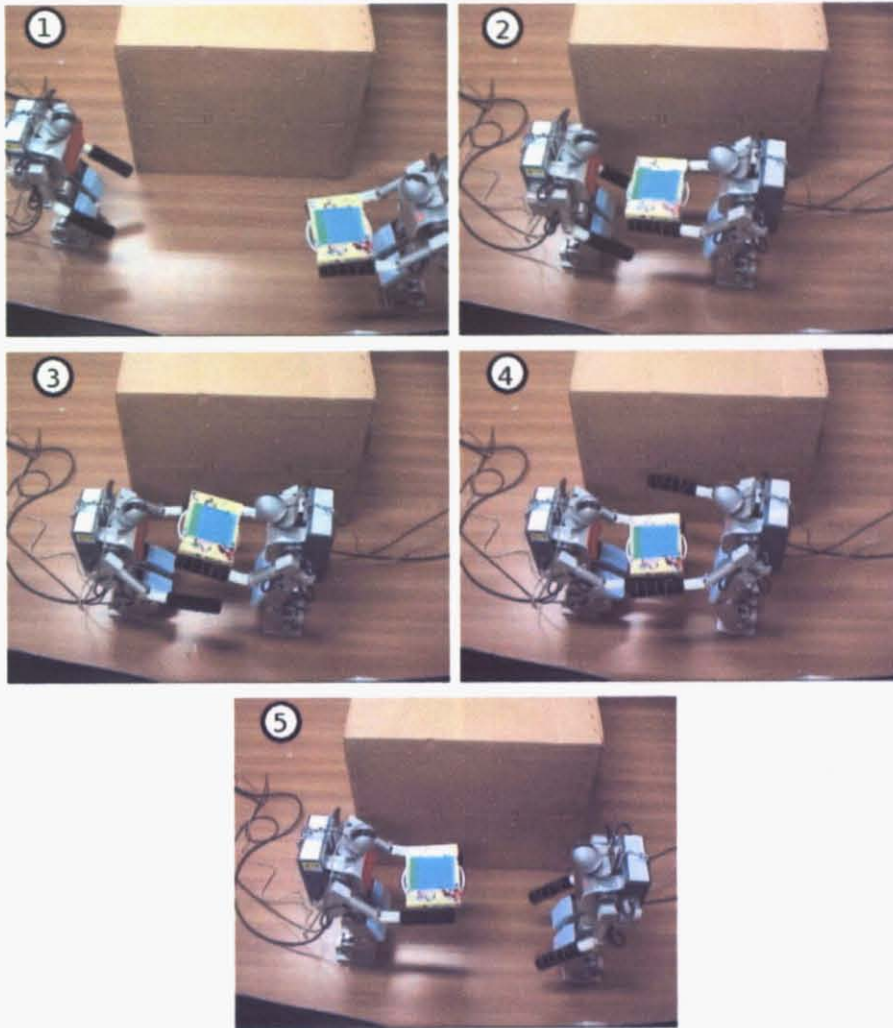


図 6.5 荷物受け渡し作業の実行例

向かって回転し、障害物を回避した(図 6.6(1)-(2))。次に、ロボットは前進し、お互いが正面に向き合うように回転した(図 6.6(3)-(4))。そして、物体を持ったロボットがもう一方のロボットに近づいていった。その間、物体を持っていないロボットは、もう一方のロボットをカメラでとらえられず、そのまま静止していた(図 6.6(5))。その後、ロボットが互いを認識して受け渡しルーチンを実行し、タスクを達成した(図 6.6(6)-(8))。

### 6.1.5 実環境での問題点

この実験を何度か行う中で、ロボットは床面との摩擦や滑りの影響でプランニングされた経路から徐々にずれていった。ずれた結果として、障害物に衝突することがよく起こった。これは経路プランニングではロボットの行動が理想状態からずれることを想定しておらず、一旦ずれてしまうと回復できないためである。実環境はノイズや摩擦などの不確定な要因が多いために、これらを正確にモデル化することは大変困難である。このことが

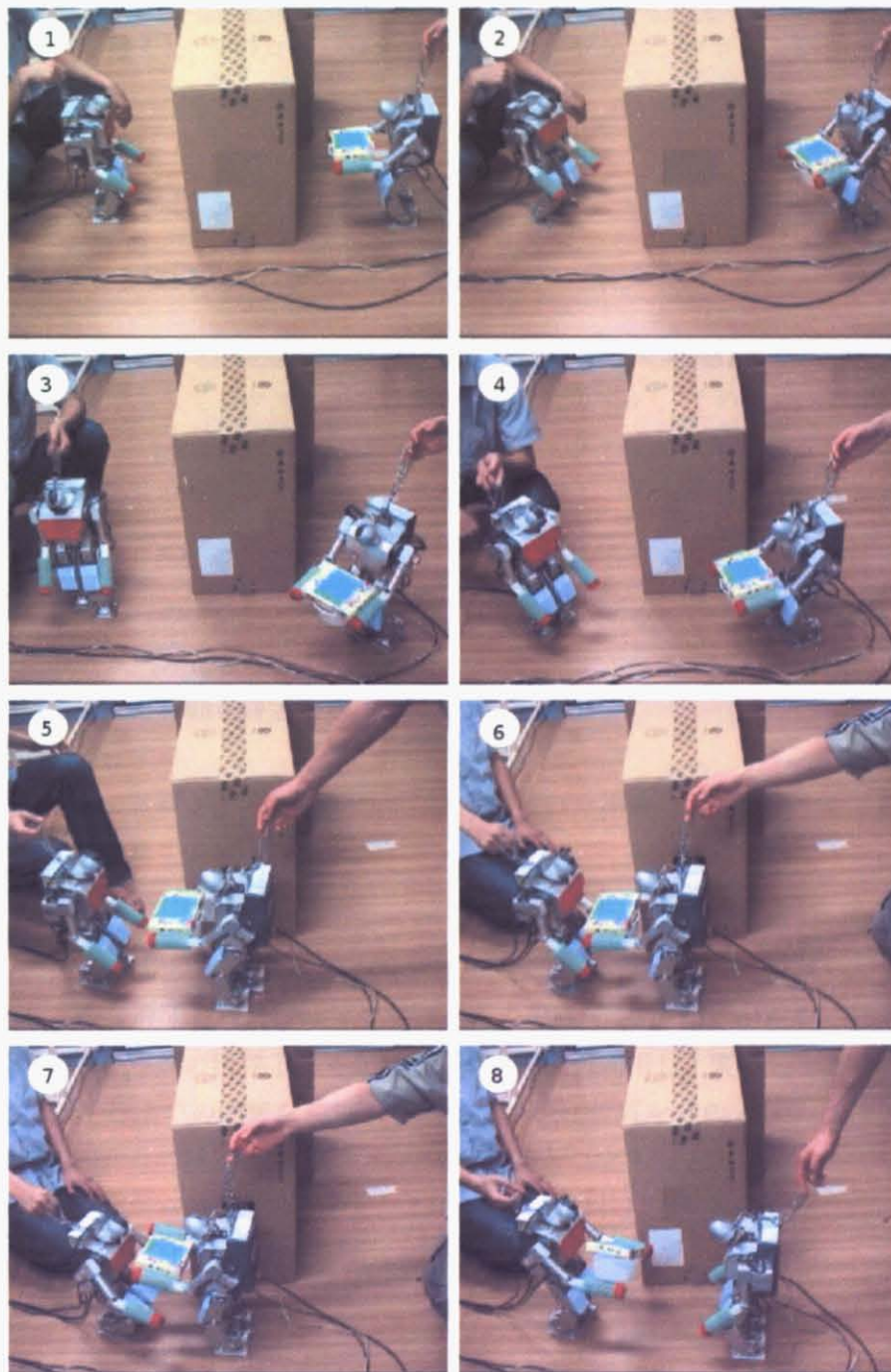


図 6.6 シンプルな適用による協調搬送タスクの成功例

ら、プランニングから得られた行動を単純に実行するだけでは、簡単な環境であってもロボットが障害物に衝突してしまい、タスクが達成できないことが分かった。

この対処として、ロボットに計画された経路からのずれを認識させることが考えられる。この実験ではロボットが移動中にカメラからの画像情報を利用していない（荷物受け渡しルーチンを除く）。そのため、ロボット自身がプランニングされた経路からずれていることを認識できなかった。

