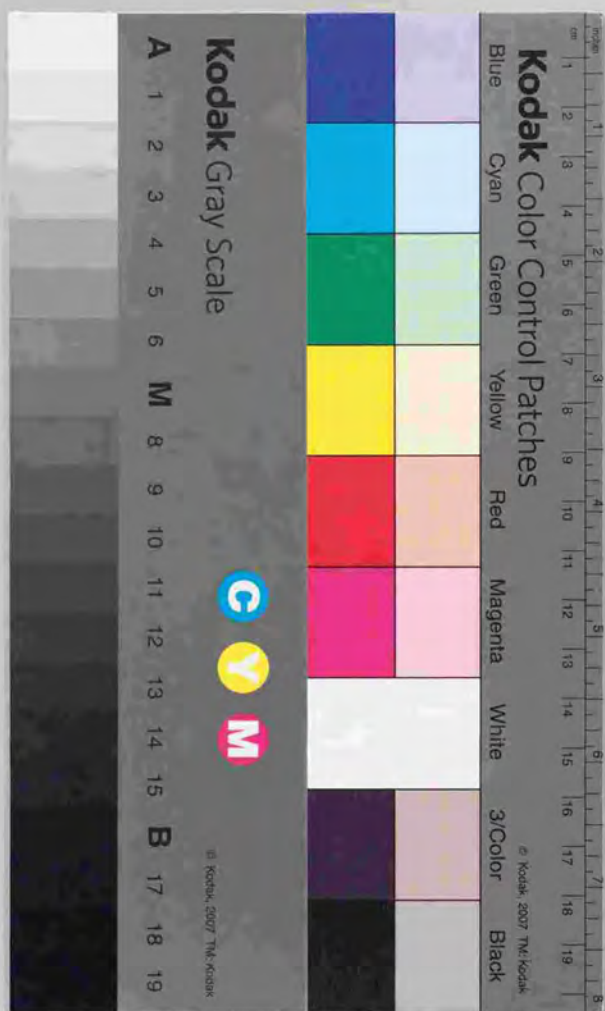


群ロボットによる異種作業割り付け型  
協調搬送システム

宮田 なつき



平成11年度 博士論文

群ロボットによる異種作業割り付け型  
協調搬送システム

指導教官 新井 民夫 教授

精密機械工学 専攻  
学生証番号 77081

宮田 なつき



## 概要

本論文では、未知環境における複数台の移動ロボットによる協調搬送システムを構築する。

近年、建設現場や大型構造物の組み立て工程における、多様な大型物体の搬送作業自動化の需要に対し、複数台の移動ロボットの協調による解決を試みる研究が多数行われてきた。その多くは、複数のロボットで共通物体を操作する、力の干渉を伴う協調制御問題としての側面のみを扱い、外部環境との関係に基づくロボット群の動的な行動計画については扱っていない。しかし、上記作業環境は動的な変化を伴う未知環境であり、障害物の発見、衝突回避、軌道確保のための障害物の移動など、物体のハンドリング以外にも環境情報の観測過程を含む種々の作業を実行する必要がある。そこで本研究では、未知環境における協調搬送システムの構築を目的とし、ロボットが必要に応じて実行するタスクを切替える“異種作業割り付け型協調搬送システム”を提案する。

このような協調システムは“情報層”、“幾何層”、“物理層”の3層からなる階層構造を成し、各層において協調が実現される必要があると考える。情報層は各ロボットの実行するタスクを、幾何層は周辺環境との幾何的な関係に基づく動作を計画する層であり、物理層はロボット間の力などの物理的な干渉を可能とするための層である。そこで本研究では、各層に対しそれぞれ課題を設定し解決を図ることとする。

まず、中間に位置する幾何層に対する課題解決として、種々の作業のうち協調ハンドリング作業の動作計画手法を構築する。その際、環境に対する柔軟性を向上させるため、ロボットによる把持位置変更動作を導入する。計画問題を一定時間毎に区切り、各周期における計画問題を対象物とロボットで分離して計画することで、環境変化に対応した動作が実時間で計算可能となる。その有効性を検証するため、シミュレーションを行う。

次に、最も下位に位置する物理層に対する課題解決として、走行誤差など、移動ロボットの機構に起因する問題を考慮し、実ロボットを用いて協調ハンドリング作業を実現する必要がある。その際、これらの問題による動作計画器や制御系の複雑化をさけるため、搬送機構の設計による解決方法と、動作計画における比較選択肢の制限の条件として扱う方法を提案する。上記で提案した動作計画手法とともに実装し、4台の移動ロボットからなる実機システムを構築して、協調ハンドリング作業実現を目指す。

さらに、異種作業割り付け型の協調作業を可能とするため、最も上位に位置する情報層に対する課題解決として、協調搬送作業のためのタスク割り当て手法の構築を取り上げる。この時、実行順序やタイミングという観点でのタスク間の相互関係を考慮し、実時間で決定する必要がある。これに対し、一定時間で1台のロボットに実行可能な作業量をタスクの単位と捉え、これをあるロボットが実行する優先度の決定方法を準備する。これにより、タスク割り当て問題が線形計画問題の一種として定式化され、相互関係を考慮した割り当て計画が実時間で可能となる。その有効性を検証するため、具体的なタスクを実装し、未知環境下での搬送シミュレーションを行う。

最後に、上記を統合したシステムを構築し、シミュレーションおよび実ロボットを用いたロボットにより、統合システムとしての有効性を検証する。

## 目次

1 序論	1
1.1 研究の背景	2
1.2 本論文に関連する従来研究	6
1.2.1 協調ハンドリング動作に関する研究	6
1.2.2 マルチエージェントのタスク割り当てに関する研究	10
1.3 研究の目的	16
1.4 論文の構成	17
2 持ち替えを伴う協調ハンドリング動作計画	19
2.1 はじめに	20
2.2 持ち替えを伴う協調ハンドリング問題	21
2.2.1 問題設定	21
2.2.2 協調ハンドリング問題の課題	23
2.3 提案する動作計画器の構成	24
2.4 対象物の動作計画	27
2.4.1 拡張仮想インピーダンス法	27
2.4.2 安定性を考慮したパラメータ設計	31
2.4.3 対象物の速度・角速度限界の求め方	36
2.5 移動ロボット群の動作計画	37
2.5.1 ロボットの動作計画の概要	37
2.5.2 持ち替え計画アルゴリズム	39

2.6 持ち替え協調ハンドリングシミュレーション	42
2.6.1 シミュレーションの設定	42
2.6.2 シミュレーション結果	44
2.6.3 持ち替え効果の評価	44
2.6.4 計算時間の評価	47
2.7 持ち替えコストの考慮	51
2.7.1 持ち替えコストを考慮した評価関数	51
2.7.2 持ち替えコストを考慮したシミュレーション	51
2.8 提案手法の適用可能な範囲	56
2.9 おわりに	58
3 実ロボットによる協調ハンドリング作業実現	59
3.1 はじめに	60
3.2 機構設計による問題解決	62
3.2.1 実ロボットの走行誤差問題への対応	62
3.2.2 非ホロノミック性への対応	62
3.2.3 提案する機構	64
3.3 移動ロボット群の動作計画アルゴリズムの拡張	65
3.3.1 トルク限界の考慮の仕方	65
3.3.2 各ロボットのハンドリング力の導出	66
3.3.3 トルク限界を考慮した持ち替え計画アルゴリズム	68
3.4 トルク限界を考慮した持ち替え搬送シミュレーション	69
3.4.1 シミュレーション内容	69
3.4.2 ハンドリング力限界の持ち替え計画への影響	72
3.5 協調ハンドリング実験システムの構築	76
3.5.1 把持機構の製作	76
3.5.2 車輪機構のモデル化および制御則	79
3.6 搬送機構の評価実験	81
3.7 持ち替え協調ハンドリング実験	83

3.7.1 障害物の存在しない環境における協調ハンドリング実験	83
3.7.2 障害物の存在する環境での協調ハンドリングシミュレーション	87
3.8 おわりに	89
4 異種作業割り付け型協調搬送システムの計画器アーキテクチャ	91
4.1 はじめに	92
4.2 タスク割り当てを含む行動計画器の課題	94
4.3 計画器アーキテクチャ	96
4.3.1 アーキテクチャの検討	96
4.3.2 提案アーキテクチャの概要	98
4.3.3 提案アーキテクチャによる問題解決	101
4.4 タスクの記述	102
4.4.1 タスクとは	102
4.4.2 タスクテンプレート	102
4.5 優先度を用いたタスク割り当てアルゴリズム	107
4.5.1 制約条件	109
4.6 タスクの実装	111
4.7 未知静的環境における協調搬送シミュレーション	123
4.7.1 シミュレーション内容	123
4.7.2 シミュレーション結果(時間経過)	125
4.7.3 シミュレーション結果のまとめ	136
4.8 未知動的環境を想定した作業への適用	137
4.8.1 動的環境への提案手法の適用	137
4.8.2 動的環境への対応のための今後の課題	141
4.9 提案手法の適用範囲の限界について	143
4.10 持ち替えを伴う協調搬送シミュレーション	146
4.10.1 シミュレーション設定	146
4.10.2 シミュレーション結果(時間経過)	147
4.11 おわりに	152



5 実ロボットによる異種作業割り付け型協調搬送作業の実現	157
5.1 はじめに	158
5.2 実験システムの全体の構成	159
5.3 各ロボットの構成	161
5.4 物体ハンドリング部	163
5.5 環境認識部	165
5.5.1 認識支援装置を用いた環境認識システム	165
5.5.2 相対位置計測アルゴリズム	166
5.5.3 マークの認識精度評価実験とロボットのセンシングエリアの特定	167
5.5.4 色識別精度評価実験	168
5.6 未知環境における協調搬送実験	171
5.6.1 実験内容と目的	171
5.6.2 実験結果(時系列)	171
5.6.3 実験結果の評価	173
5.7 実ロボットによる環境認識過程についての考察	178
5.7.1 観測動作の組み込み方について	178
5.7.2 タスク割り当てにおけるセンシング誤差の影響	178
5.8 おわりに	180
6 結論	181
6.1 結論	182
6.2 今後の展望	184
謝辞	187
参考文献	191
研究業績書	199

## Chapter 1

### 序論

---

1.1 研究の背景	2
1.2 本論文に関連する従来研究	6
1.3 研究の目的	16
1.4 論文の構成	17

---

## 1.1 研究の背景

近年、建設現場や大型構造物組み立て現場、あるいは配送センターにおける資材・荷の運搬や、原子力発電所のような極限環境におけるメンテナンス作業など、大型で多様な形状を持つ物体の搬送作業自動化の需要は大きい。クレーン等を用いた場合、搬送システム自体の設置に手間がかかることから、移動ロボットによる作業形態が注目を集めている。

多様な物体に対応するため、最も大型の物体に見合った大きさの移動ロボットを用い、単体で搬送を行う (Fig.1.1(a)) という形態が考えられるが、物体に対する無駄が多く柔軟性も低い。これに対し、比較的小型のロボットを複数台用い協調して共通の物体を搬送させる形態が考えられる (Fig.1.1(b))。この場合、ロボット間の力の干渉を考慮する必要があるが、ロボット同士の自由な位置関係が可能であることから、多様な形状の物体に柔軟に対応可能である。そのため通産省のIMS-MMHS (Intelligent Manufacturing System-Metamorphic Material Handling System) プロジェクトなど、実現に向けた研究も盛んに行われている。

本論文でも、大型で多様な物体の搬送システムを、複数台の移動ロボットの協調により実現することが有効である、という立場で以下の研究を進める。

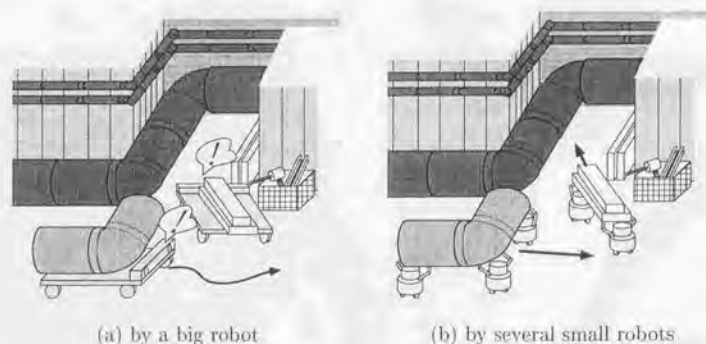


Fig. 1.1: Handling of Variable Materials

搬送システムの最終目的は、搬送対象物体をハンドリングし目標コンフィギュレーションを達成させることにある。そのため [Stilwell93][Kosuge96] など、これまで主に力の干渉を伴う対象物体のハンドリング、すなわち与えられた軌道への追従制御に関する研究がな

されてきた。しかし搬送システムの実現が求められる上記の作業環境は、構造物の組み立てや建設が進むにつれ、資材の一時的な設置や他の作業者の通行など、搬送システムが事前にその出現を感知できない障害物の存在する“未知環境”である。ロボットはその変化を観測し、周辺環境との関係に基づき適切な行動を選択する必要がある。その際、一般に各ロボットの観測能力が限られていることから、外部環境情報を取得する観測過程も、一つの作業と捉えるべき重要な過程であり、これをうまく計画に組み込む必要がある。

本論文では、この未知環境における複数台の移動ロボットによる協調搬送システムを扱う、議論を進めるにあたり、以下を想定する。

- ロボット群は数台～10台程度の小中規模群
- ロボット群の構成台数は一定 (所与のロボット群での問題解決を図り、他から援軍は呼ばない)
- 多自由度マニピュレータは持たない車輪型の移動ロボットによる、2次元平面内の運動のみによる物体のハンドリング動作を扱う。
- ロボットは限られた範囲内の観測が可能
- ロボットは同質で通信は自由
- 環境内には、ロボットが走行することを認知している作業者が数名、また作業を妨害する物体が数個存在する。

未知環境において搬送作業を行う場合、必要となる作業は対象物体の目標コンフィギュレーションを達成するハンドリング動作だけではない。搬送作業を妨害する物体の移動、ランドマークの計測など、その種類は様々であり、さらに個々の作業の質も異なる (Fig.1.2)。本論文では、これら主目的に付随する作業を含め、状況に応じて各ロボットの果たすべきタスクを切替え、協調して実行するロボット群全体を搬送システムとして捉えることを提案する。これが“群ロボットによる異種作業割り付け型協調搬送システム”である。

ここで、このような協調システムは、“情報層”、“幾何層”、“物理層”の3つから成る階層構造を成し、それぞれの層において協調が実現される必要があると考える (Fig.1.3)。“情報層”では、群内で共有する情報に基づき、どのロボットがどの作業を行うかが計画される。“幾何層”では、作業をどのように実現するかと言う具体的な動作が環境との幾何的な関係に基づき計画される。最後に“物理層”では、ロボット同士あるいは環境との力などの物理的な干渉 (協調) を可能とするため、走行誤差など実機に起因する問題が解決される層である。



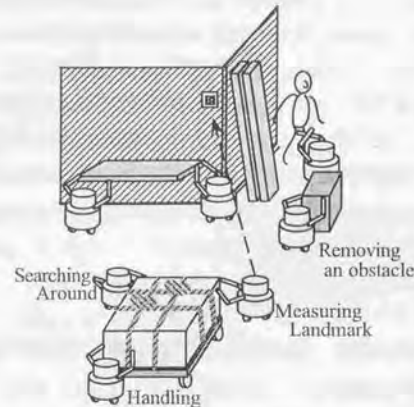


Fig. 1.2: Cooperative Transport in Unknown Environment

本論文では、未知環境における搬送システム構築のためには下記3点の解決が最も重要な課題であると考ええる。

まず情報層に関して、各ロボットの行うべき作業（タスク）を、観測された状況に応じてその場で決定する必要がある。また協調作業であることから、実行の順序やタイミングなどタスク同士の相互関係が深いという特徴があり、

- 協調作業のための実時間タスク割り当て手法

を構築する必要がある。

次に幾何層に関して、ここで扱う作業は種々あるが、搬送システムであることから、外部環境との関係に基づいた協調ハンドリング動作を計画できることが最も重要である。この時、環境に対する柔軟性を高めるため、ロボットが対象物のどこを把持するかということを含め、環境内での移動を計画する必要がある。したがって、

- ロボットによる把持位置変更を伴う協調ハンドリング動作計画手法

の構築が必要である。把持位置を変更することを本論文では以後“持ち替え”と呼ぶ。

最後に、本論文では、提案する力の干渉を伴う協調作業を、計算機上だけでなく実ロボットにより実現することを目指す。この時、周辺環境との幾何的な関係を考慮して動作が計画されるため、ロボットは位置制御ベースで動作することが望ましい。しかしその場

合、協調動作の実現過程において、ロボットにとって不可避である走行誤差が大きく影響することになる。また、移動機構によっては非ホロノミック性を有するなど、実機に起因する問題が種々存在し、これらが克服される必要がある。そのため物理層に関して、

- 実機問題を考慮した実ロボットを用いた協調ハンドリング作業の実現を課題と考える。

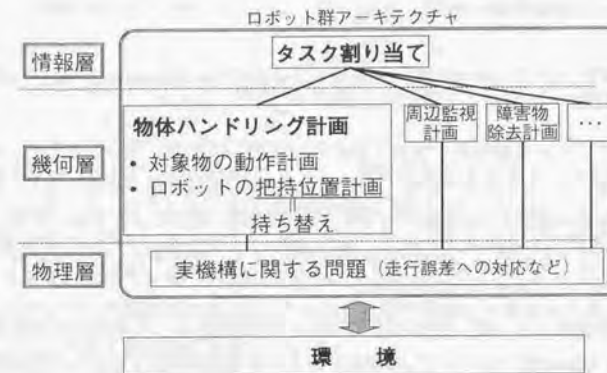


Fig. 1.3: Three-layered Architecture for Robots' Cooperation

## 1.2 本論文に関連する従来研究

本論文に関連した従来研究を

1. 協調ハンドリング動作に関する研究 (1.2.1 項)
2. マルチエージェントのタスク割り当てに関する研究 (1.2.2 項)