## 6.2 再プランニングの提案

### 6.2.1 再プランニングのアルゴリズム

前記のように、プランニングにより得られた経路をそのまま実行するだけではずれが生じ、簡単な環境であってもタスクが実行できないことが多い。そのため、ロボット頭部のカメラを用いて環境を認識しながら経路を調整しつつタスクを実行することが必要である。これは現在の状態を認識して必要に応じて再プランニングを行う必要がある。

提案する再プランニング手法は以下の手順である。

1. 初期位置からのプランニングを行い、経路を生成する。
2. 経路のスムーズ化を行う。
3. 経路に沿った行動を決定する。
4. 行動を実行する。
5. ロボットの位置が予定した経路から大きくずれた場合、現在位置をもとにプランニングをやり直す。
6. ステップ2. から繰り返す。

再プランニングではサブゴールは変更せず再利用する。再利用したサブゴールを用いるとプランニングに失敗する場合には、サブゴールの生成（4.2.2節）からやり直す。今回の実験では実験スペースの関係上、できたサブゴールを再利用し、変更しないものとした。

これを実現する上での困難な点は、ロボットの現在位置をどのように認識するかである。位置認識のための特別なハードウェアを用意することもできるが、ここでは手軽な方法として、カメラからの画像を用いた。これについては後ほど詳しく述べる。

## 6.3 システム構成

ヒューマノイドロボット上で再プランニングを行うために、制御システムを構築した。それには井上による協調型マルチエージェントシステム [56] を拡張した。このシステムは複数のロボットをサーバからの指令で制御でき、ロボットやカメラの情報を中央サーバに集めて、判断、制御を行うことができる。各ロボットとサーバの通信はTCP/IP上で同期通信を行い、コマンドを受け渡しすることで実現されている。

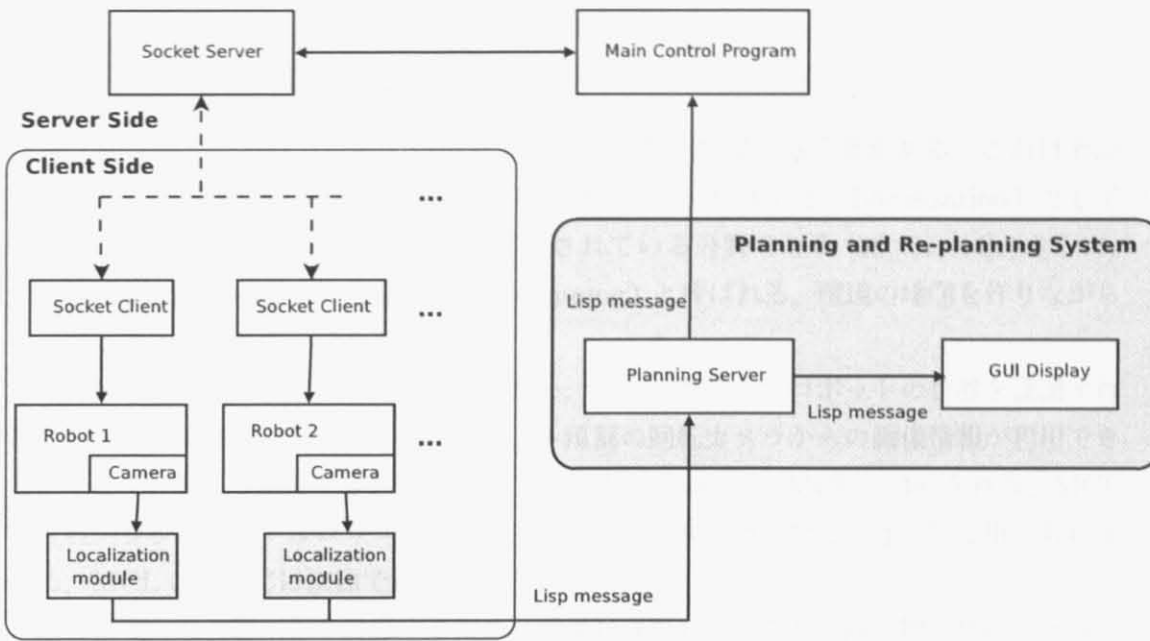


図 6.7 システム構成図 (色つきの部分と Lisp message の通信路が追加した部分)

本実験のために、経路プランニングルーチンをシステムに追加した。そのほか、画像認識による位置同定モジュール (localization module) や経路プランの送受信など、プランニングルーチンとの密接な通信を行うために新たな通信機能を追加した。図 6.7 にシステム構成図を示した。追加サーバでの通信プロトコルには Lisp の S-式の構文を用いた。その理由としては、構文やデータ構造がテキストベースであること、柔軟なデータ構造 (可変長のリスト) を容易に扱えること、データそのものを Lisp 処理系で解釈することでイベントドリブン (非同期) のプログラムが実現できること、さらにコマンドラインインターフェースを利用することで動的に機能追加変更が容易なこと、などである。ロボットの動作や位置同定に要する時間は、行動やロボットの環境中の位置などによりそれぞれ異なる。先に動作が終了したロボットから次の行動を実行できるように、非同期の制御が望ましい。ただし再プランニングを実行するにはロボットのタイミングをそろえてから行う。

S-式の解釈系の実装には Lisp の方言となる Scheme の処理系である guile [17] と gauche [15] を用いた。これらの上で Lisp の関数を書くことで容易に機能拡張できる。井上による既存のシステムは C++ 言語で書かれているため、S 式を受信するための解析ルーチンを用意することで既存ルーチンとの共存を実現した (S 式の送信は単純に文字列出力を行うだけである)。

以下では、各ロボット上で実行される位置同定モジュールで用いた位置同定 (Localization) の処理について説明する。

### 6.3.1 位置同定のための画像処理

再プランニングを行うには、ロボットの現在位置を把握する必要がある。これはセンサを用いて位置を認識するという一般的な問題であり、位置同定 (Localization) として知られている。位置同定は近年盛んに研究されている分野である [38, 61, 44]。さらに、SLAM (simultaneous localization and mapping) と呼ばれる、環境の地図を作りながらの位置同定の研究も盛んである [1]。

本実験では環境のモデルが既知であるため、モデルを用いてロボットの位置を認識すれば良い。使用できるセンサとしてはロボット頭部の回転カメラからの画像情報が利用できる。そこで、カメラ画像をもとに Monte Carlo Localization (MCL) [38] を行う。MCL は比較的シンプルなアルゴリズムであり、ロボカップサッカーなどで近年広く用いられている。MCL については次節で詳しく述べる。

画像情報から位置情報を計算するため、画像処理が必要となる。障害物の壁にマーカー (ランドマーク) を取り付けて、それをもとに位置同定を行う。ランドマークから環境中の位置が一意に定まるように、それぞれマーカーの色を変えている。この詳細は 6.3.3 節で述べる。

画像の色認識には CMVision [6] ライブラリを利用した。これは高速な色認識ライブラリである。対象色の認識には YUV 色空間が用いられている。

なお、位置同定と荷物受け渡し動作 (6.1.3 節) は同じカメラを利用しているが、利用するマーカーも認識アルゴリズムも異なる。動作モード (位置同定モードと荷物受け渡しモード) に応じて認識のモードを切り替えている。

### 6.3.2 Monte Carlo Localization による位置同定

MCL 手法では自己位置の候補を表現した多数のパーティクルで推定する。このためパーティクルフィルタとも呼ばれる。用いるパーティクル集合はロボットの存在確率分布を表現している。ロボットが移動する際には、パーティクルで表現された位置候補も移動させる。そして、カメラ画像と自己位置候補からの推定画像とを比較し、もっとも確からしい位置候補を選び出す。この処理を行動ごとに繰り返す。以下の自己位置同定での「位置」には環境中の座標に加えて、ロボットの進行方向の情報も含めて扱う。

具体的には以下のようにして位置同定を行う。

1. 初期位置  $l_0 = (x, y, \theta)$  に初期ノイズをのせて、解候補  $l_0^{(i)}$  ( $i = 0, 1, \dots, N$ ) を初期化する。
2. ロボットに (プランナからの) 行動指令を与え、行動させる。
3. ランドマークの観測を行う。
  - (a) まず、1 時刻前のロボット位置  $l_{t-1}$  に、行動による変位量を加え、仮の推定位

置を得る.

$$l_t \leftarrow l_{t-1} + a_{t-1} \quad (6.1)$$

( $a_{t-1}$  は時刻  $t-1$  で選択した行動による移動量をモデル化したものである.)

(b) 次に, 仮の推定位置  $l_t$  に基づき, 視野内に入るランドマークを観測する. このときの画像を保存しておく.

4. 解候補それぞれについて, 行動による変位量を加え, 推定位置を得る.

$$l_t^{(i)} \leftarrow l_{t-1}^{(i)} + a_{t-1} + noise \quad (6.2)$$

推定位置にはノイズ (位置に関しては正規分布乱数を, 角度に対しては一様分布乱数を使用) を付加して解候補を散らす.

5. 解候補  $l_t^{(i)}$  それぞれについて, 推定画像を計算し, ランドマークを観測して得られた画像と比較し, 一致度のスコアを計算する.

6. もっとも一致度の高い解候補の値を現在位置  $l_t$  として返す.

7. 時刻を進めて ( $t \leftarrow t+1$ ), ステップ2へもどる.

この実験に用いた MCL では, ロボットから最も近くに見えるランドマークを2つ観測して位置推定を行う [45].

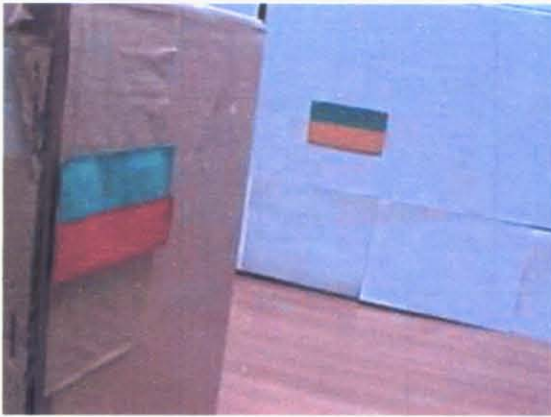
本実験の前に, 推定精度を確認するための新たな予備実験を行った. その結果, 厳密な初期位置でパーティクルを初期化するよりも, 初期位置にノイズを加えた値をパーティクルとした方が推定精度が良いことを確認した. そのため初期位置にはノイズを加えて MCL を実行した.

### 6.3.3 ランドマークの認識

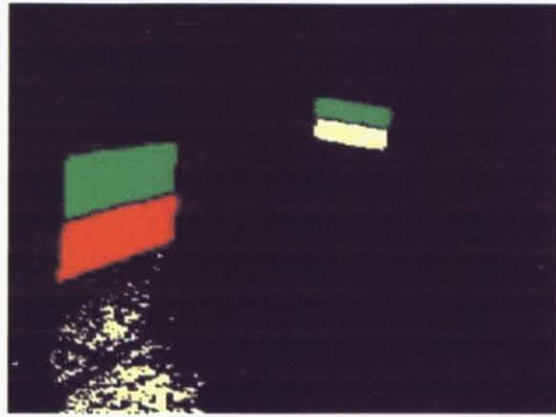
今回用いるカメラはカラー画像を認識できるが, 予備実験を行った結果, 安定して認識できる色の種類は4色程度であることが分かった. これは環境の明るさ (ホワイトバランスが自動的に調整されるため) やノイズの影響をうけるためである. 認識できる色をそれぞれ1つのランドマークに割り当ててしまうと, 4つのランドマークしか用いることができない. これでは環境中の位置を十分な精度で得ることができず, 広い環境にも対応できない.

そこで, カメラが認識できるランドマークの種類を増やすために, 図6.8(a)のように2色 (緑, 赤や緑, 黄など) を上下に隣接して1つとしたランドマークを利用した. これによりコストをかけずに, 認識できるランドマークの種類を増やすことができる. 隣接する2色領域を検出するための画像処理プログラムを作成した.

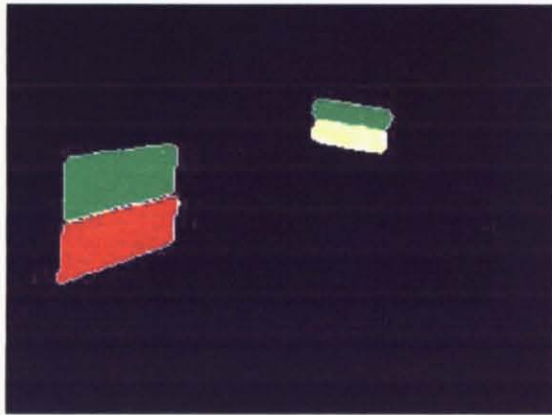
これはカメラから得られた画像をピクセル単位でスキャンし, 色領域の周囲をたどることで実現している. 具体的なアルゴリズムは以下の通りである. 実際の画像処理の例は図6.8に示した.



(a) 原画像



(b) 色認識アルゴリズムの結果



(c) 2色領域の周囲を検出した結果（検出した領域の周囲を白線で示した）

図 6.8 ランドマーク認識での画像処理の例

1. まずカメラから得られた画像（図 6.8(a)）に、色認識処理を行う。これにより、画像中からランドマークに使われる色だけを検出する（図 6.8(b)）。
2. 色認識結果画像をピクセル単位でスキャンし、ランドマークになりうる色を探す。画像の終わりに達したときは終了する。
3. 最初に見つけた色（ $c_1$  とする）の領域とそれ以外との境界線（色領域の外周）をなぞりながら、なぞったことを示すフラグを立てていく。もし境界が見つからなくなった場合には、ノイズなどの影響で間違った方向に進んだことが考えられるため、後戻りして他の境界をなぞり続ける。その間に2つ目の色（ $c_2$  とする）にたどり着いたときには、 $c_2$  の境界線をたどる。
4. 色領域を囲み終わったら、囲んだ二色（ $c_1$  と  $c_2$ ）の配置からランドマークの ID を判定して返す（図 6.8(c)）。ただし、閾値より小さい領域はノイズと見なして無視する。

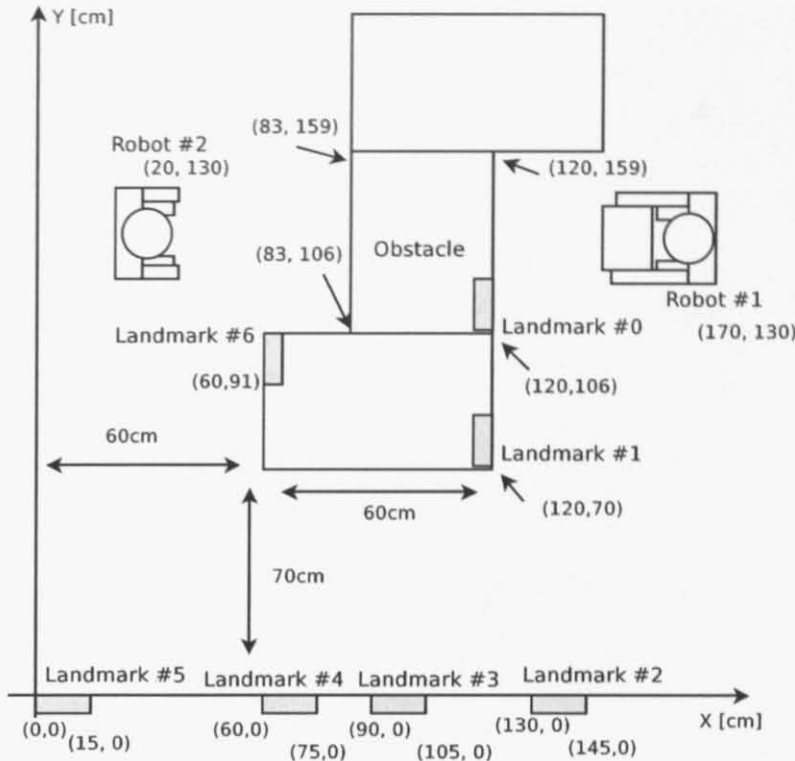


図 6.9 ランドマークの配置図 (色つきの部分がランドマークを示す)