の二つにおいて整理する。

### 1.2.1 協調ハンドリング動作に関する研究

複数のロボットが共通物体を力の干渉を伴い協調してハンドリングを行う、いわゆる“協調搬送作業”というトピックを扱った研究について、把持位置の計画手法および実ロボットを用いた作業の実現という観点からその動向を整理し以下にまとめる。ただし対象とする協調ハンドリング動作は、平面内を車輪機構によって移動する、マニピュレータを持たないロボットによる平面内の運動とする。

なおここでは扱わないが、作業を行うロボットの種類によって次のような協調ハンドリングに関する研究がなされている：移動ロボットに多自由度マニピュレータを搭載した移動マニピュレータによるもの ([Khatib99], [Desai96])；不整地への適用を考慮した脚式ロボットを用いたもの ([木村97], [Hara99])。また、車輪型のロボットでも対応可能な程度の比較的緩やかな地面の凹凸に対する問題を扱ったものとして、原らによる [安井94] [篠島97]、大隅による [Osumi96] などがある。

#### 把持位置を固定した協調ハンドリングに関する研究

代表的な研究として、[Stilwell93]、[橋本95]、[小菅98]、[王93] などがある。

橋本らは [橋本92] [銭谷93] [橋本95] において、対象物を高速通信を可能とする有線通信系を内蔵したパレットに搭載し、搬送制御の形態として従来からマニピュレータの分野で行われているように、ロボット群全体を一つの系と見なし集中制御を行う立場をとっている。非ホロノミック性を有するロボットを用いて、搬送物の位置・姿勢を制御するため、搬送物とロボットとを1自由度の能動並進関節と受動回転関節から成る連結機構を用いて固く接続することを仮定している。また車輪制御に高ゲインの速度サーボ系を用いることを仮定している。しかしシミュレーションによる検証にとどまっており、実機による作業の実現はなされていない (Fig.1.4)。

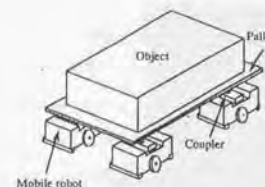


Fig.1 Overview of METROS

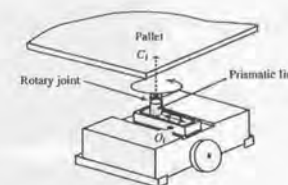


Fig.2 Schematic diagram of coupler mechanism

Fig. 1.4: Cooperative Transportation with Slider Joint [橋本92]

Stilwell らは [Stilwell93] [Stilwell94] において、複数の非ホロノミック性を有する移動ロボットを用いた、リーダーフォロワ型の分散協調搬送手法を扱っている。ロボットと対象物は受動回転関節で接続する事を仮定し、各フォロワロボットは力センサの値に基づき、それぞれキャスト状の動作を行うよう制御するというものである。これらの研究もやはり、シミュレーションによる検証にとどまっており、実機に起因する問題点は考慮されておらず、実ロボットによる検証はなされていない。

小菅らも [Kosuge95]、[Kosuge96]、[小菅98] において、複数の非ホロノミック性を有する移動ロボットによる、リーダーフォロワ型の分散協調搬送手法を扱っている (Fig.1.5)。対象物とトルクセンサを介して堅く接続された実ロボットを用い、1 kHz の高周期の制御で、与えられた軌道を追従する協調ハンドリング動作を実現している。

これらの研究では共通して、非ホロノミック性という実ロボットの移動機構に起因する問題点の一部を扱っている。しかし、協調制御理論に焦点が絞られているため、周辺環境との関係に基づく対象物やロボットの作業位置に関する考慮はなく、未知環境においてそのまま実現することはできない。

#### 把持位置計画を伴う協調ハンドリングに関する研究

協調ハンドリング動作における物体の把持位置 (ロボット配置) の計画 (“持ち替え” 動作) を扱った研究としては以下が挙げられる。

木村らは宇宙ロボットの押し動作による協調ハンドリングにおいて、押し動作を行う際の力の作用点 (把持位置) の選択方法として、可操作性楕円の体積の評価方法を作業内容に応じて切替えて用いる手法を提案している ([木村93])。作業内容と作用点配置の関係として、物体を長距離移動させる場合には任意の一定方向の加速度が出しやすくその他の



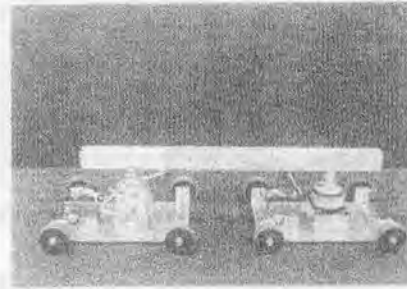


Fig. 1.5: Cooperative Transport with Force Feedback [Kosuge96]

方向の加速度は生じにくい配置を、物体の位置決めを行う場合にはあらゆる方向への加速度を出しやすい配置を求めている。同著者らは[W93]において実ロボットを用い実機検証を行っている。

また、佐々木は[佐々木99]において、対象物の重心位置を推定しロボットへの荷重配分が最適となる把持位置を求める手法を扱っている。

しかし、これらの研究では周辺環境は考慮されておらず、搬送作業が開始した後の把持位置の変更については扱っていない。

吉川らによる[Yoshikawa96]では、把持位置を変更する手順を重視した動作計画手法を提案し、シミュレーションにより検証を行っている。3台のロボットが物体を支持し残り1台が物体から離れて把持位置を変更するなど、多指ハンドにおける持ち替え動作に設定に近いが、ロボットは4台など適用台数に限界がある。

多指ハンドの分野における持ち替え計画としては、例えば[小俣94]があり、探索木を用いた多指ハンドによる多角形物体の姿勢変換法について議論している。この手法では4本以上 $n$ 本の指が想定され、安定把持を保つ範囲内であれば一度に離し得る指が2本以上の場合にも対応することが可能となっている。

しかし、これらの手法では周辺環境との関係に基づいた対象物やロボット位置の計画は抜かれておらず、また後者は計算時間がかかるため実時間計画には向いていない。

周辺環境を考慮したものとして、複数マニピュレータの協調による持ち替え動作を含

んだ動作計画アルゴリズムを扱った[Koga92]および[Koga94]がある。対象物をマニピュレータで固定把持した状態での移動と、対象物を地面に置いた状態での把持状態の変更(持ち替え)とを、交互に繰り返すことによって最終的に対象物の目標コンフィギュレーションが達成される。計画には、全ての環境をコンフィギュレーション空間として表現し、ランダムイズドポテンシャル法を用いてこれを探索するという手法をとる。すなわち作業環境についての大域的な情報を必要とし、計算時間もかかるため、本論文で扱う未知環境における動作計画にこれらの手法を適用することはできない。

これに対し[Ogasawara92]では、3台の全方向移動ロボットによる持ち替え動作を伴う協調ハンドリング動作計画が、ある瞬間においては1台のみの計画を行うという分散実時間計画により、シミュレーション上で行われている。しかし、他のロボットは計画を行ったロボットの意図を実現すべく協調して動作するという假定や、動作方向を8方向のいずれかのみより選択するなど、制限が多い。またロボットの走行誤差等の実機の性質は考慮されていないため、実ロボットシステムにそのまま適用することは困難であると考えられる。

以上より、従来の協調ハンドリング動作に関する研究の動向を次のようにまとめることができる。

- 協調ハンドリング動作に関する研究は、把持位置を固定し高周期での力制御を必要とするものが多く、周辺環境との関係を考慮した対象物やロボットの計画を扱っていない。
- 把持位置計画手法の多くは、力学的つり合い等を考慮したものであり、周辺環境との関係を考慮した対象物やロボットの計画を扱っていない。
- 周辺環境との関係を考慮した対象物やロボットの計画を扱ったものはほとんどなく、あっても大域的な情報を必要とし計算時間がかかる、あるいはロボット動作に制限があるなど、実移動ロボットによる実現が難しいものとなっている。



## 1.2.2 マルチエージェントのタスク割り当てに関する研究

各ロボットの行動戦略の切替えもタスク割り当ての一つと捉えることができる。そこで、マルチエージェントの協調行動の計画に関する研究についてまとめる。

[Parker98]によれば、マルチエージェントの協調行動に関する研究には、大きく分けて (a)swarm-type cooperation と (b)intentional cooperation の二つの流れがある。

一つ目の (a)swarm-type cooperation とは、非常に多数の同質なロボットから成る群において、分散的に制御される個々の振舞いをうまく設計することで、群全体としての望ましい挙動を得ようとする試みである。神経生物学、動物行動学、心理学、社会学等に基づく手法が多く、[Mitsumoto97][Fukuda97][Genovese92]などがある。これらの手法は、空間内への分散や群あるいは群隊形の形成など、タスク達成までの時間に厳しい制約のない場合には有効だが、時間制約があり効率の重視されるタスクに適用することは難しい。

もう一つの流れは、効率を重視する (b)intentional cooperation を扱ったものである。

まずタスク割り当てに関する研究は、古くから分散人工知能の分野において盛んに行われてきた（詳しくは[石田96]）。この分野における手法は、主に複数の計算機上のプロセス間の協調手法として適用されてきた。代表的なものに契約ネットが挙げられる。

同分野で提案された種々の手法単体では、センサ等を搭載した実体を持つロボットの特徴が反映されていないため、協調動作の実現に直接適用することはできない。しかし、複数ロボット間の協調動作のための枠組の一部として、同様の仕組みを用いているものは多い（[Noreils93], [Caloud90], [Vidal96], [Asama92]）。

Noreils は [Noreils93] において契約ネットを用いたタスク割り当てを伴う協調動作計画手法を扱っている。リーダーとなるロボットはタスクをサブタスク列に分解し、どのような手順でどれだけのロボットが必要かを提示し、これに対する他ロボットからの提案に基づいて入札を行う。割り当てたのち、各ロボットで具体的な実行計画(local plan)をたてるが、計画を少し立てては他ロボットにその結果を通知し、計画上で衝突・競合が発生した場合は、そのロボット同士でさらに密な通信を行ってこれを調整し、最終的な調整プロトコル (coordinated protocol) を作成する。coordinated protocol とは、条件とそれが満たされた時になすべきサブタスクから成る作業状態の遷移グラフ状のものであり、ここでは Predicate/Transition nets で記述している (Fig.1.6)。ロボットは coordinated protocol を使用してタスクを実行し、必要な段階で他ロボットと同期をとることで協調を実現している。有効性を示す検証実験として、2台のロボットによる convoy (護送) および壁際への箱移動作業 (1台が押す役、もう1台が距離を計測する役) をリーダーフォロワ型の協調

により実現している。

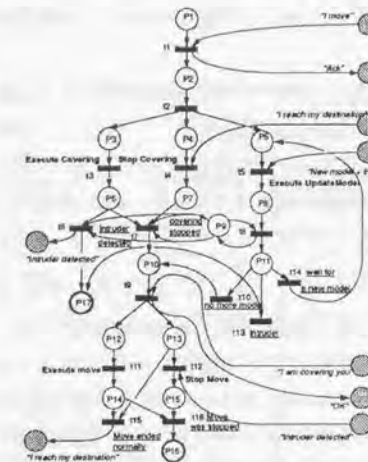


Fig. 1.6: Coordinated Protocol [Noreils92]

[Caloud90]でCaloudらにより提案された手法も、基本構造はNoreilsらの手法に似ており、タスクの割り当ては中央部が契約ネットを用いて行う。ただし、ここで想定されているのは、屋内オフィス等の環境における移動ロボットによる作業である。すなわち、飲料品や郵便物等の配達、コピー機等の操作、清掃といった、比較的小さく多量の物品の搬送作業あるいは移動を主とするタスクであり (GOFER Project)、ロボット間の協調は衝突回避等、遠慮的協調 ([Yuta92]) と呼ばれるものが主となる。そのため、契約ネットを用いて割り当てを統括するのは中央部となる。中央部は全体のおおまかな作業進行状況を把握しているため、各ロボットの提示してきた中から最適な計画を選択するか、あるいは多少の時間を待機し新たなロボットから提示されるはずの計画を待った方が良いか、という点に関して、ある程度見積もることが可能となっている。coordinated protocol の記述に、ここではベトリネットを使用している。

また Vidal らは、Caloudらと類似した目的として港湾における大量のコンテナ搬送作業 (MARTHA Project) を想定したタスク割り当て手法を扱っている ([Vidal96])。ここでも同様に、各ロボットが一つのタスク (コンテナの目的倉庫までの搬送) を終えると、



中央の管理機構に次に果たすべきタスクを問い合わせる自己動作を計画するが、中央では同じ交差点等に近い時刻に侵入する際に、これらが干渉しないよう最低限のスケジューリングにより適宜ロボットを停止させ、その侵入のタイミングを調整する(Plan merging)という手法をとっている。

さらに浅間らは[Asama92]において、ACTRESSと呼ばれる通信を用いた交渉に基づく分散的なタスク割り当て手法を提案している。ただし、ここでは中央管理機構にあたる存在はなく、各ロボットが独自にタスクを発見し、自分だけではこれを解決不可能な場合には、そのロボットが上記研究におけるリーダーとして他のロボットに対し協力を要請するという枠組である。検証を2台のロボットを模した計算機による荷片付け作業シミュレーションによって行っている。

これらの手法は、主に大量に存在する荷の片付け問題など、次のような作業に対し有効である。一つは、タスクを発見してから実行にうつすまでにそれほどきつい時間的制約がない作業である。また、ロボット群に対し一旦決めたタスク割り当てを途中で変更する必要がなく、最低限の遠慮的協調に基づく動作のみで協調関係が満たされるなどタスク同士が比較的独立している作業に対しても有効と考えられる。

しかし、協調ハンドリング作業のような問題では、例えば他のタスクを終え暇になったロボットを大型物体のハンドリングを行う群に投入することで各ロボットの荷重の減少など利益を得る場合が多い。これらの手法では、計画を練り直し再度契約ネットを用いて割り当て、互いの実行スケジュールを調整することになるため、実時間で行う事は難しい。すなわち、一旦割り当てたタスクの内容や、それを行うロボット群の構成が頻繁に変化する場合に、これらを直接用いる事はできない。

これに対し、Brooksにより提唱されたビヘイビアベースロボット([Brooks96])を用いた、協調作業のためのタスク割り当てに関する研究がいくつか行われている([Parker98], [Shibata96], [Jung99])。これは、前述の研究におけるベトリネット等のCoordinated protocolにおける状態を行動の選択肢と捉え、センサ情報あるいは通信に基づき適していると判断した行動を選択する手段と捉える事ができる。

Parkerは[Parker96][Parker97][Parker98]において、故障や失敗に対する頑健性の高いタスク割り当てシステム(ALLIANCE Fig.1.7)を提案している。ALLIANCEでは、動作に完全な同期のとられていない異質なロボット間の協調行動を扱っている。例えば、脚型ロボットによる蹴り動作や車輪型ロボットによる押し動作など、異なる方法で“移動”という効果を達成する複数台のロボットによる箱押し作業などが可能となっている。各ロ

ボットはタスクを選ぶと、周囲に自分がどのタスクを行っているかを通知(broadcast)する。他のロボットは、通信内容と環境状態の変化から観測される作業の達成度により、失敗がある場合にはこれを検知し、代わりにそのタスクを実行する事が可能となっている。

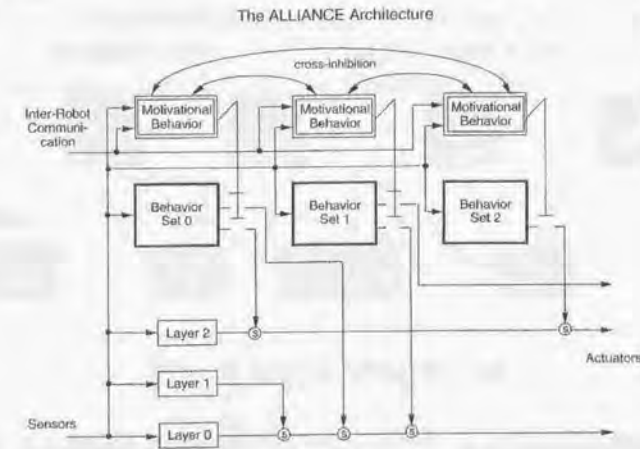


Fig. 1.7: ALLIANCE [Parker96]

しかし適用対象となっているタスクは箱押し作業に代表される互いの関係が緩い(loosely coupled)ものであり、同時に動く/止まるといった状態や、実行の順序を確実に保証することはできない。多数のタスクが検知された時にどのタスクから選択・実行していくか、という問題に対しても、いくつかの戦略をケーススタディによって比較および評価しているものの、枠組として有効な選択戦略は示されていない。類似した手法として[Shibata96]等もあるが、同様の問題が生じる。

[Jung99]では、逆にタスクの実行順序を重視した分散的協調行動選択手法を提案している。ベトリネット等の状態遷移の接続関係を学習により動的に変更し得る、という手法であり、ロボットのとり得る全ての行動要素はABBAと呼ばれるネットワークで表現される(Fig.1.8)。ここでは各行動が“相関”あるいは“前提”といった関係で結合されており、各瞬間において選択されるのはこのうちの一つの行動のみとなっている。“前提”が全て満たされている時に初めてその行動が発火するという条件が記述されるため、実行順序を

厳しく守る場合には、その前段階にあたる行動が発火していることを“前提”にする、という手段をとることができる。しかしロボット群内での協調関係に関しては、他のロボットが何をなそうとしているかを観察あるいは通信によって知り、基本的に同じ状態をとらないために、必ず異種作業を実行する場合、あるいは異質なロボット間の協調には向いているが、他のロボットと同時に同じタスクを行う等の協調行動を記述することは難しい。

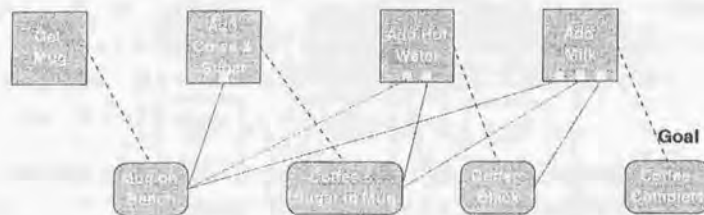


Fig. 1.8: ABBA network [Jung99]

國吉らは[Kuniyoshi94]において、ビヘービアベストロボットの枠組の中で、他ロボットの行動を視覚により認識して、それを支援する自己の協調行動を決定するという「観察に基づく協調」に関する研究を行っている。他を支援する基本協調行動パターンとして、“複雑環境中での追従行動 (Posing)” “障害物排除行動 (Unblocking)” “運搬引き継ぎ行動 (Passing)” の3つを、実ロボットによりそれぞれ個別に実現はしているが、いくつかの状況が組み合わされた場合にどの協調行動をとるのが適しているか、といった判断まではなされていない。

また、割り当て決定手法に関しては、ジョブショップスケジューリングの分野で多くの研究がなされている（種々の手法の詳細については[園川96]を参照）。ここでは大規模な問題を工程順序等の拘束条件を厳しく守った上で最適な計画を行うことが可能である。しかし多くの場合、タスクの変動に関しては扱っておらず、事前計画という立場で行われているものがほとんどであり、計算時間および動的な変動への対応という観点で、移動ロボットの協調システムに直接適用する事はできない。

以上より、従来のタスク割り当て手法の特徴を以下のようにまとめる事ができる。

- タスク割り当て決定に要する時間が実行に際しそれほど厳しい問題とならない荷片

付け問題等を対象としたものが多く、必要なタスクの動的な変化に対し実時間で対応することが難しい。

- タスクの割り当てが実時間で可能なものは、実行順序等、タスク間の相互関係の記述が難しい。

すなわち、搬送対象物を把持してハンドリングを行う協調搬送作業のように、実行順序に制約があり、なおかつ実時間で決定し実行する必要がある作業に、これらを直接適用することはできない。



### 1.3 研究の目的

従来研究を概観した結果、未知環境における協調搬送システム実現に必要な課題を解決する要素研究はなされていないことが分かった。

そこで、本研究の目的を

#### 未知環境における複数台の移動ロボットによる協調搬送システムの構築

とする。これに対し、本論文では“異種作業割り付け型協調搬送システム”を提案する。実現のための具体的な課題として、協調システムを成す各階層それぞれに対する課題である以下3点の解決をめざす。

- タスク間の相互関係を考慮した協調作業のための実時間タスク割り当て手法の構築
- 持ち替え動作を伴う協調ハンドリング動作計画手法の構築
- 実機問題の考慮に基づく、協調ハンドリング作業実現

### 1.4 論文の構成

1章では、未知環境における柔軟な協調搬送システムの必要性と、これに関連する従来の研究について示し、問題点を考察した。また、本研究の目的を述べた。

本研究では、まず幾何層に属する課題である協調ハンドリング動作計画手法の構築から始め、その実現のための物理層の課題を解決する。

2章では、持ち替え動作を伴う実時間協調ハンドリング動作計画手法を提案し、アルゴリズムの有効性をシミュレーションにより検証する。

3章では、2章で構築した計画に基づき実ロボットが行動し力の干渉を伴う協調作業を実現するために解決されるべき実機に起因する問題を、機構およびアルゴリズム面において考慮し、4台の実ロボットを用いた協調ハンドリング実験により実機適応性を検証する。

次に4章および5章では、上記で扱ってきた協調ハンドリング動作をタスクの1つと捉えた、異種作業割り付け型搬送システム全体の枠組を扱う。

4章では、情報層に属する課題解決として、目的達成に必要な様々なタスク間の相互関係を考慮し、タスクの割り当てを実時間で決定し動作を計画する計画器アーキテクチャを提案し、その有効性をシミュレーションにより検証する。また、前章で提案した持ち替え動作を伴う協調ハンドリング動作計画器を組み込んだシミュレーションを行い、総合システムとしての有効性についても検証する。

5章では、異種作業割り付け型協調搬送システムの枠組の実機適応性を検証するため、2台の移動ロボットを用いて未知障害物存在下における協調搬送実験を行う。

最後に6章において、本研究の結論と今後の展望について述べる。

本論文の構成をFig.1.9に示す。

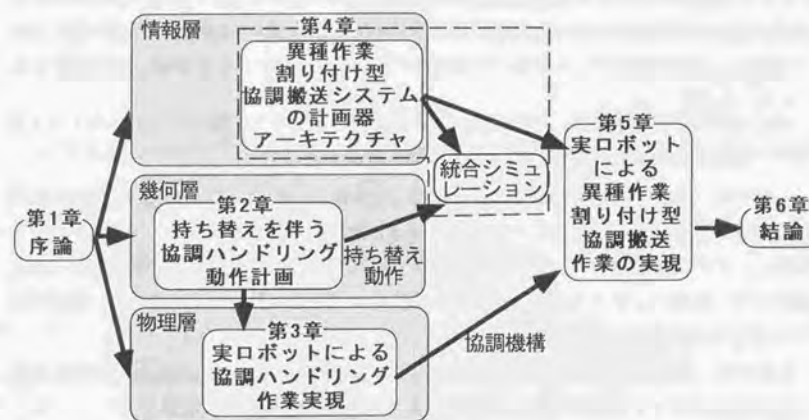


Fig. 1.9: Schematic view of this thesis

## Chapter 2

### 持ち替えを伴う協調ハンドリング動作計画

2.1 はじめに	20
2.2 持ち替えを伴う協調ハンドリング問題	21
2.3 提案する動作計画器の構成	24
2.4 対象物の動作計画	27
2.5 移動ロボット群の動作計画	37
2.6 持ち替え協調ハンドリングシミュレーション	42
2.7 持ち替えコストの考慮	51
2.8 提案手法の適用可能な範囲	56
2.9 おわりに	58



## 2.1 はじめに

本章では、幾何層に対する課題解決である“把持位置変更を伴う協調ハンドリング動作計画手法の構築”を目指す。

未知環境における協調ハンドリング動作計画では、対象物が環境内でどのような経路をとるか、またその動作をロボットによってどのように実現するかの二点を、ロボットによって得られる対象物周辺の局所的な環境情報に基づき、実時間で求める必要がある。

本章ではまず、動作計画問題を扱う上での前提条件等を示して協調ハンドリング動作の計画問題で扱うべき性質を明らかにし(2.2節)、動作計画器の概念設計を行う(2.3節)。これを構成する動作計画アルゴリズムの詳細を説明し(2.4節、2.5節)、提案するアルゴリズムの有効性を、持ち替えを伴う協調ハンドリング作業シミュレーションによって検証する(2.6節)。次に2.7節において、作業中の持ち替え動作実現コストが高い場合への対応を考慮し、シミュレーションによる検証を行う。最後に2.8節において、提案したアルゴリズム適用範囲の限界及び拡張方法について触れる。

## 2.2 持ち替えを伴う協調ハンドリング問題

### 2.2.1 問題設定

まず本論文では、複数台の移動ロボットが未知環境において剛体対象物を“持ち替え”ながら物体をハンドリングする、という問題を扱う。計画に用いることのできる情報は各ロボット周辺の情報であり、各ロボットへの動作指令は“速度指令”および“持ち替えるか否か”の2種類となる。

持ち替え動作の実現方法として、[Yoshikawa96]では多指ハンドの分野における持ち替え方法を導入し、対象物を一旦止めた状態で、動かす事の可能なロボットの把持状態を変えろという方法をとっている。しかし、持ち替えのために対象物から大きく離れ別の把持位置まで移動することは、移動ロボット自体の位置決めという観点から困難であると考えられる。また1つの持ち替え動作を完了するまでに時間がかかるため、変更を行っている間に周辺の状況が変化する場合も考えられる。したがって本論文では持ち替え動作を

ロボットがハンドリングを行いながら、対象物を支えたまま把持位置を変更することと定義する。

アルゴリズムの提案にあたり、以下の仮定を設ける。

1. 各ロボットは外界センサにより自分自身を中心とした一定半径内の障害物までの距離が計測可能。
2. ロボット間では自由に情報交換が可能。
3. 各ロボットは全方向に移動可能。
4. 各ロボットは、対象物の形状と重心位置をあらかじめ認識している。
5. 対象物はロボットと比較して大型で凸。
6. 各ロボットは対象物の周縁部を把持する。
7. 各ロボットは水平面内で有限の2次元の力を対象物に対し与えることができる。
8. ある瞬間においては、2台以上のロボットが物体を堅く把持して対象物の運動を実現し(把持ロボット)、残りのロボットが持ち替え動作を行う(持ち替えロボット)。

7, 8 番目の仮定は, 1 台のロボットの手先だけで大きな力およびモーメントの3自由度を制御して対象物をハンドリングすることが現実的でないことから置いた。これに関連して, ロボットが水平面内の対象物の運動を実現するために対象物に与える水平面内2次元の力を以後ハンドリング力と呼ぶ。

なお, 本論文では可搬重量を満たしている場合のみを扱い, 荷重の大きさについては考慮しない。

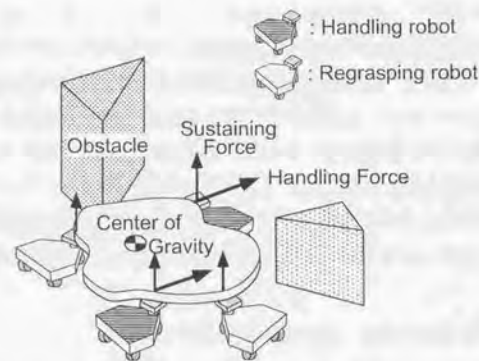


Fig. 2.1: Problem Settlement

### 2.2.2 協調ハンドリング問題の課題

複数台の移動ロボットにより一つの物体を協調してハンドリングする際の動作計画に対する要求はTable 2.1のようにまとめることができる。同表から分かるように, 協調ハンドリング作業に対する要求は多数あり, それら複数の要求の間にはトレードオフが存在する。また, 対象物とロボットの両方に対し要求されるものと, どちらか一方にだけ要求されるものとが混在しており, その関係が複雑である。そのため, 計画問題は非常に複雑であり, 実時間で解決するには問題の構造化が必要となる。

Table 2.1: Demands for the planning

demands	object	robot group
to reach goals	○	×
to avoid obstacles	○	○
to keep stable grasping	×	○
to keep handling energy small	×	○
to keep velocity limit	×	○

○: necessary      ×: not necessary

この他, 実ロボットで作業を実現するためには, 各ロボットのトルク限界による生成可能な運動の制限や, 車輪型ロボット特有の非ホロノミック性による運動方向の制限等の条件を考慮する必要があるが, これらについては作業を行うロボットの種類や個体差に依存する項目であるため, 3章において扱う事にする。

また, 「各ロボットの荷重をなるべく均等に配分する」といった要求も考えられるが, 本論文では荷重限界は十分満たされた状態を想定し, 荷重配分については言及しない。

さらに「持ち替え動作をなるべく少なくする」という要求も考えられる。これについても走行精度といった各ロボットの性能に依存する項目であるため, まずは持ち替えコストが十分低いという前提で理論を構築し, 持ち替えコストの高い場合については2.7節で触れる。



### 2.3 提案する動作計画器の構成

環境がロボットにとって相対的に変化する場合、搬送対象物が目標コンフィギュレーションに到達するまでに起こり得る状況をすべて予測することは不可能である。したがって動作計画は、その場で得られた局所的な情報に基づき、逐次的になされることが望ましい。

障害物の存在する空間におけるロボットの動作計画を行う代表的な手法として、コンフィギュレーション空間として表現した作業環境を小さい区画（セル）に分解したものを用いる手法が考えられる。現在の対象物およびロボットのコンフィギュレーションから遷移可能なあらゆる状態をこの空間上で表現し、セル毎に障害物との干渉をチェックを行って、衝突の無い状態をすべて評価・比較し、もっとも評価がよく、なおかつゴールに到達する計画を採択する、という手法である。しかし、持ち替えるロボットの台数が多くなればなるほど、あるいはロボットが把持位置を変更する刻み幅を細かく評価すればするほど、探索すべき状態が爆発的に増加するため、実時間計画には適していない。[福地99]でも、同様の手法を用いて持ち替えを伴うハンドリング計画を行っているが、オフライン計画と位置付けている。

そこで本論文では、2.2節に示した計画問題を解決するために、対象物の動作計画とロボットの動作計画という二つの部分問題に分離し、一定単位時間毎に

**Step 1** 環境情報の取得・融合

**Step 2** 対象物の動作計画

**Step 3** ロボットの動作計画

という3つの手続きをシーケンシャルに繰り返すものを提案する（Fig.2.2）。一定単位時間をここでは“動作計画周期”と呼ぶ。

すなわち、ある時刻 $t$ における**Step 1**において各ロボットが環境情報を取得し通信によってそれを融合し、**Step 2**において環境情報に基づいて対象物の運動を、**Step 3**においてその対象物の運動を実現するロボットの動作を計画する。

このアプローチの特長は以下の二つである。

- 対象物と移動ロボット群とに対しそれぞれ異なる要求を、分離して扱うことができるため、問題の単純化が期待できる。
- 対象物の動作計画は、対象物周辺の局所情報に基づいて計画を行う局所動作計画器を用いて行い、ロボットの動作計画は、対象物座標系におけるロボット配置に関す

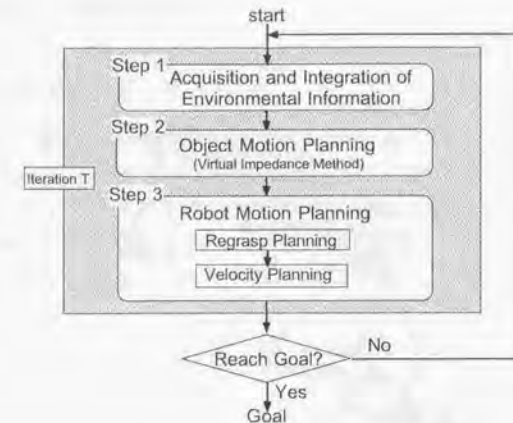


Fig. 2.2: Conceptual Design of Cooperative Handling Planner

る制約条件付非線形最適化問題として解く、という組合せにより、対象物及びロボットの把持位置の最適計画を両立させることができる。

この際、動作計画をロボットと対象物で分離するデメリットとして、それぞれの動作計画の結果が不整合で実現不可能とならないために、別途拘束条件を考慮しなければならないという点があげられる。しかし、分離した結果扱う変数が減り動作計画が簡単になるというメリットの方が大きいことから、提案手法ではこれを分離するアプローチを選択した。

この場合、別途考慮しなければならないのはロボットの速度限界（Fig.2.3）である。考慮の方法として、全てのロボットが、どの位置で、どのような動きをしてもこれを満たすように、十分緩やかな対象物の運動を計画するという対処方法がある。しかし対象物の動作を必要以上に制限することは、ハンドリング作業の柔軟さおよび効率低下を引き起こし好ましくない。

そのため、対象物の動作計画では、ロボットの動作計画において少なくとも1つ解の存在するような最低限の条件を与え、詳細はロボットの動作計画で考慮することにする。条件の与え方については、2.4.3項において説明する（Fig.2.4）。

以降、2.4節において対象物の動作計画器の詳細を、2.5節においてロボットの動作計画器の詳細を説明する。

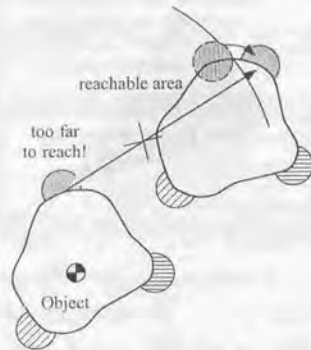


Fig. 2.3: Reachable Area of Robot

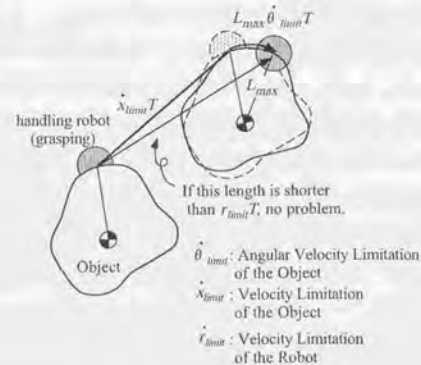


Fig. 2.4: How to Limit Object Velocity

## 2.4 対象物の動作計画

## 2.4.1 拡張仮想インピーダンス法

対象物の動作計画では、対象物周辺の局所的な情報に基づき、未知障害物を回避する経路を実時間で計画する必要がある。そのため本論文では、複数移動ロボット系のための実時間動作計画手法の一つである仮想インピーダンス法[新井93]を用いる。同手法では計画対象であるロボットとして円形を想定し質点として扱っていたが、ここでは対象物の姿勢も考慮するため剛体モデルに拡張する (Fig.2.5)。

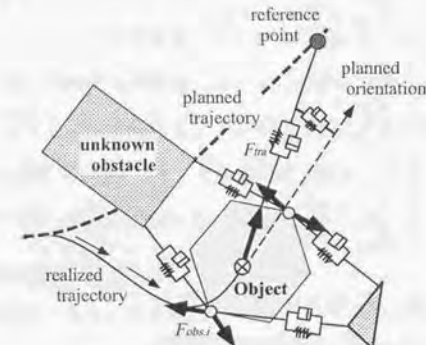


Fig. 2.5: Schematic View of Extended Virtual Impedance Method

この方法は、計画対象である対象物(object)と、それを把持するロボットにより認識できるセンシングエリア内の障害物(obstacle)との間、また対象物の位置・姿勢と対象物の参照位置・姿勢(reference position)間に仮想的なバネ・ダンパを張り、これらにより生成される仮想力、仮想モーメントから生じる運動により対象物の動作を計画する方法である。

このとき一定動作計画周期毎の対象物の参照位置・姿勢、参照速度・角速度は、既知障害物を回避する軌道を搬送以前にあらかじめ計算しておくものとする。既知障害物が特に存在しない場合は、スタート地点とゴール地点とを直線で結び、これを目標到達時間で単純に割ったものを与えれば良い。

障害物からの仮想力の作用点は、人工ポテンシャル法において提案されている“制御点”



の概念を導入し [Latombe91], 対象物上のいくつかの代表点を選んで, これに対してそれぞれ障害物からの斥力を付与することにする. 制御点の概念とは, 一般のポテンシャル法において, 計画対象上のいくつかの特徴点 (多角形ならばその頂点など) にポテンシャルを作用させるというものである. これにより, 外力の作用点が次々に切り替わることによる振動の影響を減じることができる.