#### 5. ステップ 2. に戻る.

あらかじめランドマークごとに割り当てる色の組み合わせを定義しておく。例えば、2色を上下に配置する場合には、ランドマーク#1は「上側が赤で、下側が緑」、ランドマーク#2は「上側が黄で、下側が緑」などと定義する。上記アルゴリズムは、発見した色領域の情報と、定義されたランドマークの情報から、ランドマークを識別する。これは図 6.8 に示したように、画像中に複数のランドマークが存在しても適切に処理できる。

本実験では図 6.9 のように 7 つのランドマークを配置した。いずれも縦 8cm, 横 15cm の長方形のマーカーを用いている。実験環境中では図 6.10 のように配置されている。

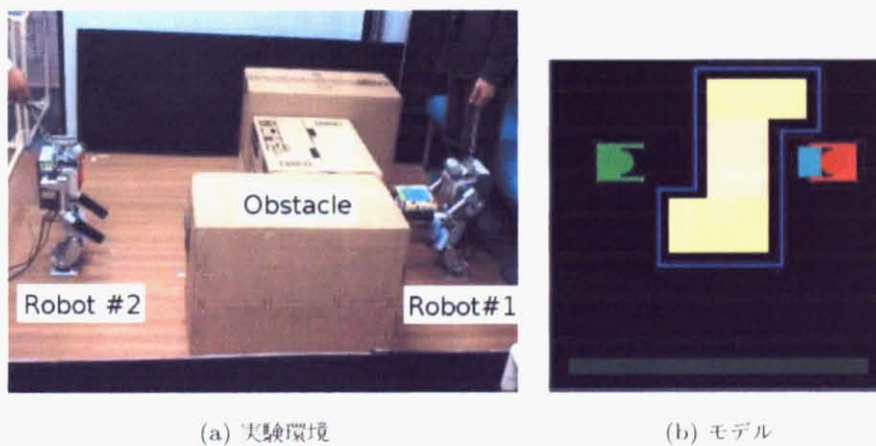
## 6.4 実験結果

実験に用いた環境とそのモデルは図 6.11 に示した。経路プランニングのためのモデルとしては、障害物の領域を拡大したモデル (図 6.11(b)) を用いた。これはロボットの 1 行動による移動量の余裕を持たせるためである。このようにモデルの幅を広げる手法は、安全な動きをプランニングするために [34] などでも用いられている。また、RRT を用いたプランニングでは壁に沿うような動きができやすいため [2], 壁と衝突しやすい経路を防ぐ意味もある。実環境ではずれが起こるため、このような工夫を行ったとしても経路からのずれや壁への衝突をなくすことは難しい。





図 6.10 環境中のランドマーク (いずれも図 6.9 の上方から下方を見た様子)



(a) 実験環境

(b) モデル

図 6.11 実験環境とモデル

実験結果のスナップショットを図 6.12 に示した。荷物を受け渡すためのサブゴールは画面の下方である。図 6.12(1)-(5) の右図ではロボットによる自己位置推定の結果を示している。図中の点線は現在の経路プランである。ユーザは図と同じインタフェースで、タスク実行中に現在位置を確認できる。

実験開始直後、サブゴールに向けて進むために、ロボットはまず壁を避ける方向へと回転した(図 6.12(1))。次にサブゴールに向けて進んだ(図 6.12(2))。ある程度進んだところで、現在位置が経路プランから大きくずれ障害物に接近したため、以降の行動で衝突

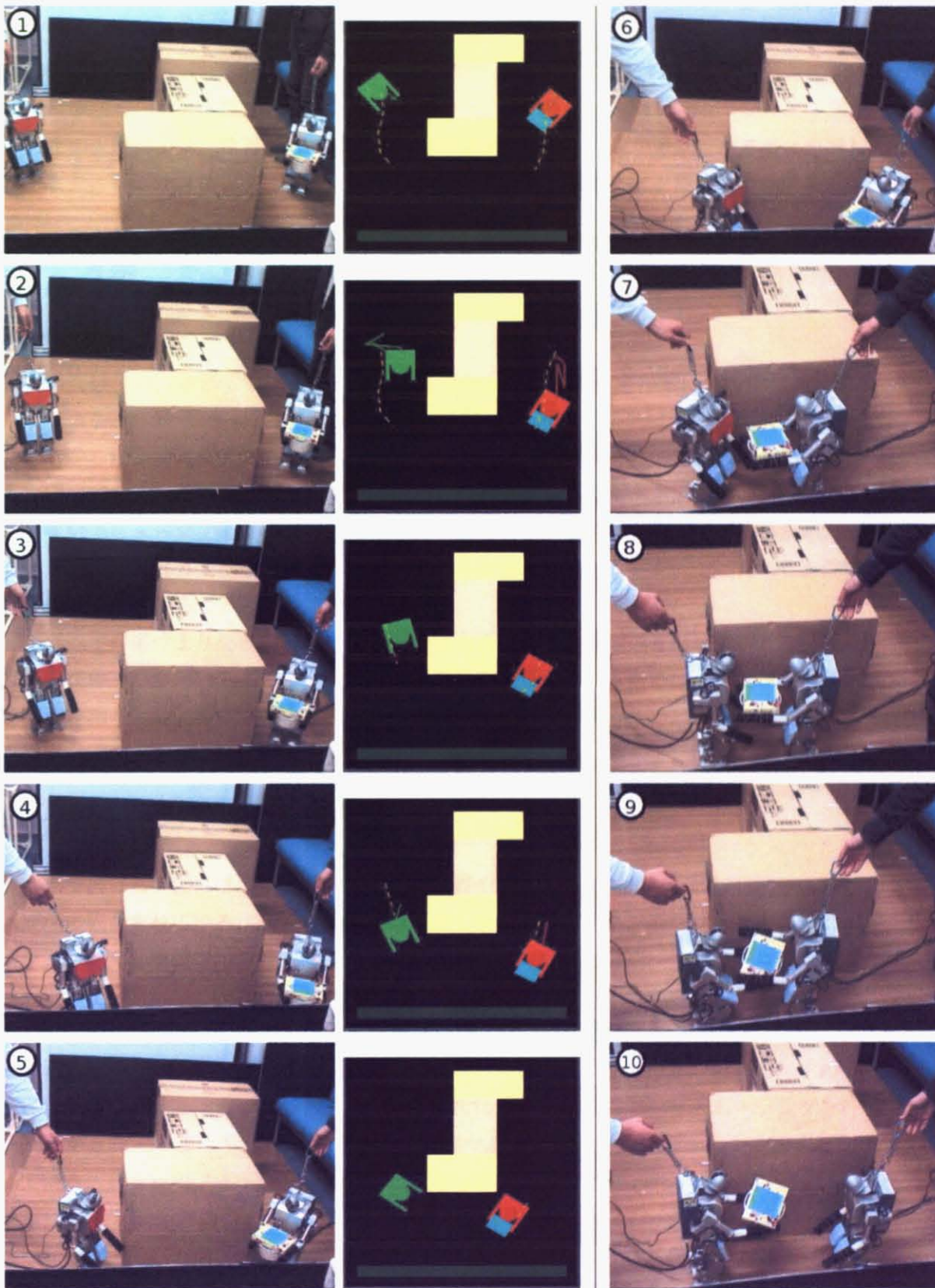


図 6.12 再プランニングの実験結果

する可能性が検出された。そこで再プランニングが行われ、新たな経路プランが得られた (図 6.12(3))。新たなプランにしたがって行動し、サブゴール位置まで到達した (図 6.12(4)-(5))。この後、荷物受け渡しルーチンが実行された (図 6.12(6)-(10))。荷物受け渡しに成功し、タスクは達成された。