以下に対象物の運動方程式を示す.

$$\ddot{\mathbf{x}} = \frac{1}{M_v} (\mathbf{F}_{tra} + \sum_i^{n_{obs}} \mathbf{F}_{obs,i}) \quad (2.1)$$

$$\ddot{\theta} = \frac{1}{I_v} (N_{tra} + \sum_i^{n_{obs}} L_i \times \mathbf{F}_{obs,i}) \quad (2.2)$$

$$\dot{\mathbf{x}}_{next} = \begin{cases} \dot{\mathbf{x}}_{pre} + T\ddot{\mathbf{x}}, & \text{for } |\dot{\mathbf{x}}_{pre} + T\ddot{\mathbf{x}}| \leq \dot{x}_{limit} \\ \frac{\dot{\mathbf{x}}_{pre} + T\ddot{\mathbf{x}}}{|\dot{\mathbf{x}}_{pre} + T\ddot{\mathbf{x}}|} \cdot \dot{x}_{limit}, & \text{for } |\dot{\mathbf{x}}_{pre} + T\ddot{\mathbf{x}}| > \dot{x}_{limit} \end{cases} \quad (2.3)$$

$$\dot{\theta}_{next} = \begin{cases} \dot{\theta}_{pre} + T\ddot{\theta}, & \text{for } |\dot{\theta}_{pre} + T\ddot{\theta}| \leq \dot{\theta}_{limit} \\ \frac{\dot{\theta}_{pre} + T\ddot{\theta}}{|\dot{\theta}_{pre} + T\ddot{\theta}|} \cdot \dot{\theta}_{limit}, & \text{for } |\dot{\theta}_{pre} + T\ddot{\theta}| > \dot{\theta}_{limit} \end{cases} \quad (2.4)$$

$\mathbf{F}_{tra} = (F_{tra,x}, F_{tra,y})^T$ : 参照位置からの力

$\mathbf{F}_{obs,i} = (F_{obs,i,x}, F_{obs,i,y})^T$ : 障害物  $i$  からの力

$F_{tra,x} = K_{tra,x}(x_{ref} - x_{pre}) + D_{tra,x}(\dot{x}_{ref} - \dot{x}_{pre})$

$F_{tra,y} = K_{tra,y}(y_{ref} - y_{pre}) + D_{tra,y}(\dot{y}_{ref} - \dot{y}_{pre})$

$N_{tra} = K_{ori}(\theta_{ref} - \theta_{pre}) + D_{ori}(\dot{\theta}_{ref} - \dot{\theta}_{pre})$ : 参照姿勢からのモーメント

$$F_{obs,i,x} = \begin{cases} \sum_j^{n_{point}} \left( K_{obs} \left( \frac{1}{d_{ij} - L_{min}} - \frac{1}{L_{obs} - L_{min}} \right) \frac{x_{obs,i} - x_{pre}}{d_{ij}} \right. \\ \quad \left. + D_{obs} \frac{L_{obs} - L_{min}}{d_{ij} - L_{min}} (\dot{x}_{obs,i} - \dot{x}_{pre}) \right), & \text{for } d_{ij} \leq L_{obs} \\ 0, & \text{for } d_{ij} > L_{obs} \end{cases}$$

$$F_{obs,i,y} = \begin{cases} \sum_j^{n_{point}} \left( K_{obs} \left( \frac{1}{d_{ij} - L_{min}} - \frac{1}{L_{obs} - L_{min}} \right) \frac{y_{obs,i} - y_{pre}}{d_{ij}} \right. \\ \quad \left. + D_{obs} \frac{L_{obs} - L_{min}}{d_{ij} - L_{min}} (\dot{y}_{obs,i} - \dot{y}_{pre}) \right), & \text{for } d_{ij} \leq L_{obs} \\ 0, & \text{for } d_{ij} > L_{obs} \end{cases}$$

$L_i$ : 対象物の重心から  $\mathbf{F}_{obs,i}$  の仮想外力作用点へのベクトル

$d_{ij}$ : 障害物  $i$  と対象物の制御点  $j$  との最短距離

$n_{obs}$ : 移動ロボット群のセンシングエリア内にある障害物の個数

$n_{point}$ : 対象物上の制御点の数

$\mathbf{x}_{pre} = (x_{pre}, y_{pre})^T$ : 対象物の現在位置

$\theta_{pre}$ : 対象物の現在姿勢

$(x_{obs,i}, y_{obs,i})^T$ : 障害物  $i$  の位置

$(x_{ref}, y_{ref})^T$ : その周期における対象物の参照位置 (あらかじめ決定される)

$\theta_{ref}$ : その周期における対象物の参照姿勢 (あらかじめ決定される)

$\dot{\mathbf{x}}_{ref}$ : その周期における対象物の参照速度 (あらかじめ決定される)

$\dot{\theta}_{ref}$ : その周期における対象物の参照角速度 (あらかじめ決定される)

$\dot{\mathbf{x}}_{next}$ : 次の周期における  $\dot{\mathbf{x}}$  (対象物に対する並進速度入力)

$\dot{\theta}_{next}$ : 次の周期における  $\dot{\theta}$  (対象物に対する回転速度入力)

ここでは大域座標系が採用されている.

また以下の定数は設計パラメータである.

(1)  $M_v, I_v$ : 対象物の仮想質量, 仮想慣性モーメント

(2)  $K_{tra,x}, K_{tra,y}, D_{tra,x}, D_{tra,y}$ : 参照位置からのバネ・ダンパ係数

(3)  $K_{ori}, D_{ori}$ : 参照姿勢からのバネ・ダンパ係数

(4)  $K_{obs}, D_{obs}$ : 障害物からのバネ・ダンパ係数

(5)  $L_{obs}$ : 障害物からのバネの自然長

(6)  $L_{min}$ : 障害物からのバネの最小長

(7)  $L_{block}$ : ある時刻における参照点と実際の位置とのずれの閾値

(8)  $\dot{x}_{limit}, \dot{\theta}_{limit}$ : 対象物の速度・角速度の最大値

(1) から (7) のパラメータは対象物軌道の安定性に関する設計パラメータであり, その設計方法を 2.4.2 項で示す.

また(8)の2つのパラメータは、ロボットの動作計画と対象物の動作計画を分離したことにより必要となった設計パラメータであり、ロボットの速度限界から求められるが、その設計方法については2.4.3項で示す。

### 2.4.2 安定性を考慮したパラメータ設計

本章では文献[太田93]にしたがって、2.4.1項で示した設計パラメータのうち計画軌道の安定性に関するパラメータの設計を行う。ただし動的な環境下での適切なパラメータ設計は困難なため、対象物が障害物からの斥力および目標からの引力が釣りあった静的状態における安定な過渡応答を設計する。

はじめに設計に当たって

- 仮想質量および仮想慣性モーメントは、対象物が複数存在する場合に優先度等を示すものであり、ここでは設計対象としない。
- 障害物からのバネ最小長  $L_{min}$  は0とする。
- 簡単のため、目標位置からの仮想バネ定数、仮想ダンパ定数を  $x, y$  方向に関して同一とする。

という前提を設けるため、設計対象となるパラメータは

$K_{tra}(=K_{tra,x}=K_{tra,y}), D_{tra}(=D_{tra,x}=D_{tra,y}), K_{ori}, D_{ori}, K_{obs}, D_{obs}, L_{obs}, L_{block}$  の8種類となる。

ここで、定常特性を決定する  $L_{block}$  はそのままでは扱いにくいので、最も近接した際の対象物・障害物間のつり合い距離  $dist$  を用いる。また、環境が変化する状況でバネ・ダンパ係数を直接指定するのは困難なため、つり合い状態における離散時間運動方程式の特性方程式の極を直接設計するアプローチをとる。すなわち、環境（障害物に囲まれる割合）が最も疎な場合と密な場合の特性方程式を求め、それらの極が安定性の条件を満たせば良い。ここで「環境が疎」とは、Fig.2.6(a)のように周辺に障害物が何もなく参照点からのみ力を受ける場合を意味し、「環境が密」とは、Fig.2.6(b)のように周囲を障害物に取り囲まれ、最も多くの力を受ける場合を意味する。これは、対象物の周囲を障害物にどの程度囲まれるかによって対象物に作用する仮想外力および仮想モーメントが変化するため、最大の仮想外力および仮想モーメントがかかる場合と、最小の仮想外力および仮想モーメントがかかる場合とを考慮して、そのどちらにおいても安定ならば、これらの中間とみなせる環境においても安定であるという考え方に基づいている。

まず特性方程式は、並進および回転に関する系の離散的な運動方程式から1動作計画周期を  $T$  として

$$M_v s^2 + (D_{trans} T - 2M_v) s + M_v + K_{trans} T^2 - D_{trans} T = 0 \quad (2.5)$$

$$I_v s^2 + (D_{rot} T - 2I_v) s + I_v + K_{rot} T^2 - D_{rot} T = 0 \quad (2.6)$$



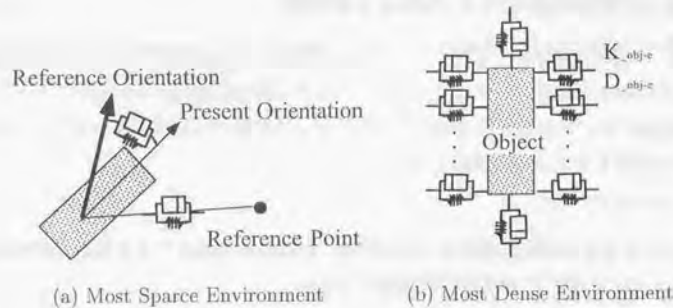


Fig. 2.6: Variable Environment

と表される、ただし

$K_{trans}$ :  $K_{trans.s}$  または  $K_{trans.d}$

$K_{rot}$ :  $K_{rot.s}$  または  $K_{rot.d}$

$D_{trans}$ :  $D_{trans.s}$  または  $D_{trans.d}$

$D_{rot}$ :  $D_{rot.s}$  または  $D_{rot.d}$

であり、添字の  $s$  および  $d$  は、それぞれ、対象物を取り巻く環境が疎な場合と密な場合を表す。また、添字の  $trans$  および  $rot$  は、それぞれ、並進に関するもの、回転に関するものであることを表す。

これらの特性方程式における極を

(1) 最も疎な状態における並進方程式の極  $p_{trans.s.1}$ ,  $p_{trans.s.2}$

(2) 最も疎な状態における回転方程式の極  $p_{rot.s.1}$ ,  $p_{rot.s.2}$

(3) 最も密な状態における並進方程式の極  $p_{trans.d.1}$ ,  $p_{trans.d.2}$

(2) 最も密な状態における回転方程式の極  $p_{rot.d.1}$ ,  $p_{rot.d.2}$

と置くと、直接値を設計するパラメータおよびその制約条件は以下のようにまとめることができる。

• 指定するパラメータ

$$p_{m.e.i} \ (m: trans, rot \ e: s, d \ i: 1, 2), \ dist, L_{obs}$$

• 上記パラメータへの制約条件

$$- p_{trans.s.1} > p_{trans.s.2} > p_{trans.d.1} > p_{trans.d.2}$$

$$p_{rot.s.1} > p_{rot.s.2} > p_{rot.d.1} > p_{rot.d.2}$$

$$- |p_{m.e.i}| < 1 \ (m: trans, rot \ e: s, d \ i: 1, 2)$$

$$- L_{obs} < (\text{ロボットのセンシングエリア})$$

$$- L_{min} < dist < L_{obs}$$

• 最終的に導出するパラメータ

$$K_{tra}, D_{tra}, K_{ori}, D_{ori}, K_{obs}, D_{obs}, L_{block}, L_{obs}$$

ここで、 $dist$  および  $L_{obs}$  はそのまま値を与えるため8個の極から設計するパラメータは6個となるが、満たすべき条件式は8個で過剰となる。そこで6個の極を上記の条件により決定し導出されたパラメータ群が残りの2個の極（ここでは最も密な場合の回転に関する極  $p_{rot.d.1}$ ,  $p_{rot.d.2}$  とする）を与える特性方程式の安定条件を満たすための条件を求めることにする。

したがって、パラメータ設計の手順をもう一度まとめると

(1) 前述の制約条件下で  $p_{trans.s.i}$ ,  $p_{trans.d.i}$ ,  $p_{rot.s.i}$  ( $i=1, 2$ ),  $dist$ ,  $L_{obs}$  を決定する

(2) Eq.(2.8), Eq.(2.12) から  $K_{tra}, D_{tra}, K_{obj}, D_{obj}, K_{ori}, D_{ori}$  を、この結果と Eq.(2.13) から  $K_{obs}, D_{obs}, L_{block}$  を導出する。

(3) 上記で求めたパラメータが残りの極の安定条件を示す Eq.(2.15) を満たすかどうかチェックする。

以下、導出に必要な式を示す。

まず極とバネ・ダンパ係数との関係は以下で表される.

$$\begin{cases} K_{trans.d(s)} = \frac{M_v}{T^2}(-1 + p_{trans.d(s),1}p_{trans.d(s),2}) + \frac{1}{M_v T^2}(2 - p_{trans.d(s),1} - p_{trans.d(s),2}) \\ D_{trans.d(s)} = \frac{1}{M_v T}(2 - p_{trans.d(s),1} - p_{trans.d(s),2}) \end{cases} \quad (2.7)$$

$$\begin{cases} K_{rot.d(s)} = \frac{I_v}{T^2}(-1 + p_{trans.d(s),1}p_{trans.d(s),2}) + \frac{1}{I_v T^2}(2 - p_{trans.d(s),1} - p_{trans.d(s),2}) \\ D_{rot.d(s)} = \frac{1}{I_v T}(2 - p_{trans.d(s),1} - p_{trans.d(s),2}) \end{cases} \quad (2.8)$$

次に, Fig.2.6で考慮したような, 環境が最も疎な場合と最も密な場合の, 対象物の運動方程式におけるバネ・ダンパ定数をそれぞれ以下に求める.

参照点以外の環境との間に張ったバネ・ダンパ係数を  $K_{obje}, D_{obje}$  と置き, 対象物の重心から周上への最大距離を  $L_{max}$  対象物の周囲長を  $L_{cir}$  とする.

#### (1) 最も疎な場合

##### (a) 並進に関してかかる最小の仮想外力

対象物に寄与するのは目標位置・姿勢からのバネ・ダンパのみなので

$$\begin{cases} K_{trans.s} = K_{tra} \\ D_{trans.s} = D_{tra} \end{cases} \quad (2.9)$$

##### (b) 回転に関してかかる最小の仮想モーメント

目標姿勢からのバネ・ダンパのみなので

$$\begin{cases} K_{rot.s} = K_{ori} \\ D_{rot.s} = D_{ori} \end{cases} \quad (2.10)$$

#### (2) 最も密な場合

##### (a) 並進に関してかかる最大の仮想外力

$$\begin{cases} K_{trans.d} = K_{obje} L_{cir} + K_{tra} \\ D_{trans.d} = D_{obje} L_{cir} + D_{tra} \end{cases} \quad (2.11)$$

##### (b) 回転に関してかかる最大の仮想モーメント

対象物の周囲上の各点に重心からの距離に比例した力がかかるが, これらによるモーメントの上限を考慮すると

$$\begin{cases} K_{rot.d} = K_{obje} L_{max} L_{cir} + K_{ori} \\ D_{rot.d} = D_{obje} L_{max} L_{cir} + D_{ori} \end{cases} \quad (2.12)$$

また  $K_{obs}, D_{obs}, L_{block}$  は以下の式から求まる.

$$\begin{aligned} K_{obs} &= \left( \frac{dist - L_{min}}{L_{obs} - L_{min}} \right)^2 K_{obje} \\ D_{obs} &= \left( \frac{dist - L_{min}}{L_{obs} - L_{min}} \right) D_{obje} \\ L_{block} &= \frac{K_{obje}}{K_{tra}} (dist - L_{min})^2 \left( \frac{1}{dist - L_{min}} - \frac{1}{L_{obs} - L_{min}} \right) \end{aligned} \quad (2.13)$$

最後に, 残り2個の極, すなわち最も環境が密な場合の回転に関する特性方程式の安定条件は, 離散時間系の特性方程式の安定条件を求めるジュリーの判別式[美多86]を用いて次のように表せる.

$$\begin{aligned} K_{rot.d} T^2 &> 0 \\ 4I_v - 2D_{rot.d} T + K_{rot.d} T^2 &> 0 \\ D_{rot.d}^2 - 4I_v K_{rot.d} T^2 &> 0 \end{aligned} \quad (2.14)$$

これらは, 他のパラメータから結局, 以下の式で表せる.

$$\begin{aligned} (K_{trans.d} - K_{trans.s}) L_{max} + K_{rot.s} &> 0 \\ 4I_v - 2T((D_{trans.d} - D_{trans.s}) L_{max} + D_{rot.s}) + T^2((K_{trans.d} - K_{trans.s}) L_{max} + K_{rot.s}) &> 0 \\ ((D_{trans.d} - D_{trans.s}) L_{max} + D_{rot.s})^2 - 4I_v((K_{trans.d} - K_{trans.s}) L_{max} + K_{rot.s}) &> 0 \end{aligned} \quad (2.15)$$



## 2.4.3 対象物の速度・角速度限界の求め方

ここでは、ロボットの動作計画と対象物の動作計画を分離したことにより必要となった設計パラメータである、対象物の速度・角速度限界  $\dot{x}_{limit}$  および  $\dot{\theta}_{limit}$  の設計方法を示す。

ロボットの動作計画において少なくとも1つ解の存在するような最低限の条件は以下のように捉えることができる。すなわち、全てのロボットが搬送対象物の周縁上において、回転中心となる重心から最も遠い位置を把持し、計画上の最大速度最大角速度で動作した場合にも、これをロボットの速度限界内で実現可能であれば (Fig.2.7)、これよりも重心に近いどの場所を各ロボットが固く把持しても同様の動作を実現する事が可能である。

したがって、その限界速度は以下を満たす範囲で決定することにする。

$$\dot{x}_{limit}T + L_{max}\dot{\theta}_{limit}T \leq \dot{r}_{limit}T \quad (2.16)$$

ただし

$\dot{r}_{limit}$  : ロボットの限界速度

$L_{max}$  : 回転中心となる重心から最も遠い位置までの距離

$T$  : 1動作計画周期の時間

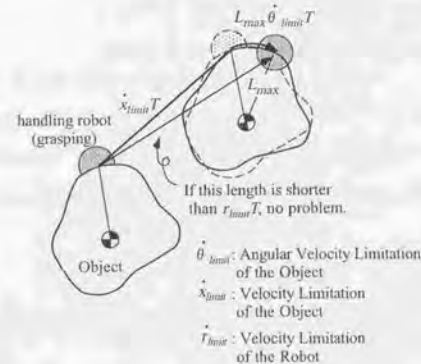


Fig. 2.7: Relationship between Velocity of Object and Robot

## 2.5 移動ロボット群の動作計画

## 2.5.1 ロボットの動作計画の概要

ここでは、計画された対象物の目標位置・姿勢を実現するロボット動作の計画手法を構築する。与えられた要求仕様を満たすロボット群の動作計画問題は、以下のように記述できる。

制約条件 1	対象物の目標位置・姿勢を実現すること
制約条件 2	ロボットの速度限界を守ること
評価指標	障害物回避, 物体の安定支持, 小エネルギーでのハンドリング
パラメータ	移動ロボット群の配置

本論文では、この問題を以下のように解く。

- 制約条件1は、最低限2台のロボットを対象物座標系において固定することで実現する。これは、対象物の荷重を支えと共に、水平面内のハンドリング力を加え、対象物に計画された運動を生成させることを意味する（これを把持ロボットと呼ぶ）(Fig.2.8(a))。
- 評価指標の最適化は制約条件2を満たす範囲で残りのロボットの配置を変更することで実現する。これは、物体の荷重を支えながら計画された最適把持位置まで持ち替えることを意味する（これを持ち替えロボットと呼ぶ）。対象物の動作が決定していることから、1動作計画周期後の対象物コンフィギュレーションにおけるロボット配置を求めることになる (Fig.2.8(b))。
- 上記の把持位置に基づき、各ロボットへの速度指令を導出する (Fig.2.8(c))。

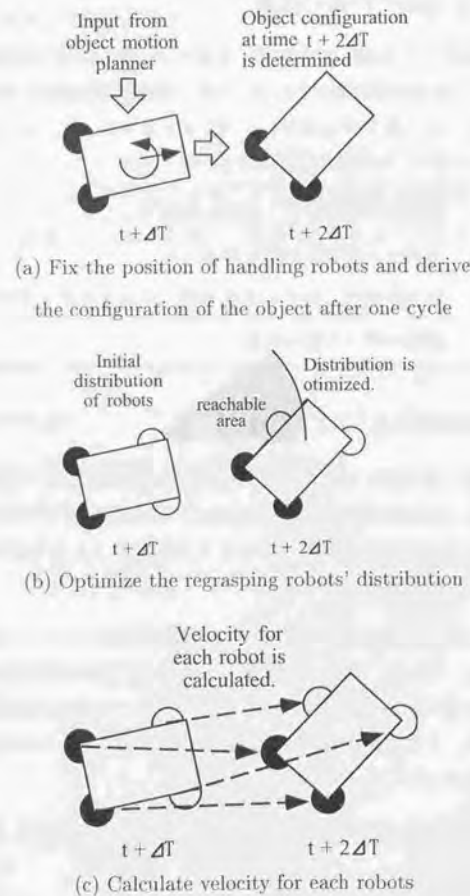


Fig. 2.8: Process of Robot Motion Planning

## 2.5.2 持ち替え計画アルゴリズム

ロボットの配置（持ち替え）計画は以下に示す手順で行う。

**Step 1**  $n_{rob}$  台のロボットから把持ロボットを  $m$  台 ( $m \geq 2$ ) 選び  $\sum_{m=2}^{n_{rob}} C_m$  通りの全ての組合せに対し次のプロセス **Process 1** ~ **Process 3** を試行する。

**Process 1** 把持ロボットの把持位置を固定したと仮定する。

**Process 2** ロボットの速度の制約条件下で評価関数  $PI$  を最小化する持ち替えロボットの最適な把持位置を求める。

**Process 3** 終了。

**Step 2** **Step 1** で求まった解のうち評価関数値が最も小さい場合を選ぶ。

2.5.2.1 評価関数  $PI$ 

ロボットの動作計画に対する要求仕様は

- (1) 障害物を回避すること
- (2) 対象物を安定に把持すること
- (3) 小エネルギーでのハンドリング

である。ここでは、それぞれを

- (1) 障害物からなるべく遠ざかること
- (2) 対象物の重心が、ロボットのなす多角形のなるべく中心にあること
- (3) 対象物の重心まわりのモーメントを各ロボットがより小さいハンドリング力で発生するため、重心からなるべく遠ざかること

であると捉える。

したがって、評価関数として次式を用いる (Fig.2.9)。

$$PI = \frac{\alpha}{d_1^2} + \frac{\beta}{d_2^2} + \frac{\gamma}{d_3^2} \quad (2.17)$$



ただし

$$d_1 = \min_{1 \leq j \leq n_{\text{rob}}} \min_{1 \leq k \leq n_{\text{obs}}} D_1(j, k)$$

$$d_2 = \min_{1 \leq j \leq n_{\text{rob}}} D_2(j)$$

$$d_3 = \min_{1 \leq j \leq n_{\text{rob}}} D_3(j)$$

$D_1(j, k)$ : ロボット  $j$  と障害物  $k$  の間の最短距離

$D_2(j)$ : ロボット  $j$  とロボット  $j+1$  の成す直線と対象物の重心との最短距離

$D_3(j)$ : ロボット  $j$  と対象物の重心の間の最短距離

$\alpha, \beta, \gamma$ : 重み係数

式(2.17)の第1項, 第2項, 第3項はそれぞれ要求仕様(1)~(3)に対応する.

ここで **Process 2** は不等式拘束条件付きの最適化問題であり, 非線形計画法のひとつであるバナルティ乗数法〔今野78〕を用いて解く.

### 2.5.2.2 重み係数の設計方法

重み係数  $(\alpha, \beta, \gamma)$  については, 各指標  $(d_1, d_2, d_3)$  に対しておおよそその変域を求め, その中で「これ以上値が下がった場合, 状況が悪化したとみなされる限界値 (範囲)」を決定し, この時,

$$\frac{\alpha}{d_1^2} \approx \frac{\beta}{d_2^2} \approx \frac{\gamma}{d_3^2} \quad (2.18)$$

となるように定める.

これによって, ある項目に対応する変数の値が限界値以上に悪化した場合に, その項目を優先的に改善することが可能になる.

ただし, 短期間で区切り繰り返し解を求めるために, ある瞬間における最適化の結果が, その後得られる環境情報に対して時系列としても有効である保証はない. したがって, 上記の指針のもとでおおよその値を求めたのち, 試行錯誤的に値を変更し決定する事になる.

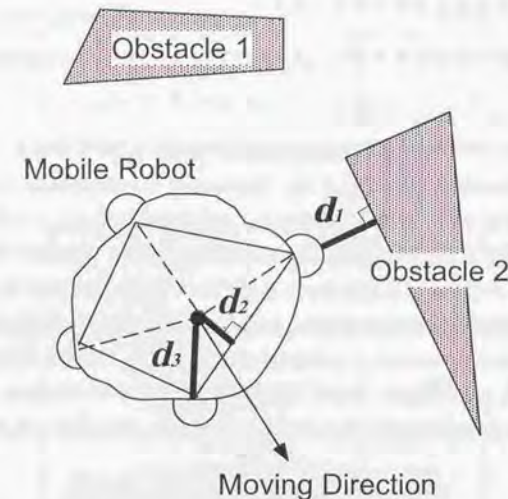


Fig. 2.9: How to derive  $d_1$ ,  $d_2$  and  $d_3$

## 2.6 持ち替え協調ハンドリングシミュレーション

### 2.6.1 シミュレーションの設定

本章では、2.5節で提案したロボットの動作計画アルゴリズムの有効性を検証するため、全方向移動ロボットによる持ち替えハンドリングシミュレーションを行う。

ここでは“持ち替え動作の効果”の評価のため、

**Simulation 1:** 持ち替えを伴うハンドリング

**Simulation 2:** 持ち替えを伴わないハンドリング

の2種類のシミュレーションを行う。

各シミュレーションにおける評価関数の係数をTable 2.4にまとめる。その他のシミュレーション条件および環境は、Fig.2.10, Table 2.2, Table 2.3に示すものを共通して用いる。Fig.2.10はロボット群の初期状態および対象物の目標コンフィギュレーションを表す。図中の黒い多角形は障害物を、白い長方形は対象物を、複数の小さい円はロボットを表す。ロボットの初期配置は長方形の各頂点であり、周辺に障害物が存在しない場合に、評価関数の第2項及び第3項を最適化する配置となっている。搬送開始時には障害物は未知であり、対象物のグローバルな目標軌道にはスタートとゴールを等速直線運動で結んだ単純なものを与える。Table 2.2は、対象物およびロボットの形状等に関する条件を、Table 2.3は対象物の動作計画に用いた拡張仮想インピーダンス法の条件を示している。

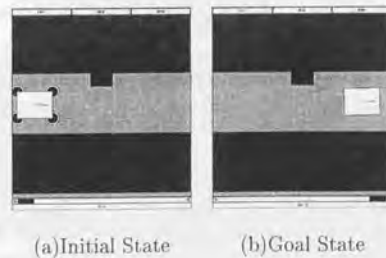


Fig. 2.10: Start and Goal Positions for the Simulations

Table 2.2: Simulation Conditions (general)

the shape of the object	0.8×0.6 [m] rectangle
center of gravity of the object	center of the object
the real weight of the object	10.0 [kg] rectangle
the real inertia of the object	0.83 [kg.m <sup>2</sup> ]
the velocity limit of the object ( $\dot{x}_{limit}$ )	0.05 [m/s]
the angular velocity limit ( $\dot{\theta}_{limit}$ )	0.08 [rad/s]
the maximum velocity of the robots ( $\dot{r}_{limit}$ )	0.1 [m/s]
the number of the robots	4
the shape of the robots	circle (radius=0.15 [m])
sensing area of the robots	circle (radius=5.0 [m])
motion planning cycle	2.0 [s]

Table 2.3: Simulation Conditions (parameters for Virtual Impedance Method)

virtual weight of the object ( $M_v$ )	1.0 [kg]
virtual Inertia of the object ( $I_v$ )	1.0 [kg.m <sup>2</sup> ]
$p_{trans.s.1}$ (translation, sparse)	0.9
$p_{trans.s.2}$ (translation, sparse)	0.45
$p_{trans.d.1}$ (translation, dense)	0.4
$p_{trans.d.2}$ (translation, dense)	0.1
$p_{rot.s.1}$ (rotation, sparse)	0.95
$p_{rot.s.2}$ (rotation, sparse)	0.1
$dist$	0.22
$L_{obs}$	0.4
$L_{min}$	0.0



Table 2.4: Coefficients of Performance Index in Each Simulation

	Weight Coefficient		
	$\alpha$ (regrasping)	$\beta$	$\gamma$
Simulation 1	0.2 (○)	5.0	3.0
Simulation 2	0.0 (×)	5.0	3.0

### 2.6.2 シミュレーション結果

Fig.2.11に持ち替えを行う場合 (Simulation 1) の、Fig.2.13に持ち替えを行わない場合 (Simulation 2) のシミュレーションの様子を示す、またFig.2.12は、Simulation 1の全ての動作計画周期における持ち替え計画結果を表している。図中の白い円は持ち替えロボットを、黒い円は把持ロボットを表す。

対象物の動作は2つの戦略で同一であり、拡張仮想インピーダンス法による対象物の障害物回避が良好になされ、障害物に衝突する事なく180秒において目標コンフィギュレーションを達成している。

### 2.6.3 持ち替え効果の評価

Simulation 1 および Simulation 2 の比較により、協調ハンドリング作業における持ち替え動作導入効果の評価を行う。

0秒のハンドリング作業を開始時には、両シミュレーションともロボットは同じ把持位置をとっている (Fig.2.11(a), Fig.2.13(a))。18秒において、Simulation 1では、進行方向左前の通路の突起部への衝突回避のためrobot1が後方に持ち替えを行っている (Fig.2.11(b) から (c))。しかしSimulation 2では、同時刻に持ち替えは行われなかったため (Fig.2.13(b))、その結果、24秒においてrobot1が障害物に極度に近接している (Fig.2.13(c))。

その後もSimulation 1では、180秒でハンドリング作業が完了するまでの間に持ち替え動作が頻繁に見られる。Fig.2.11の(d)から(e)、および(f)から(g)では、robot1が障害物を回避する方向に把持位置を変更し、この時robot4との間で安定条件が崩れないよう同時にrobot4が安定条件を改善する方向に動いている。この持ち替え動作を受け、Fig.2.11の(h)から(i)では、robot2およびrobot3が、安定条件を改善する方向に把持

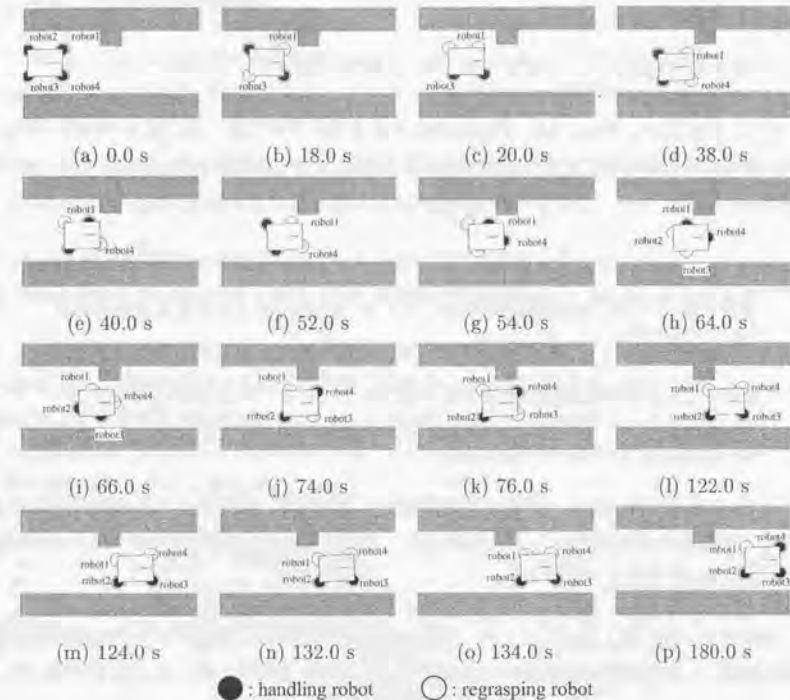


Fig. 2.11: Results of Simulation 1

位置を変更している。Fig.2.11の(j)から(k)、(l)から(m)、および(n)から(o)では、再びrobot1が突起部との距離を保つために把持位置の変更を行っている。最後にFig.2.11の(p)では突起部からの距離が大きくなる。したがって、残りの安定把持および小エネルギーでのハンドリングの評価項が最も改善される把持位置、すなわち長方形の各頂点を支持するという状態をとり、作業を終了している。

これらの結果より、持ち替え動作を導入した提案手法によるロボット群の動作計画について以下のことがわかった。

- 周囲に障害物が存在しない領域においては安定把握およびハンドリング力の減少が

重視されている。

- 障害物が近接している領域では、障害物回避が重視されている。

また、Fig.2.14, Fig.2.15, Fig.2.16にそれぞれ示された $d_1$ ,  $d_2$ ,  $d_3$ の時間変化およびこれらの平均値をまとめたTable 2.5より定量的にこれを評価すると、以下のことがわかる。