以上の実験により、位置情報にもとづき再プランニングの実現が可能であることを示した。また、再プランニングによって適切な行動経路が得られることを確認した。

## 6.5 考察

### 6.5.1 実行時間

この実験では、ロボットの行動には 6~15 秒かかり、位置同定作業には 10~20 秒、再プランニングには 3~5 秒かかった。本実験で用いた位置同定手法は速度が比較的遅かった。しかし提案アルゴリズムを実環境で検証することが本実験の目的であったため、そのまま用いている。実用を考える場合には、より高速化する必要がある。

本実験でヒューマノイドロボットに取り付けたカメラが単眼であり視野も狭いため、位置同定のためにはカメラをランドマークに向ける作業が必要であった。そのため位置同定作業に時間がかかることとなってしまった。

位置同定作業はより高速な手法や特別な装置を用いることで高速化可能であろう。例えば全方位カメラを用いることで、一度に全方向の画像を得られるため、画像取得にかかる時間が短縮される。用いる手法や装置は位置推定精度に影響するため、環境で必要とされる精度に応じて決定されるべきである。

### 6.5.2 関連研究

本章で扱った実験には今後ロボットが産業分野や日常空間での作業一般に必要とされる機能が多数含まれている。それは、障害物がある中での経路プランニングとその実行、再プランニング、環境中での位置同定、物体の認識などの機能である。

実ロボットを対象とする経路プランニングは多くの研究が行われている。その中でも障害物と衝突しないように余裕を持たせた経路プランニングは重要な問題である。特に A\* などのグリッドベースのプランニングでは障害物に近い経路ができやすいために、その安全性を保証することは重要である。そのため障害物から離れた方向へとスムージングを行う研究がされている [51]。これらは主に車輪型ロボットのためのプランニングを目標としている。ヒューマノイドロボットの経路プランニングも多数行われている [43, 29]。しかしロボット同士の協調作業の自動化に焦点を当てた研究はわずかである。井上は複数のヒューマノイドロボットが協調するためのシステムを構築し検証した [56]。Shinozawaらは人間サイズのロボットと小型のロボットという異種ロボット同士での協調や作業領域ごとの役割分担を提案した [35]。しかし本研究では、これら研究では実現されていない環境中の自己位置同定や経路プランニング機能を統合したことでより広範囲の応用に有用で

ある。

次にロボットに必要な機能として位置同定や環境地図の自動作成 (SLAM) が挙げられる。MCL は前述 (6.3.1 節) したように位置同定手法として広く研究されている [38, 61, 44]。本実験では環境の地図をあらかじめ与えたため、SLAM を行う必要がなかった。しかし自動的に地図が作製されれば、環境が変化する場合でも人手で地図を作製する手間を省くことができる。SLAM の研究ではレーザレンジファインダが広く用いられている。レーザレンジファインダはレーザを用いて対象物までの距離を測定する装置で、その正確さと認識領域の広さから使われている。レーザレンジファインダやビジョン、超音波センサーなど複数のセンサからの情報を組み合わせて環境の地図を作る研究も行われている [37]。センサ情報を統合して用いることでパーティクルフィルタによる推定精度が向上すると報告されている。また、Rao-Blackwellized Particle Filter を用いると、SLAM と位置同定を同時に行うことができる [52, 38]。ただし計算量は非常に多い。

環境地図の作製を人間の生活空間で行う場合には、人間などの移動物体をどのように認識し、どのように取り扱うかが課題となる。動的物体 (人間など) や準動的物体 (頻繁には動かないが、まれに動く物体、ドア、椅子など) を扱う研究が行われている。Wolf らは静的物体のマップと動的物体 (準動的物体も含む) のマップを同時に生成する手法を提案している [40]。そのほか、準動的物体を扱うために、RFID と組み合わせて地図から除く研究が試みられている [48]。

今回のタスクでは扱っていないが、今後は経路プランニングにおいても他の移動物体の存在を考慮できることが重要である。つまり移動している物体を認識し、それと衝突しないような経路をプランする必要がある。

Petti ら [34] や坂原ら [60] は、移動物体を考慮する実時間 RRT を提案している。これらのアルゴリズムでは、移動物体の速度を短時間では一定と見なせるとして将来の位置を予測し、RRT で経路を生成する。これら手法はシミュレーション環境で有効性が確かめられた。ただし通常、実環境ではノイズなどの影響があり移動物体の位置と移動速度を正確に測定することは難しく、それらの情報をロボットが知覚できる範囲も限られるはずである。このため、ノイズを含む限定された情報の下で得られる経路プランの有効性をより詳細に検証する必要があるだろう。

物体の位置と移動速度の精度良い測定は難しい問題であるが、その推定に MCL を使った例がある [18]。これはカメラ画像中のボールの位置と速度を MCL によって推定し、トラッキングを行う研究である。これによると、ボールが視野から出るまで位置と速度を推定できたと報告されている。ただし、この研究はカメラが固定された状況で実験されていた。今後の応用を行う上では、カメラ (ロボット) が動いている場合にも精度良く推定できることが必要となろう。また、ボールだけに限らず日常生活中に現れるような物体を追跡できることも必要である。

以上述べたようにこれらの分野ではまだまだ取り組むべき課題が多く、更なる研究が必要である。

## 6.6 本章のまとめ

本章では、第4章で提案した経路プランニングアルゴリズムの実環境への応用を想定し、2体のヒューマノイドロボットで協調荷物搬送作業を行った。そのためにカメラの画像処理や位置同定、再プランニングを実現するモジュール、そしてヒューマノイドロボットの制御システムを構築した。実験の結果、カメラ画像からの位置同定結果を用いた再プランニングが実現可能であることを確認した。

## 6.7 謝辞

6.3節で述べた協調型マルチエージェントシステムは井上豊氏により作成された制御システム [56] を拡張して利用させて頂いた。6.3.2節のMCLアルゴリズムの実装は堀口研一氏 [46, 45] によるものを使わせて頂いた。6.3.3節で用いたランドマークの認識アルゴリズムはCem Erdemir氏により、われわれのために作成された。協力して頂いた方々に感謝致します。

## 第7章

# 結論

### 7.1 本研究のまとめ

マルチエージェント環境での経路プランニングを実現するアルゴリズムを提案した。これは複数のプランニングルーチンからなり、タスクに応じて組み合わせて用いる。本論文では、2つのタスクに適用し、シミュレーションとヒューマノイドロボットを用いた実験を行うことで提案アルゴリズムの有効性を確認した。

第2章ではマルチエージェント環境に必要となる経路プランニングを分類した。そしてそれらを実現するアルゴリズムを示した。

第3章では提案アルゴリズムを物体再配置問題に適用した。提案手法を用いることで、複数ロボットによるタスク分散を考慮した経路を得られることを示した。比較実験として、ダイナミックプログラミング (DP) による手法との比較を行った。その結果、提案手法のほうがより短い実行時間で DP と同程度の経路プランが安定して得られることが確認された。

第4章では提案アルゴリズムを協調荷物搬送問題に適用した。少ない情報からマルチエージェント協調作業のための経路が得られることを示した。タスクに必要なサブゴールは提案アルゴリズムを用いることで、自動的に生成できた。また、提案手法では短い経路が選ばれやすいという望ましい性質を持つことを確認した。