- $d_1$ について、ロボットと障害物の最近接距離が、持ち替えありの場合では $8.44 \times 10^{-2}$  [m]であるのに対し、持ち替えなしの場合では、 $3.96 \times 10^{-4}$  [m]にまで接近していることがわかる。
- $d_2$ に関しては、最も不安定に近い状況で、持ち替えありの場合が $2.02 \times 10^{-1}$  [m]となっていることがわかる。一方、持ち替えなしの場合は把持位置が固定されているため、 $3.0 \times 10^{-1}$  [m]と一定である。
- $d_3$ に関しては、最もハンドリング力を要する状況で、持ち替えありの場合が $3.08 \times 10^{-1}$  [m]であるのに対し、持ち替えなしの場合は把持位置が固定されているため、 $5.0 \times 10^{-1}$  [m]と一定である。

すなわち、持ち替えの導入により、把握の安定度を約37%、ハンドリング力の抑制を約40%犠牲にして障害物回避性能の大幅な改善（2.13倍）がなされていることが分かった。

Table 2.5: Average Distance of  $d_1$ ,  $d_2$  and  $d_3$  [m]

Distance	Simulation 1 (with regrasp)	Simulation 2 (without regrasp)
$d_1$	0.240	0.176
$d_2$	0.285	0.300
$d_3$	0.422	0.500

## 2.6.4 計算時間の評価

SPARCstation20を用いて計算を行ったところ、各ステップ平均で0.3秒かかった。これは動作計画周期の2.0秒に比較して十分短く、提案手法の実時間処理性が示されたと言える。

ただし、提案した持ち替え計画アルゴリズムでは総当たり探索を行っているため、ロボット台数が増加した場合には計算量低減のための工夫が必要となる。

以上をまとめると、提案手法により、

対象物周辺の局所的な環境情報に基づき、各動作計画周期において、障害物を回避する対象物の運動と、その運動を実現する周辺環境に応じたロボットの動作が実時間で導出可能であり、これを繰り返す事で対象物の目標コンフィギュレーションを達成する動作が計画可能である

ことが分かり、その有効性が示された。



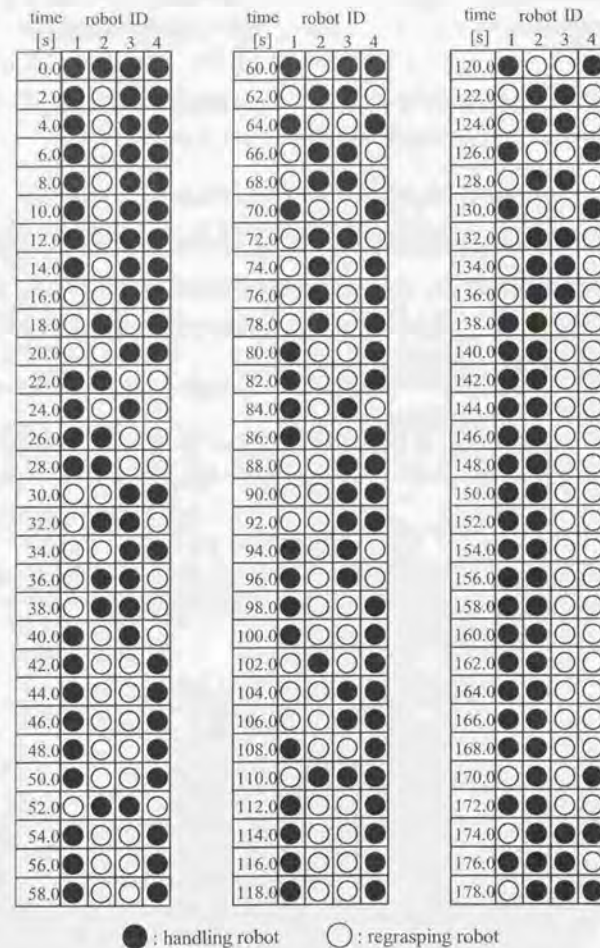


Fig. 2.12: Transition of grasp state (Simulation 1)

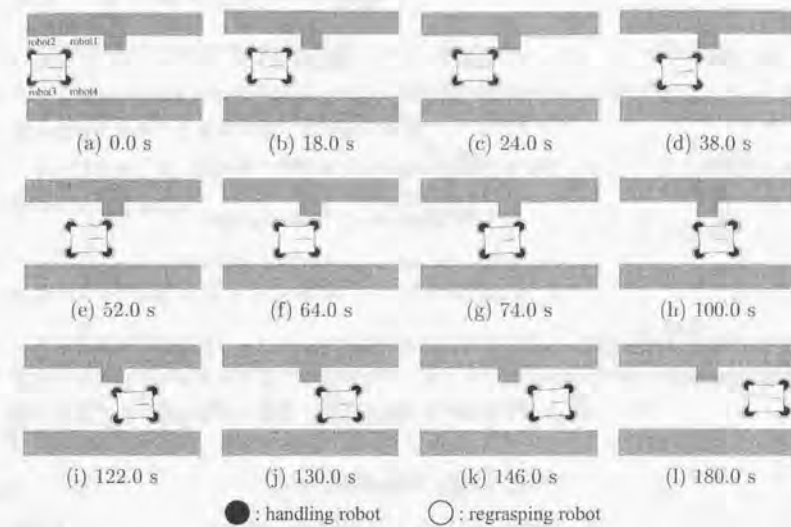


Fig. 2.13: Results of Simulation 2

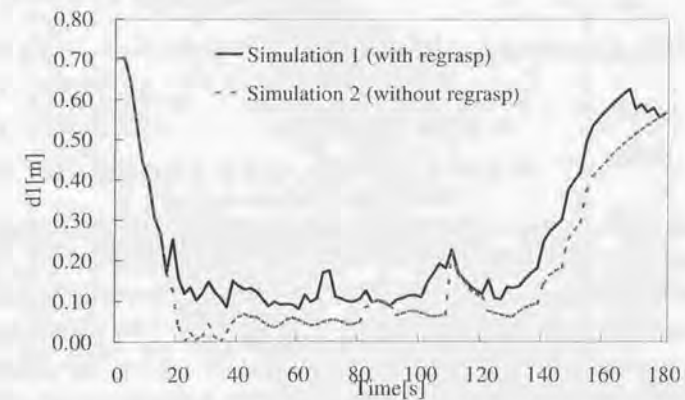
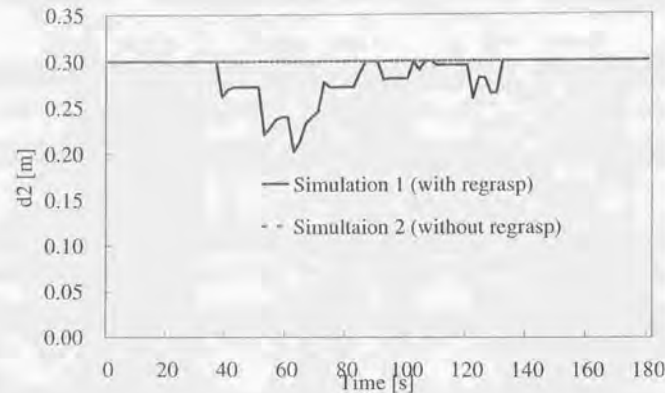
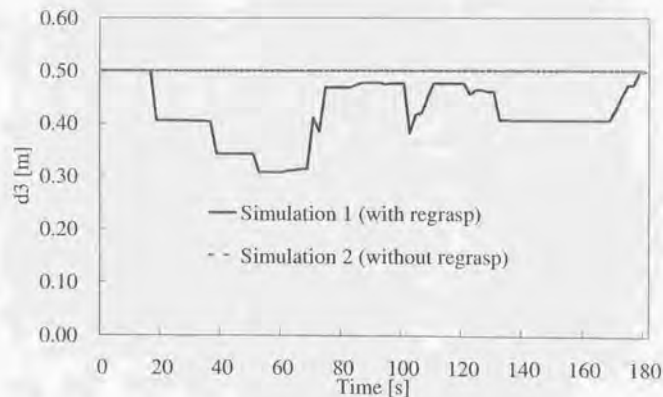


Fig. 2.14: Transition of  $d_1$

Fig. 2.15: Transition of  $d_2$ Fig. 2.16: Transition of  $d_3$ 

## 2.7 持ち替えコストの考慮

ここまでは、ロボットによる持ち替えコストを考慮せずにアルゴリズムを構築し検証を行って来た。しかしロボットによっては、持ち替え動作の実現に非常な困難を伴い、持ち替えの回数を減らす必要の生じる場合がある。

そこで本節では、前節までで提案した手法の拡張により持ち替えコストの考慮を可能とする手法を提案し、シミュレーションによる検証を行う。

### 2.7.1 持ち替えコストを考慮した評価関数

持ち替えの回数を減らすため、持ち替えを実行するロボット台数が少ない状態ほど評価の高くなる評価関数を用いることを考える。そこで、前節までで用いていた評価関数に、持ち替えコストの評価項を加えた以下の式を用いる事にする。

$$PI = \frac{\alpha}{d_1^2} + \frac{\beta}{d_2^2} + \frac{\gamma}{d_3^2} + \delta \cdot n_{regrab} \quad (2.19)$$

ただし

$n_{regrab}$ : 持ち替えを実行するロボット台数

$\delta$ : 持ち替えコスト評価項の重み係数

提案アルゴリズムでは最適化過程において評価関数の最小化を行うため、持ち替え台数が少ないほど値が小さくなり好ましい状態と評価される。

### 2.7.2 持ち替えコストを考慮したシミュレーション

上記で提案した評価関数を用いた協調ハンドリングシミュレーション (Simulation 3) を行い、有効性を検証する。シミュレーションにおける評価関数の係数を Table 2.6 に示す。比較のため、2.6 節で行った 2 種類のシミュレーション Simulation 1 および Simulation 2 の条件を併せて示す。これらのシミュレーションは持ち替えコストが非常に低い場合と考えられるため、持ち替えコストの項の重みが 0 となっている。Simulation 3 では、持ち替えコストの項以外の係数およびその他環境等の条件は 2.6 節における Simulation 1 と同一となっている。



シミュレーション結果をFig.2.17に示す。90ステップの動作計画のうち、持ち替え動作は、ロボット1が、20秒の動作計画周期（Fig.2.17(b) から(c)）および90秒の動作計画周期（Fig.2.17(d)から(e)）において1度ずつ実行するのみという結果が得られた。

この時の各評価項目 $d_1$ ,  $d_2$ ,  $d_3$ の値の遷移をFig.2.18, Fig.2.19, Fig.2.20に、それらの平均値をTable 2.7に示す。

Fig.2.18から、持ち替えコストの高いSimulation 3は、持ち替えを行わないSimulation 2に比べ、障害物に最も近接している18秒から48秒において、障害物からの距離を大きく保っており、安全な状態であると言える。また、持ち替えコストの低いSimulation 1と比較すると、20秒から36秒にかけては持ち替えコストが高いと設定したSimulation 3の方が障害物からの距離が約0.05[m]大きい値となっているが、その他の時刻では、Simulation 2とほぼ同様の値をとっている。

$d_2$ および $d_3$ の項については、持ち替え動作が2回のみ実行されたことから変化量が少なく、 $d_2$ の項についてはFig.2.19より、持ち替えを行わないSimulation 2とほぼ同様の結果となっている。ただし $d_3$ の項については、2度目の持ち替え動作の後、最良値までは回復せず、作業が終了している。これは2度目の持ち替え動作において、開始時点で把持していた頂点まで戻らず、その後は障害物からの距離が全体的に増大したことから持ち替え動作が起こらなかったためである。

これらの結果から、持ち替えコストが高い場合、障害物が非常に近接する環境以外では、主に障害物回避の度合を犠牲にすることで持ち替え回数を減らす計画となっていることが分かった。

以上より、評価関数に持ち替えコストの評価項を追加することで、持ち替え回数を減少させ、障害物に接近した場合のみ持ち替え動作を起こすという動作計画が可能であることが分かった。すなわち、提案手法のように短時間に区切って計画を繰り返す枠組によって、持ち替えコストの高い場合にも対応可能であることが分かった。

Table 2.6: Coefficients in each Simulation

	Weight Coefficient			
	$\alpha$	$\beta$	$\gamma$	$\delta$
Simulation 1 (Low regrasping cost)	0.2	5.0	3.0	0.0
Simulation 2 (without regrasp)	0.0	5.0	3.0	0.0
Simulation 3 (High regrasping cost)	0.2	5.0	3.0	100.0

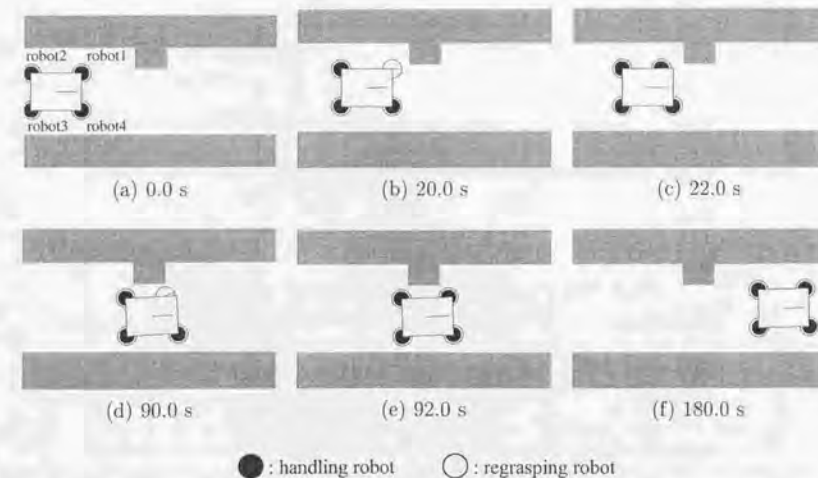


Fig. 2.17: Simulation 3 (with consideration of regrasping cost)

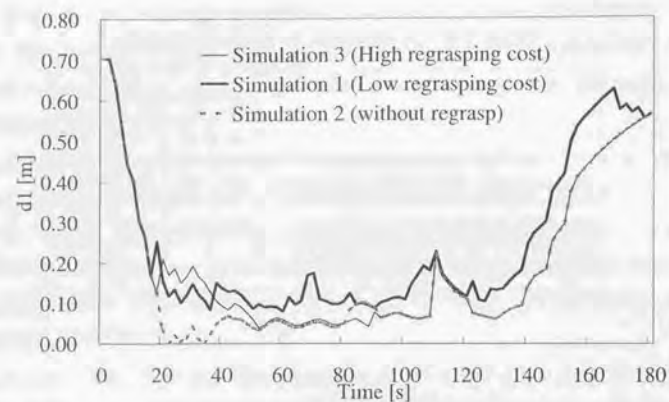


Fig. 2.18: Transition of  $d_1$

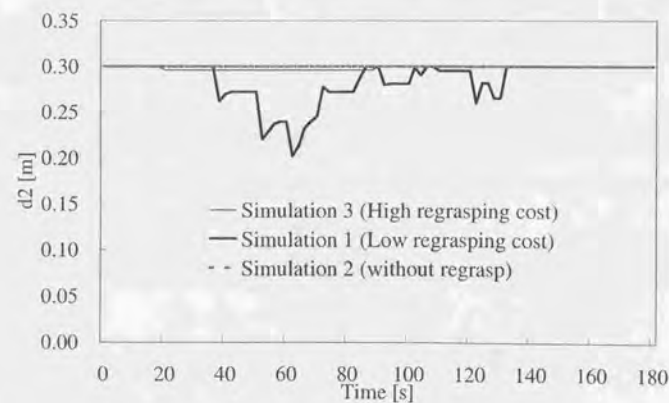


Fig. 2.19: Transition of  $d_2$

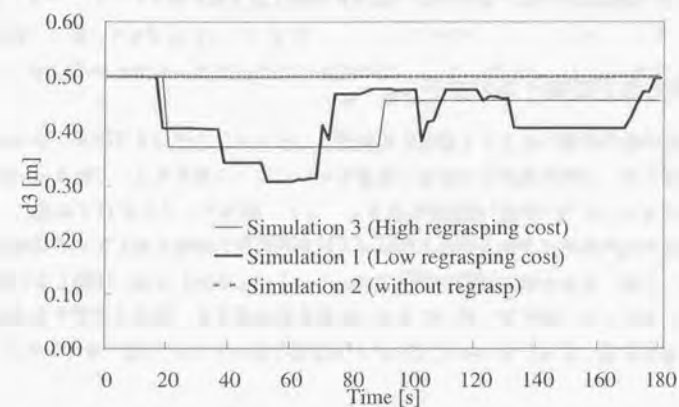


Fig. 2.20: Transition of  $d_3$

Table 2.7: Average Distance of  $d_1$ ,  $d_2$  and  $d_3$  [m]

Distance	Simulation 1 (Low regrasping cost)	Simulation 2 (without regrasp)	Simulation 3 (High regrasping cost)
$d_1$	0.240	0.176	0.198
$d_2$	0.285	0.300	0.298
$d_3$	0.422	0.500	0.428



## 2.8 提案手法の適用可能な範囲

提案した動作計画手法を適用可能な範囲について言及しておく。

### 対象物の動作計画手法の限界

対象物の動作計画手法として採用した仮想インピーダンス法は人工ポテンシャル法の一つであるため、目標経路からの引力と障害物からの斥力の和がちょうど釣り合い対象物が動けなくなるという“停留”状態が存在する。また、剛体モデルに拡張する際に、障害物からの仮想力の作用点が切り替わる事による対象物軌道の振動を減じるため制御点の概念を導入したが、完全な振動抑制は困難であった。これらに対しては、移動しない障害物を発見した場合には、障害物に沿った方向に軌道を再生成する、軌道を変更する方向は仮想外力の感度を低くする、といった工夫により改善を図ることが可能と考えられる。

### 把持可能な領域の連続性

提案手法では、対象物周縁上のすべての点において把持可能（周縁は連続）としてアルゴリズムを構築したが、対象物の形状によっては、周縁上の全ての点で把持可能なわけではなく、把持可能な領域同士が不連続な場合もある。このとき滑りにより対象物の荷重を支持しながらの持ち替えを行うことは不可能である。しかし、対象物を固く把持するロボットを3台以上とし、なおかつモーメントのつり合い式を満たす組合せに対してのみ最適化計算を行い比較対象とすることで、持ち替えロボットは持ち替えを行う間対象物を支持する必要がなくなり、持ち替え動作が可能となる。

### 対象物の形状と持ち替え計画アルゴリズム

現在の持ち替え計画アルゴリズムでは、対象物がロボットに対して比較的大型であることを想定しているため、計算上で他のロボットを飛び越える持ち替え動作の抑制については考慮していない。したがってロボット同士が適度に分散して配置された状態から開始した場合には有効な動作が得られるが、ロボットが非常に近接している場合には他のロボットの飛び越えが起り得る。この点については、評価関数にロボット間の距離を考慮することで対処可能と考えられる。

また、現在のアルゴリズムにおける物体の安定支持に関する評価方法では、物体支持に

関わる全てのロボットの成す多角形の内部に必ず重心が含まれている必要があるため、対象物として凸形状のみが扱える。しかし、上記の把持可能な領域の連続性の項と同様に、物体を支持するロボットを固定と考えた場合、残りのロボットについてはこの項目を必ずしも同項目を満たす必要はない。したがって、凸包を成すロボットを選び、これらに対してのみ、安定把持の項目の最適化計算を適用することで、凹形状にも対応可能と考えられる。

## 2.9 おわりに

本章では、幾何層に関する課題解決として、移動ロボット群による持ち替えを伴う大型対象物のハンドリング動作計画を、対象物周辺の局所的な環境情報に基づき実時間で行う手法を構築した。

計画対象である対象物およびロボットの動作に対する要求は多数ある。それらの要求の間にはトレードオフが存在し、また対象の異なる要求が混在しているため、要求同士の関係が複雑である。そこで本論文では、目標コンフィギュレーション達成までの時間を一定の短時間で区切り、周期的に繰り返し計画を行うことを考えた。さらに各周期における計画問題を対象物の動作計画と移動ロボット群の動作計画の2つに分割した。

対象物の動作計画手法としては、局所的な環境情報に基づき障害物を回避する動作を計算可能な局所動作計画器として、質点系に対する動作計画手法である仮想インピーダンス法を剛体モデルに拡張して用いた。

また、移動ロボット群の動作計画手法としては、2台以上の把持ロボットの全ての組合せに対し、それぞれ、制約条件付きの最適化問題を解く事で持ち替えロボットの把持位置を求め、これらの中から最も評価関数値の良い結果を選択する動作計画アルゴリズムを提案した。評価関数としては障害物回避・安定把握・小エネルギーでのハンドリングを、制約条件としてはロボットの速度限界を考慮し、最適化手法として非線形計画法の一つであるペナルティ乗数法を適用した。

協調ハンドリングシミュレーションにより、各動作計画周期において、近接した障害物を回避する対象物動作およびトレードオフの存在する3つの要求を満たす持ち替え動作が、実時間で計算可能であることが分かった。また、これらの繰り返しにより、最終的に目標コンフィギュレーションを達成する計画が可能である事が示された。

以上より、局所的に環境情報が取得された場合に、環境との幾何的な関係に基づく対象物動作および把持位置の動的な変更を伴うロボット動作を計画可能な、協調ハンドリング動作計画手法が構築できた。

## Chapter 3

実ロボットによる協調ハンドリング作業  
実現

3.1 はじめに	60
3.2 機構設計による問題解決	62
3.3 移動ロボット群の動作計画アルゴリズムの拡張	65
3.4 トルク限界を考慮した持ち替え搬送シミュレーション	69
3.5 協調ハンドリング実験システムの構築	76
3.6 搬送機構の評価実験	81
3.7 持ち替え協調ハンドリング実験	83
3.8 おわりに	89



### 3.1 はじめに

本章では、物理層に対する課題解決として、“実機問題の考慮に基づく協調ハンドリング作業実現”を目指す。

2章では未知環境における持ち替え動作を伴う協調ハンドリング動作計画手法を提案したが、実際にこれを移動ロボットにより実現するためには、いかに環境の変化を観測するか、またロボット群をいかに制御し力の干渉を伴う協調作業を実現するか、という二つの側面の課題がある。

前者については4章において考慮することにし、対象物周辺の情報は取得できていることを仮定して話を進める。そこで後者の、「複数台の移動ロボットによる、力の干渉を伴う共通物体の協調ハンドリング作業の実現」と言う観点から、考慮すべき課題を整理すると、

- (1) 車輪型移動機構に特有の非ホロノミック性
- (2) 移動誤差の存在下での共通物体の搬送
- (3) ロボットのトルク限界の考慮

の3つとなる。

(1)の性質に関して、一般に安定性の観点から把持位置を車軸中心におく事が多いが、車軸中心では速度方向（力を加える方向）に制限が加わることから、対象物とロボットの動作計画の分離が難しくなるなど、計算量の増加・動作計画問題の複雑化を引き起こすといった問題がある。しかし、搬送システムの実用化を考えた場合、2章で動作計画手法を構築する際に仮定していた全方向移動ロボットだけでなく一般の車輪型移動ロボットに適用できることは、導入コスト、可搬重量、メンテナンスのしやすさといった観点から大きな利益となる。

また(2)の走行誤差により生ずる過大な内力について、従来研究の多くのように、物体を固く把持し速度サーボ系による高周期での正確な力制御により回避することは一般に難しい。

(3)について、移動ロボットは床面との摩擦で駆動されており、対象物との間に限界以上のトルクがかかると車輪が滑りを起こす。これは車輪を駆動するトルクとは別のものであり、ロボットが計画した搬送作業を車輪の滑りなく達成するためには、物体ハンドリングに要するトルクの限界として考慮する必要がある。

そこで本論文では、ロボットの動作計画のうち把持位置の計画問題と車輪制御の計画問題を分離することを考え、(1)および(2)の問題に対しては機構的な解決を探り、(3)の問題を動作計画における条件として考慮するというアプローチをとる。

(3)のトルク限界を、動作計画器のどの部分において考慮するかという点に関しては、2.3節において示したロボットの速度限界と同様の扱いとする。

すなわち、対象物の動作計画器において、全てのロボットがどこを把持するかに関係なくトルク限界を超えないような加速度指令を全ての場合において適用することは作業に対し大きな拘束となってしまう。そのため、対象物の動作指令導出の際には、全てのロボットが把持しても生成できない加速度は出さない、という最低限の条件のみを考慮し、ロボットの持ち替え動作計画において制約条件として考慮することにする。

そのため、まず3.2節において、走行誤差および非ホロノミック性に対応するための搬送機構を設計する。

また、2.5節において示したロボット動作計画アルゴリズムを、トルク限界を考慮したものに拡張し(3.3節)、その有効性を協調ハンドリングシミュレーションによって検証する。

これに基づき把持機構を試作して実ロボットシステムを構築し(3.5節)、3.6節、3.7節において協調ハンドリング実験を行い、提案手法の有効性を示す。

## 3.2 機構設計による問題解決

### 3.2.1 実ロボットの走行誤差問題への対応

はじめに示したように、実ロボットでは走行誤差が不可避であり、協調ハンドリング作業を行うなかで、これにより生じる過大な内力を回避する必要がある。

しかし移動ロボットの走行誤差に起因する内力を、速度サーボ系による高周期での正確な力制御により回避することは一般に難しい。

これに対し、走行誤差を受動的なコンプライアンスにより吸収するという機構的な解決策をとったものとして、移動マニピュレータに受動関節を導入し協調ハンドリングを実現した[Osuni96]や、2台の2DW型移動ベースをコンプライアンスを有するリンクで接続し高自由度移動機構を実現した[Borenstein93]などがある。

本研究でも機構的に解決を図る立場をとり、各ロボットの把持機構に水平面内に2自由度のコンプライアンス（バネおよびダンパ）を持たせることにする。また、平面内で2自由度の力のみを生成させ、不要なモーメントを発生させないため、コンプライアンス機構の根元を受動回転関節によってロボットに取り付ける。

### 3.2.2 非ホロノミック性への対応

ここでは、一般的な車輪型の移動機構を持つ移動ロボットを、本論文で提案する協調ハンドリング作業に適用することを検討する。

一般的な車輪型移動ロボットでは全方向移動ロボットの場合と異なり、“非ホロノミック性”という性質を有し、車軸中心において車軸方向の速度成分を生成することができないため、対象物をハンドリングする方向にも制限が生じ、問題が複雑である。

非ホロノミックなロボットを複数台用いた協調ハンドリングに関する研究には[Stilwell93]、[Kosuge99]などがある。ここでは分散型のリーダーフォロワ型のハンドリング手法を扱っており、フォロワロボットの把持位置を制御則上で車軸中心からオフセットをもった位置にあるものとしてフォロワロボットにキャスタと同様の動作を行わせることで、非ホロノミックなロボット間の協調を実現している。

しかし、この手法を用いても任意の方向へ対象物のハンドリングは不可能であり、対象物の動作計画が各ロボットの姿勢による限定を受ける。すなわち、ロボットの動作計画と分離することができず動作計画問題が複雑になり、持ち替え計画を伴う搬送の計画には

適さない。

そこで、搬送対象物を支持する位置（代表点）を、提案手法における全方向移動ロボットと同等に扱うこと、すなわちどの方向にも制御可能な点で物体を把持するという機構的な解決策を考える。

[Yamamoto94]においてYamamotoらが示しているように、把持機構の設置位置（代表点の位置）に車軸中心からオフセット  $L$  を持たせることで、常に非ホロノミック性によるハンドリング力方向の制限を回避することができる。例えば代表的な車輪型移動機構である2DW1C型のロボットの場合、Fig.3.1のように変数を置き、車輪の微小駆動量を  $[dr_1, dr_2]^T$  とすると

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{L}{d} & -\frac{L}{d} \end{bmatrix} \begin{bmatrix} dr_1 \\ dr_2 \end{bmatrix} = J_1 \begin{bmatrix} dr_1 \\ dr_2 \end{bmatrix} \quad (3.1)$$

$$\det J_1 = -\frac{L}{d} \neq 0 \quad (3.2)$$

となり、確かにこの点を常に全方向に制御することができる。同様の考え方を車輪自体に適用し車輪型移動機構による全方向移動ロボットを実現した研究もある（文献[和田97]）。

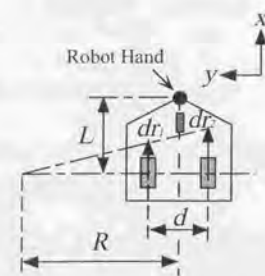


Fig. 3.1: 2DW1C type Wheeled Robot



## 3.2.3 提案する機構

以上の議論と、移動ロボットに共通の走行誤差による内力回避問題を考慮した車型ロボットによる協調搬送のための機構の概念図をFig.3.2に示す。なお、ここではロボットの把持機構の設置点位置のみを制御し、ロボットの姿勢については制御を行わない。

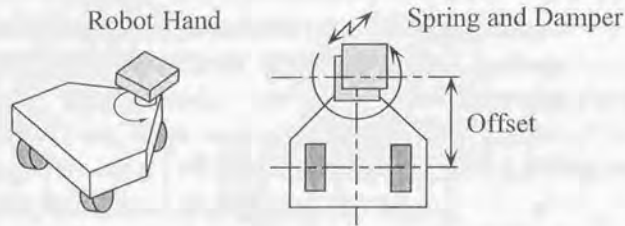


Fig. 3.2: Schematic view of the Handling Mechanism

## 3.3 移動ロボット群の動作計画アルゴリズムの拡張

## 3.3.1 トルク限界の考慮の仕方

搬送中の把持ロボットにかかるトルクは“走行誤差から生じる内力”と“対象物の運動を生成するためのハンドリング力”の和として表される。すなわち、

$$\text{“トルク限界”} = \text{“内力限界”} + \text{“ハンドリング力限界”}$$

となる。ここでトルク限界および内力限界は、ロボットのエンドイフェクタの特性、走行精度、床面との摩擦により計測・計算されるものである（その手法については3.7節で述べる）。

そこで、ロボットの動作計画では、ハンドリング力限界を考慮することになる。したがって、再度ロボット群の動作計画問題をまとめると、以下のように記述できる。

制約条件 1	対象物の目標位置・姿勢を実現すること
制約条件 2	ロボットの速度限界を守ること
制約条件 3（追加条件）	ロボットのハンドリング力限界を守ること
評価指標	障害物回避，物体の安定支持， 小エネルギーでのハンドリング
パラメータ	移動ロボット群の配置

本論文では、制約条件3を以下のように扱う。

- 制約条件3は、把持ロボットを選択する際に、そのロボットの組合せによってハンドリングが可能かどうかの判断に用いる。

これはすなわち、ハンドリング力が限界( $F_{lim}$ )を超えなければ良いという立場であり、環境に対する計画の柔軟性を優先させることを意味する。

## 3.3.2 各ロボットのハンドリング力の導出

ロボットの持ち替え動作計画において各ロボットが対象物に与えるハンドリング力の大きさを評価するため、ここでは対象物の目標動作（加速度および角加速度）が与えられ、かつ、把持ロボットが決められた時に、各ロボットが対象物に及ぼすべきハンドリング力を導出する手法を示す。対象物の目標動作は対象物の動作計画器（2.4節）の中で求められている。

まず変数の座標系およびロボット台数を以下のように定義する。

$\Sigma_W$  : 世界座標系

$\Sigma_O$  : 対象物座標系

$n_{rob}$  : 全ロボット台数

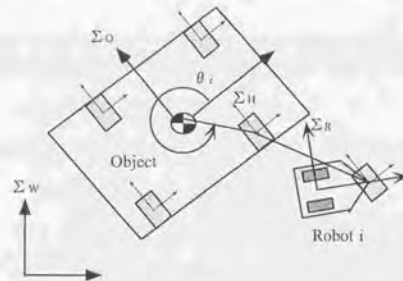


Fig. 3.3: Variables

以後用いる変数の左肩の添字はこの座標系を表し、特に指定の無いものは $\Sigma_W$ 上の変数とする。次に

$P_o := [x_o, y_o, \phi_o]$  対象物の位置・姿勢

${}^o p_i$  : ロボット*i*が対象物に力を作用させる位置

${}^o F_i$  : ロボット*i*が対象物に作用させるハンドリング力

と置くと、対象物の運動方程式は以下のように表せる。

$$M_o T \ddot{P}_o = W G {}^o F \quad (3.3)$$

ただし

$M_o$  : 対象物の慣性行列

$$T = \begin{pmatrix} \cos \phi_o & \sin \phi_o & 0 \\ -\sin \phi_o & \cos \phi_o & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$W = \begin{pmatrix} E_2 & \cdots & E_2 \\ {}^o p_1^T s & \cdots & {}^o p_{n_{rob}}^T s \end{pmatrix} \quad s = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

$$G = \text{diag}[G_1 \cdots G_{n_{rob}}]$$

$$G_i = \begin{cases} O & \text{ロボット } i \text{ が持ち替えロボット} \\ E_2 & \text{ロボット } i \text{ が把持ロボット} \end{cases}$$

$${}^o F = [{}^o F_1 \cdots {}^o F_{n_{rob}}]^T$$

Eq.(3.3)より、ロボットが対象物に与えるべき最小ノルムのハンドリング力は、疑似逆行列を用いて次のように求まる。

$${}^o F = (W G)^T ((W G)(W G)^T)^{-1} M_o T \ddot{P}_o \quad (3.4)$$

なお、上記導出方法では持ち替えロボットも含めた全てのハンドリングロボットに対して定式化がなされているが、行列 $G$ の導入により、持ち替えロボットのハンドリング力は0と導出される。



## 3.3.3 トルク限界を考慮した持ち替え計画アルゴリズム

上記で求めたハンドリング力を利用し、ロボットの配置計画、すなわち持ち替え計画を以下に示す手順で行う。

[Step 1]  $n_{rob}$  台のロボットから把持ロボットを  $m$  台 ( $m \geq 2$ ) 選び  $\sum_{m=2}^{n_{rob}} C_m$  通りの全ての組合せに対し次のプロセス Process 1 ~ Process 4 を試行する。

Process 1 把持ロボットの把持位置を固定したと仮定する。

(拡張部分) 各ロボットに対し必要なハンドリング力  ${}^aF_i$  を求める。  
 $\max_{1 \leq i \leq n_{rob}} |{}^aF_i| < F_{lim}$  ならば Process 2 へ  
 その他の場合 → “解なし” として Process 3 へ。

Process 2 ロボットの速度の制約条件下で評価関数  $PI$  を最小化する持ち替えロボットの最適な把持位置を求める。

Process 3 終了。

[Step 2] [Step 1] で求めた解のうち評価関数値が最も小さい場合を選ぶ。

ただし、

$F_{lim}$ : ハンドリング力限界

## 3.4 トルク限界を考慮した持ち替え搬送シミュレーション

## 3.4.1 シミュレーション内容

本章では、3.3 節で提案したロボットの動作計画アルゴリズムの有効性を検証するため、全方向移動ロボットによる持ち替えハンドリングシミュレーションを行う。

ここでは“ハンドリング力限界を考慮した場合のロボットの挙動への影響”を評価するため、異なるハンドリング力限界を与えたロボット群 S および W を用いて次の 3 種類のシミュレーションを行う。