第5章では協調荷物搬送問題の受け渡し手順探索に応用した。サブタスクの順序をロボットのモデルと環境のモデルを利用して自動的に導出できることを示した。また、第4章で示したアルゴリズムが必要とする情報をさらに少なくすることに成功した。これはシンボルによるプランニングで必要とされる物理的制約の記述を環境モデルなどから自動的に導出することに相当する。

第6章では協調荷物搬送問題の提案アルゴリズムをヒューマノイドロボットに適用し、実環境での有効性を示した。シミュレーションでの結果を単純に適用するだけでは、実環境での動作で「ずれ」を生じるため、タスクを達成できないことが多い。そこでずれが生じた場合に、再プランニングを行い経路を生成し直すことで、実環境でも有効な経路プランニングを実現した。再プランニングを行うには、ロボットが自己位置同定を行うことが

必要である。そのため、カメラ画像を用いた自己位置同定と連携するロボット制御システムも実現し、実験で確認を行った。

本論文では、複数ロボットの協調作業を実現するための経路プランニングに取り組んだ。これは複数ロボットによる作業を自動化し、その実用化に向けた一歩になるものと考えている。

## 7.2 将来の展望

計算機の大容量化、高速化のおかげで、単純なアルゴリズムを高速に実行することで、複雑な機能が実現できるようになってきた。本論文で用いている MCL (6.3.2 節) や RRT (1.5 節) もそのようなアルゴリズムの一種である。これらは疑似乱数を用いて解候補やサンプル状態を生成し、要求条件に合致するものを残すだけである。それだけのシンプルな手順から位置同定や経路プランニングが実現できる。これらはアルゴリズムが簡単であるためその高速化にはハードウェア化が有効かも知れない。

ロボット研究は総合的な研究であり、ハードウェアからソフトウェア、機械工学から人工知能、画像処理、音声処理など、情報処理の多くの分野が関連する。それだけに、各分野で取り組むべきことは多い。ロボットのハードウェアとしてはより軽くて丈夫な構造体や小さくて強力なモータなどが望まれる。ロボットが計画通りに正確に移動するためには精密なセンサなどで制御する必要がある。プランニングの処理スピードが遅いならば、専用チップを作成することでリアルタイムで処理できるだろう。複数のロボット環境でもリアルタイムでプランニングできるようになるかも知れない。ロボットが他のロボットや人のような移動する障害物を認識し、回避することはまだ難しい。これは環境中に多数のカメラを配置してそれぞれの動きを追跡することで解決できるかも知れない。これには高度な画像処理が必要である。

近年のロボットブームによって、数多くのロボットが市販されるようになった。ヒューマノイド型のロボットもあれば、エンターテイメント用途のロボットなどもある。日本では経済産業省による HRP プロジェクト [54] や NEDO による開発推進事業 [53] などを通じて、国家戦略としてヒューマノイドロボットの実用化を推進している。研究だけでなく実用化の取り組みがなされてきている。

ヒューマノイドロボットのような複雑なロボットでも、単に動かすだけならば比較的簡単にプログラムできるようになってきた。しかしロボットが人間にとって本当に役立つものとなり、人間の生活環境に普及するためにはさらなる知能化が必要である。

現在はまだヒューマノイドロボット 1 台につき人間が最低一人は付き添っていないと不安である。実用的と言えるためにはロボットが自律して人間の手助けなく安全に行動できることが望ましい。また、産業分野などでは生産性を上げるためにロボットが有用となるだろう。そのためには複数のロボットを一人で操作できることも必要であろう。本研究ではその基本的な機能の実現を試みた。

ロボットは最先端の分野である一方で、一般の人にも目的が理解しやすいため、研究開

発に過大な期待が寄せられているようにも感じられる。すべてを実現しようとするのではなく、有用性の高い技能・分野を見極めて研究開発を進めるべきと考えている。



# 謝辞

本論文の主査であり指導教員である伊庭斉志教授には心より御礼申し上げます。学士，修士，博士時代と長きにわたり研究を指導して頂き，さまざまな助言，提案を頂きながら，博士論文までやり遂げることができました。また何度も海外発表の機会を与えてくださり，何物にも代え難い貴重な経験をさせて頂きました。ヒューマノイドロボットというきわめて貴重な実験環境を自由にに使わせていただいたことにも深く感謝しております。

本論文の審査をしてくださった相田仁教授，相澤清晴教授，森川博之教授，峯松信明助教授，杉本雅則助教授には多くのご助言をいただきました。先生方のご指摘やアドバイスを頂いたことで本論文を最後までまとめ上げることができました。ここに厚く御礼申し上げます。

この研究を進める上では研究室の多くの方々にお世話になりました。伊庭研究室の秘書をして頂いた島津美和さんには事務手続きなどで長い間支えて頂きました。特に海外発表時など面倒な手続きにもかかわらず快くサポートして頂き大変感謝しています。

ヒューマノイドロボットを用いた実験に関しては，伊庭研究室で特にロボットグループだった方々，多くの卒業生に支えられました。協調型マルチエージェントシステムを作成された井上豊氏，卒業研究として Monte Carlo Localization を実装してくれた堀口研一君，そしてトルコからインターンシップのために来日し，画像処理プログラムを作成してくれた Cem Erdemir 君にはとりわけ感謝しています。彼らのおかげでヒューマノイドロボットを用いた実験システムの構築が大いに助けられました。そのほかロボットグループとして多くの方々が協力し作成してくださったヒューマノイドロボットの動作やシステムは，数多くの実験を支えてくれました。ここにロボットグループとして支えて頂いたすべての方の名前を挙げることで感謝の意を表します。劉 宏偉さん，井上 豊さん，青木 勝洋君，三橋 秀行君，峠 隆広君，野沢 康文君，Lukiyanov Maxim さん，伊東 和紀君，柳瀬 利彦君，Deniz Aydemir 君，島田 健太君，堀口 研一君，岡本 洋平君，本間 大輝君（順不同），以上の方々にお世話になりました。

実験の協力や研究生活，その他の面では研究室の多くの先輩，同輩，後輩に支えて頂きました。研究室でご飯を炊いて，実際に同じ釜の飯を食って過ごしたことも貴重な思い出です。留学生の Deniz Aydemir 君や Claus Aranha 君には海外発表のたびに英語のチェックをしてもらい大変感謝しています。そのほか個人的に支えて頂いた友人，そして長い学生生活を見守ってくれた両親に深く感謝いたします。

この研究は一部，日本学術振興会特別研究員 DC としての支援を受けています。

## 参考文献

- [1] Wolfram Beetz, Mark Moors, and Frank Schneider. Collaborative exploration of unknown environments with teams of mobile robots. In M. Beetz, et al., editors, *Plan-Based Control of Robotic Agents*, LNAI 2466, pp. 52–70. Springer, 2002.
- [2] K.E. Bekris, B.Y. Chen, A.M. Ladd, E. Plaku, and L.E. Kavraki. Multiple query probabilistic roadmap planning using single query planning primitives. In *Proc. of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, pp. 656–661, 2003.
- [3] Ohad Ben-Shahar and Ehud Rivlin. Practical pushing planning for rearrangement tasks. *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 4, pp. 549–565, 1998.
- [4] Ohad Ben-Shahar and Ehud Rivlin. To push or not to push: On the rearrangement of movable objects by a mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, Vol. 28, No. 5, pp. 667–679, 1998.
- [5] J. Bruce, M. Bolwing, B. Browning, and M. Veloso. Multi-robot team response to a multi-robot opponent team. In *Proc. of International Conference on Robotics and Automation (ICRA 2003)*, pp. 2281–2286, 2003.
- [6] James Bruce, Tucker Balch, and Manuela Veloso. Fast and inexpensive color image segmentation for interactive robots. In *Proc. of The 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2000.
- [7] James Bruce and Manuela Veloso. Real-time randomized path planning for robot navigation. In *Proc. of IROS-2002*, pp. 2383–2388, 2002.
- [8] S. Caselli and M. Reggiani. ERPP: An experience-based randomized path planner. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 1002–1008, 2000.
- [9] M. Cherif, M. Vidal, and Ch. Laugier. A roadmap-based algorithm for planning handling operations in changing industrial plants. *Research Report INRIA RR-3629*, 1999.
- [10] Joe Culberson. Sokoban is pspace-complete. *Technical Report TR 97-02, Dept. of Computing Science, University of Alberta*, 1997. also: <http://web.cs.ualberta.ca/~joe/Preprints/Sokoban>.

- [11] M. M. Curtiss. Motion planning and control using rrts. Master's thesis, Electrical Engineering and Computer Science Dept., Case Western Reserve University, May 2002.
- [12] Benoît Dacre-Wright, Jean-Paul Laumond, and Rachid Alami. Motion planning for a robot and a movable object amidst polygonal obstacles. In *Proc. of the 1992 IEEE International Conference on Robotics and Automation*, pp. 2474–2480, 1992.
- [13] Yusuke Fukazawa, Chomchana Trevai, Jun Ota, Hideo Yuasa, Tamio Arai, and Hajime Asama. Controlling a mobile robot that searches for and rearranges objects with unknown locations and shapes. In *Proc. of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems (IROS2003)*, pp. 1721–1726, 2003.
- [14] Ron Fulbright and Larry M. Stephens. Classification of multiagent systems. Technical Report ECE 06-94-02, University of South Carolina, June 1994.
- [15] <http://www.shiro.dreamhost.com/scheme/gauche/index-j.html>.
- [16] Fabien Gravot, Stephane Cambon, and Rachid Alami. aSyMov: a planner that deals with intricate symbolic and geometric problems. In *11th International Symposium on Robotics Research (ISRR'2003)*, Vol. 15 of *Springer Tracts in Advanced Robotics*, pp. 100–110. Springer Berlin, 2003.
- [17] <http://www.gnu.org/software/guile/guile.html>.
- [18] Jun Inoue, Akira Ishino, and Ayumi Shinohara. Ball tracking with velocity based on monte-carlo localization. In T. Arai, et al., editors, *Proceedings of Intelligent Autonomous System 9*, pp. 686–693. IOS Press, 2006.
- [19] S. Kagami, J.J. Kuffner, K. Nishiwaki, K. Okada, M. Inaba, and H. Inoue. Humanoid arm motion planning using stereo vision and RRT search. In *Proc. of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, pp. 2167–2172, 2003.
- [20] Shotaro Kamio and Hitoshi Iba. Real-time adaptation technique to real robots: An experiment with a humanoid robot. In *Proc. of 2003 Congress on Evolutionary Computation (CEC2003)*, pp. 506–513, Canberra, Australia, December 8-12 2003. IEEE Press.
- [21] J. Kuffner and S. LaValle. RRT-Connect: An efficient approach to single-query path planning. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA'2000)*, San Francisco, CA, April 2000., pp. 995–1001, 2000.
- [22] James Kuffner, Koichi Nishiwaki, Satoshi Kagami, Masayuki Inaba, and Hirochika Inoue. Motion planning for humanoid robots. In *Proc. 11th Int'l Symp. of Robotics Research (ISRR 2003)*, 2003.
- [23] S. M. LaValle. *Planning Algorithms*. [Online], 2004. Available at

- <http://msl.cs.uiuc.edu/planning/>.
- [24] S. M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In B. R. Donald, K. M. Lynch, and D. Rus, editors, *Algorithmic and Computational Robotics: New Directions*, pp. 293–308. A K Peters, Wellesley, MA, 2001.
  - [25] S. LaValle and S. Hutchinson. Optimal motion planning for multiple robots having independent goals, 1996.
  - [26] Joël Marchand. The algorithm by schwartz, sharir and collins on the piano mover's problem. In *Proceedings of the Workshop on Geometry and Robotics*, Vol. 391 of *Lecture Notes In Computer Science*, pp. 49–66. Springer-Verlag, 1988.
  - [27] K. Okada, A. Haneda, H. Nakai, M. Inaba, and H. Inoue. Environment manipulation planner for humanoid robots using task graph that generates action sequence. In *Proc. of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004. (IROS 2004)*, Vol. 2, pp. 1174–1179, 2004.
  - [28] Kei Okada, Masayuki Inaba, and Hirochika Inoue. Walking navigation system of humanoid robot using stereo vision based floor recognition and path planning with multi-layered body image. In *Proc. of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, pp. 2155–2160, 2003.
  - [29] Stasse Olivier, Jean Semere, Neo Ee Sian, Takashi Yoshimi, and Kazuhito Yokoi. Vision-based virtual information and semi-autonomous behaviours for a humanoid robot. In T. Arai, et al., editors, *Proceedings of Intelligent Autonomous System 9*, pp. 794–803. IOS Press, 2006.
  - [30] J. Ota, N. Miyata, T. Arai, E. Yoshida, D. Kurabatashi, and J. Sasaki. Transferring and regrasping a large object by cooperation of multiple mobile robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems 95*, pp. 543–548, 1995.
  - [31] Jun Ota. Rearrangement planning of multiple movable objects by using realtime search methodology. In *Proc. of the 2002 IEEE International Conference on Robotics and Automation*, pp. 947–953, 2002.
  - [32] Jun Ota. Rearrangement of multiple movable objects - integration of global and local planning methodology. In *Proc. of the 2004 IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 1962–1967, 2004.
  - [33] David Parsons and John Canny. A motion planner for multiple mobile robots. In *Proc. of 1990 IEEE Conference on Robotics and Automation*, pp. 8–13, 1990.
  - [34] S. Petti and T. Fraichard. Safe motion planning in dynamic environments. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.
  - [35] Kazuhiko Shinozawa, Takhiro Miyashita, Noriaki Mitsunaga, Ren Ohmura, and

- Norihiro Hagita. A prototype system for cooperation among heterogeneous robots. In *Proc. of 36th International Symposium on Robotics (CD-ROM)*, Nov. 2005.
- [36] T. Siméon, J-P. Laumond, and F. Lamiroux. Move3d: a generic platform for path planning. In *Proceedings of the 4th IEEE International Symposium on Assembly and Task Planning*, pp. 25–30, 2001.
- [37] Simon Thompson and Satoshi Kagami. Stereo vision and sonar sensor based view registration for 2.5 dimensional map generation. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, Vol. 4, pp. 3444–3449, 2004.
- [38] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. The MIT Press, 2005.
- [39] G. Wilfong. Motion planning in the presence of movable obstacles. In *SCG '88: Proceedings of the fourth annual symposium on Computational geometry*, pp. 279–288, New York, NY, USA, 1988. ACM Press.
- [40] Denis Wolf and Gaurav S. Sukhatme. Online simultaneous localization and mapping in dynamic environments. In *Proc. of the Intl. Conf. on Robotics and Automation (ICRA)*, 2004.
- [41] Ping Xuan and Victor Lesser. Multi-agent policies: from centralized ones to decentralized ones. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pp. 1098–1105, New York, NY, USA, 2002. ACM Press.
- [42] Kazuhiko Yokoyama, Hiroyuki Handa, Yutaro Hukase, Kenji Kaneko, Fumio Kanehiro, Yoshihiro Kawai, Fumiaki Tomita, and Hirohisa Hirukawa. Cooperative works by a human and a humanoid robot. In *Proc. of the 2003 IEEE International Conference on Robotics and Automation*, pp. 2985–2991, 2003.
- [43] 稲葉雅幸, 加賀美聡, 西脇光一. ロボットアナトミー. 岩波講座: ロボット学 7. 岩波書店, 2005.
- [44] 田中完爾, 木室義彦, 岡田伸廣, 近藤英二. 非静的環境における確率的アプローチによる変化検出と自己位置推定. 電子情報通信学会論文誌, Vol. J88-D-II, No. 5, pp. 854–863, 2005.
- [45] 堀口研一. 回転カメラによるモンテカルロ自己位置同定を用いたタスク支援. 東京大学工学部電子情報工学科 学士論文, 2006.
- [46] 堀口研一, 神尾正太郎, 伊庭斉志. 回転カメラによるモンテカルロ自己位置同定を用いたロボットのタスク支援. 情報処理学会 第 68 回全国大会講演論文集, 2006.
- [47] 木村公一, 村瀬良一. 「HOAP」シリーズの事業化. 日本ロボット学会誌, Vol. 22, No. 1, pp. 10–12, 2004.
- [48] 周洪鈞, 坂根茂幸. 準動的環境における移動ロボットの地図生成と位置決め—RFID

- タグとレーザレンジファインダの利用— 第 24 回日本ロボット学会学術講演会予稿集 CD-ROM, p. 1E12, 2006.
- [49] 木村浩. 作業移動型ロボットの協調制御. 日本ロボット学会誌, Vol. 13, No. 7, pp. 912–915, 1995.
- [50] 太田順, 新井民夫, 倉林大輔. 作業の性質を考慮したロボット群の動作計画. 日本ロボット学会誌, Vol. 13, No. 4, pp. 517–524, 1995.
- [51] 金原正則, 加賀美聡, James J. Kuffner, 溝口博. 車輪型ロボットのための余裕を持った障害物回避経路の生成. 第 24 回日本ロボット学会学術講演会予稿集 CD-ROM, p. 3E13, 2006.
- [52] 加賀美聡, Simon Thompson. 自律移動の高速化のための階層的な位置同定、地図作成、経路計画、経路制御システム. 第 24 回日本ロボット学会学術講演会予稿集 CD-ROM, p. 1E13, 2006.
- [53] 山本哲也. 次世代ロボット実用化プロジェクト 実用システム化推進事業について . 日本ロボット学会誌, Vol. 24, No. 2, pp. 151–152, 2006.
- [54] 井上博允, 比留川博久. 人間協調・共存型ロボットシステム研究開発プロジェクト. 日本ロボット学会誌, Vol. 19, No. 1, pp. 2–7, 2001.
- [55] 比留川博久. 人間型ロボットの近未来応用. 日本ロボット学会誌, Vol. 22, No. 1, pp. 6–9, 2004.
- [56] 井上豊. ヒューマノイドロボットを用いた協調型マルチエージェントシステムの構築に関する研究. PhD thesis, 東京大学新領域創成科学研究科, 3月 2005.
- [57] 井上豊, 峠隆広, 伊庭齊志. ヒューマノイドロボットによる協調搬送 — 目標位置を考慮した修正学習—. 第 22 回日本ロボット学会学術講演会予稿集, p. 1L35, 2004.
- [58] 深澤祐介, Trevai CHOMCHANA, 太田順, 新井民夫. 複数個のサブタスクを有する問題解決器のための中間ゴールの自律的生成. 第 21 回日本ロボット学会学術講演会予稿集, 2003.
- [59] 下笹洋一, 若林潔, 有木孝夫, 大町利夫, 末富大剛, 杉浦正則, 小谷健太郎. 警備ロボット “ガードロボ” による警備サービス. 日本ロボット学会誌, Vol. 24, No. 3, pp. 308–311, 2006.
- [60] 坂原洋人, 升谷保博, 宮崎文夫. 時空間 rrt による未知の動的環境における実時間軌道生成. 第 24 回日本ロボット学会学術講演会予稿集 CD-ROM, p. 3E26, 2006.
- [61] 上田隆一, 新井民夫, 浅沼和範, 梅田和昇, 大隅久. パーティクルフィルタを利用した自己位置推定に生じる致命的な推定誤りからの回復法. 日本ロボット学会誌, Vol. 23, No. 4, pp. 466–473, 2005.
- [62] 日浦亮太. wakamaru の挑戦. 日本ロボット学会誌, Vol. 24, No. 3, pp. 284–287, 2006.

# 発表論文

## 学術論文

1. 神尾 正太郎, 伊庭 斉志: “マルチエージェント協調作業のためのランダムサンプリングを用いた経路プランニングアルゴリズム”, 電子情報通信学会論文誌, **J89-D**, 2, pp. 250–260 (2006).
2. Shotaro Kamio and Hitoshi Iba: “Adaptation technique for integrating genetic programming and reinforcement learning for real robots”, *IEEE Transactions on Evolutionary Computation*, **9**, 3, pp. 318–333 (2005).
3. 神尾 正太郎, 三橋 秀行, 伊庭 斉志: “遺伝的プログラミングと強化学習の統合に基づく実ロボットの行動獲得”, 情報処理学会論文誌:数理モデル化と応用, **45**, pp. 134–143 (2003).

## 国際会議論文 (査読付き)

1. Shotaro Kamio and Hitoshi Iba: “Cooperative object transport with humanoid robots using RRT path planning and re-planning”, *Proc. of the 2006 IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS2006)*, Beijing, China, IEEE Press, pp. 2608–2613, (2006).
2. Shotaro Kamio and Hitoshi Iba: “Search algorithm of the order of object transportation by multiple robots”, *Proc. of the 2006 IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS2006)*, Beijing, China, IEEE Press, pp. 4097–4102, (2006).
3. Shotaro Kamio and Hitoshi Iba: “Object transportation using two humanoid robots based on multi-agent path planning algorithm”, *Proc. of The 9th International Conference on Intelligent Autonomous Systems (IAS-9)*, IOS Press, pp. 318–325, (2006).
4. Shotaro Kamio and Hitoshi Iba: “Random sampling algorithm for multi-agent cooperation planning”, *Proc. of the 2005 IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS2005)*, Edmonton, Alberta, Canada, IEEE Press, pp. 1676–1681 (2005).
5. Shotaro Kamio and Hitoshi Iba: “Incremental learning of a robot controller by

- means of genetic programming”, Proc. of Joint 2nd International Conference on Soft Computing and Intelligent Systems and 5th International Symposium on Advanced Intelligent Systems (SCIS & ISIS 2004), Yokohama, Japan (2004).
6. Shotaro Kamio and Hitoshi Iba: “Real-time adaptation technique to real robots: An experiment with a humanoid robot”, Proc. of 2003 Congress on Evolutionary Computation (CEC2003), Canberra, Australia, IEEE Press, pp. 506–513 (2003).
  7. Shotaro Kamio, Hideyuki Mitsuhashi and Hitoshi Iba: “Integration of genetic programming and reinforcement learning for real robots”, Proc. of the Genetic and Evolutionary Computation Conference (GECCO2003), Chicago, IL, USA, Springer, pp. 470–482 (2003).
  8. Shotaro Kamio, Hongwei Liu, Hideyuki Mitsuhashi and Hitoshi Iba: “Researches on ingeniously behaving agents”, Proc. of 2003 NASA/DoD Conference on Evolvable Hardware, Chicago, Illinois, USA, IEEE Press, pp. 208–217 (2003).

#### 国内会議論文

1. 神尾 正太郎, 伊庭 斉志: “RRT 経路プランニングアルゴリズムを用いたマルチエージェント物体再配置問題のプランニング”, 日本ロボット学会 第 24 回学術講演会予稿集 (2006).
2. 神尾 正太郎, 伊庭 斉志: “複数ロボットによる荷物搬送共同作業のための RRT 経路プランニングアルゴリズム”, 日本ロボット学会 第 23 回学術講演会予稿集 (2005).
3. 神尾 正太郎, 伊庭 斉志: “漸次的なシミュレータ構築に基づく実時間ロボット行動学習”, 日本ロボット学会 第 22 回学術講演会予稿集 (2004).
4. 神尾 正太郎, 伊庭 斉志: “ロボットの行動特性への実時間適応 — 遺伝的プログラミングと強化学習の統合 —”, 日本ロボット学会 第 21 回学術講演会予稿集 (2003).

#### 共著論文

1. 堀口 研一, 神尾 正太郎, 伊庭 斉志: “回転カメラによるモンテカルロ自己位置同定を用いたロボットのタスク支援”, 情報処理学会 第 68 回全国大会講演論文集 (2006).