Simulation 1: ロボット群 S による持ち替えを伴うハンドリング

Simulation 2: ロボット群 S による持ち替えを伴わないハンドリング

Simulation 3: ロボット群 W による持ち替えを伴うハンドリング

Simulation 2 は提案手法と比較するために行う。各シミュレーションにおける評価関数の係数や用いるロボット群等を Table 3.2 にまとめる。各ロボット群のハンドリング力限界は Table 3.1 に示す通りであり、ロボット群 S はロボット群 W より強いハンドリング力を対象物に与えることができるロボット群である。

その他のシミュレーション条件および環境は、全てのシミュレーションで Table 3.3, Fig. 3.4 に示すものを共通して用いる。図中の黒い多角形は障害物を表す。搬送開始時には障害物は未知であり、対象物の大域的な目標軌道にはスタートとゴールを等速直線運動で結んだ単純なものを与える。

Table 3.1: Handling Force Limit of each Robot Group

Robot Group	Handling Force Limit [N]
Robot Group-S	0.02
Robot Group-W	0.005

Fig. 3.5 にシミュレーション結果を示す。長方形は対象物を表し、小さい白い円は持ち替えロボットを、黒い円は把持ロボットを表す。ただし対象物の動作は三つの戦略で同一である。

Table 3.2: Contents of Each Simulation

Simulation	Weight Coefficient			Robot Group
	$\alpha$ (regrasping)	$\beta$	$\gamma$	
Simulation 1	0.2 (○)	5.0	4.0	Group-S
Simulation 2	0.0 (×)	5.0	4.0	Group-S
Simulation 3	0.2 (○)	5.0	4.0	Group-W

Table 3.3: Simulation Conditions

the shape of the object	0.8×0.6 [m] rectangle
center of gravity of the object	center of the object
the weight of the object	10.0 [kg]
the inertia of the object	0.83 [kg.m <sup>2</sup> ]
the number of the robots	4
the shape of the robots	circle (radius=0.15 [m])
sensing area of the robots	circle (radius=5.0 [m])
motion planning cycle	2.0 [s]

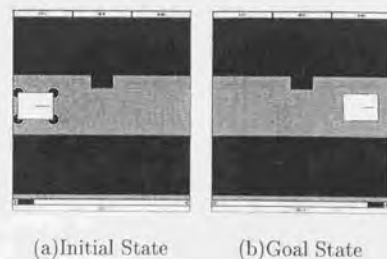


Fig. 3.4: Start and goal positions for the simulation

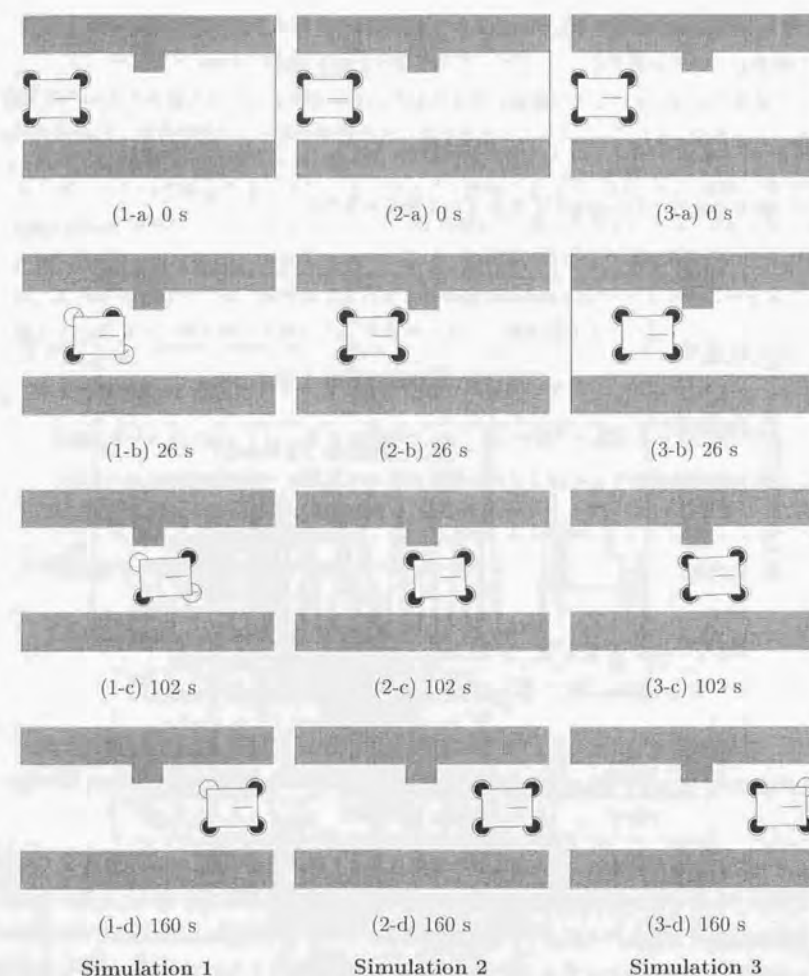


Fig. 3.5: Simulation Results



## 3.4.2 ハンドリング力限界の持ち替え計画への影響

Simulation 1 および Simulation 3 の比較により、ハンドリング力限界の動作計画への効果について説明する。

各瞬間でロボットにより対象物に与えられたハンドリング力のうち最大のものの時間変化を Fig.3.6 に示す。ハンドリング作業を通しての把持ロボットの平均台数は Simulation 1 で約 2 台、Simulation 3 で約 3 台であった。グラフより、各ロボット群のハンドリング力限界が満たされた計画が行われている様子が分かる。

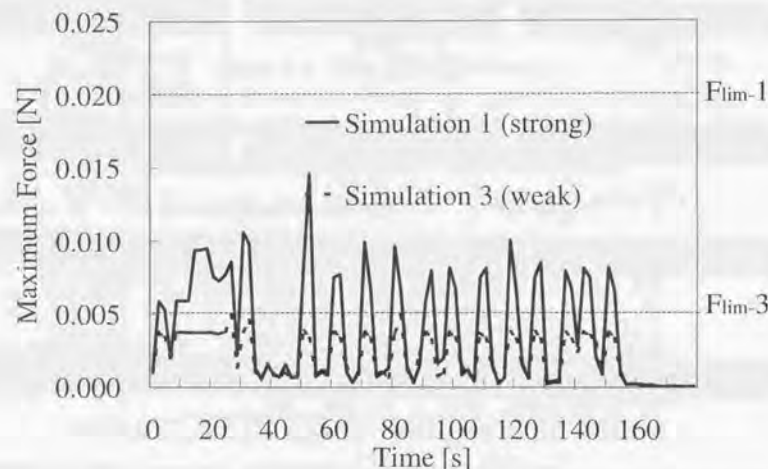


Fig. 3.6: Transition of Maximum Handling Force applied by Robot Group

またこのグラフにより Fig.3.5 における両ロボット群の挙動の違いが説明できる。26 秒において両ロボット群は障害物に接近しており、障害物からの斥力を受けて計画された対象物の運動は大きなハンドリング力を必要としている。この時 Simulation 1 ではロボット群は 2 台で物体をハンドリングする余裕があるため残り 2 台のロボットが持ち替えを行っている。一方 Simulation 3 では対象物の計画動作を実現するために全てのロボットが対象物を把持しハンドリングに従事しなければならない、Simulation 1 のロボット群よりも障害物に近接した場所を通過している。

このことから、

- 周囲に障害物が存在する領域において、ハンドリング力限界の大きいロボット群ほど、障害物を回避した安全な動作が計画されている

ことが分かる。

しかし、評価関数における  $d_1$ ,  $d_2$ ,  $d_3$  の時間変化を表す Fig.3.7, Fig.3.8, Fig.3.9 およびこれらの平均値をまとめた Table 3.4 から分かるように、20 秒から 26 秒にかけて Simulation 3 のロボットは持ち替えを行う事が出来ず、持ち替えを行わない Simulation 2 のロボットとはほぼ同じ  $d_1$  の値をとっているが、その他の時間帯では持ち替えの効果が現れ、 $d_1$  の平均値としては、持ち替えを行わない Simulation 2 のロボット群よりも大きな値をとっており、障害物から離れた位置を通過している事が分かる。

以上の評価より、提案したロボット群の動作計画器により、

各ロボットの“ハンドリング力の限界”という拘束条件を満足する範囲で“計画された対象物の運動”を実現する周辺環境に応じたロボットの行動計画が可能である

ことが分かり、その有効性が示された。

Table 3.4: Average Distance of  $d_1$ ,  $d_2$  and  $d_3$  [m]

Distance	Simulation 1 (strong robots)	Simulation 2 (without regrasp)	Simulation 3 (weak robots)
$d_1$	0.231	0.176	0.227
$d_2$	0.284	0.300	0.290
$d_3$	0.426	0.500	0.444

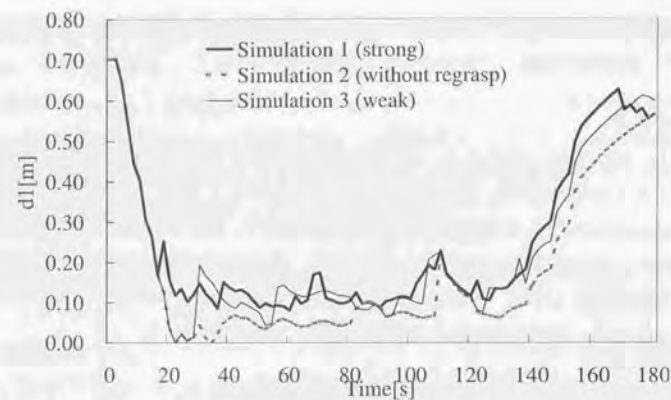


Fig. 3.7: Transition of  $d_1$

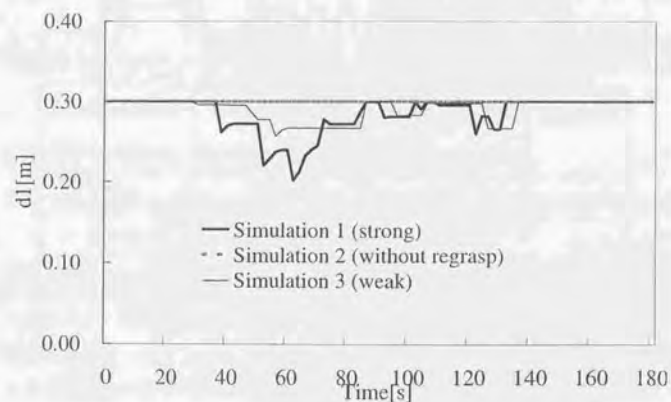


Fig. 3.8: Transition of  $d_2$

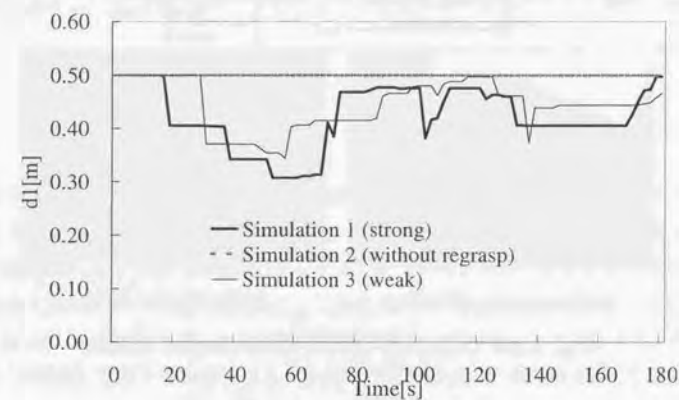


Fig. 3.9: Transition of  $d_3$

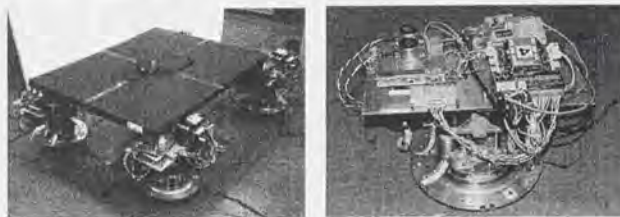


## 3.5 協調ハンドリング実験システムの構築

本実験では各ロボットの計算処理能力が低いことから以下の方向でシステムを構築する。

- 環境情報を取得するための外部カメラシステムを用いる。
- 動作計画およびロボットの速度指令の部分はホストコンピュータを導入してこれに行わせる。
- 実際の機構への速度指令導出および駆動部の制御を行う部分のみロボットに行わせる。

構築した実験システムの外観を Fig.3.10 に、構成を Fig.3.11 に示す。



(a) Cooperative Handling

(b) Profile of a Robot

Fig. 3.10: Overview of the Experimental System

## 3.5.1 把持機構の製作

3.2節で考慮したようにコンプライアンスを導入した把持機構を実際に試作し、車軸から100[mm]のオフセットをとった位置に搭載した。

受動コンプライアンス部は Fig.3.12 に示すスプリングユニット部で実現した。これはユニットの両側から圧縮バネを挿入して挟み込む機構であり、バネの変位（最大ストローク  $\pm 15$ [mm]）は内蔵した2つのリニアポテンシオメータでそれぞれ検出することが可能となっている。バネ定数は約  $0.1$ [kgf/mm] である。ダンパ機構は特に用いず、ポテンシオメータ等によって起こるダンパ効果を利用した。

スプリングユニット部をロボットに接続する部分について、ロボットが対象物を強く把持している間は対象物にモーメントを与えないよう受動回転関節となっている必要がある。

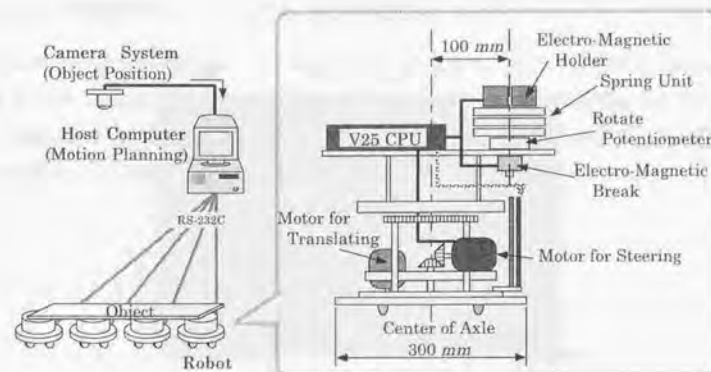


Fig. 3.11: Hardware Structure of the System

が、持ち替えを行っている間には対象物との接触により必要以上に把持部が回転してしまわないよう、電磁ブレーキを取り付けている。このとき把持機構のロボット座標系に対する姿勢は接続部にとりつけた回転ポテンシオメータで計測することができる (Fig.3.13)。

持ち替え動作については、対象物として発泡スチロールに鉄板を張り付けたもの (600 × 800 [mm], 1.85[kg]) を用い、把持機構側の電磁ホルダのオン・オフにより、把持状態 (把持/非把持) を切り替え実現する。なお、動作計画は1計画周期2秒として行った。

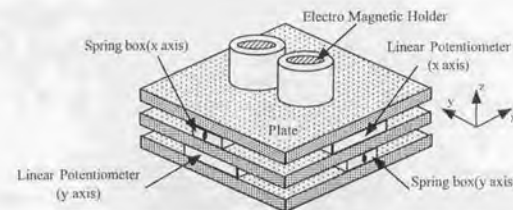


Fig. 3.12: Schematic View of Developed Spring Unit

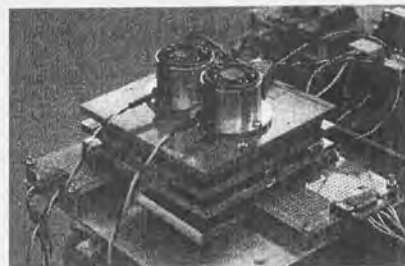


Fig. 3.13: Overview of the Developed Handling Mechanism (Spring Unit)

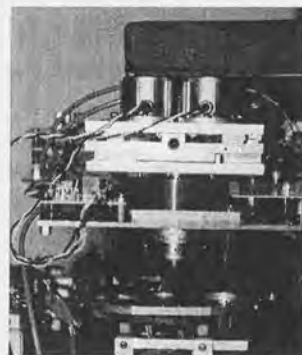
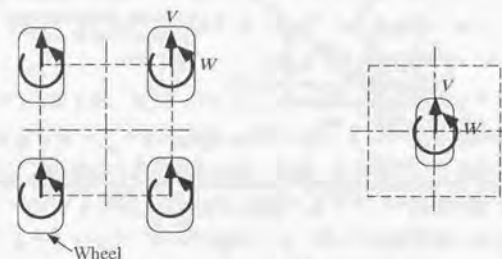


Fig. 3.14: Elevation of the Robot Hand (Connected Part)

## 3.5.2 車輪機構のモデル化および制御則

今回実験に用いた移動ロボット (Fig.3.15(a)) は正方形の四頂点にそれぞれ配置された車輪が同時に駆動および操舵されるため、車輪機構のモデルとしては Fig.3.15(b) のように、駆動および操舵される車輪が1つ中央に配されたものと捉える事ができる。把持機構は車軸中心より 100[mm] のオフセットを持たせて設置してある。



(a)Wheels of the Experimental Robot

(b)Modeled Wheels

Fig. 3.15: Modeling of a Wheel Mechanism of a Robot in Experiment

モデル化されたロボットへの車輪速度入力とは並進速度  $v$  と操舵速度  $w$  の2種類である。

Fig.3.16に変数を示す。これらの変数は全て各ロボットのロボット座標系  $\Sigma_R$  におけるものとする。幾何的に解くため、把持機構位置の速度指令を微小時間  $\Delta t$  における微小変位  $\Delta x$  と  $\Delta y$  に変換する。

$\Delta y \neq 0$  の場合、旋回半径  $R$  と  $\sin \theta$  は次のように表せる。

$$R = \frac{\Delta x^2 + \Delta y^2 + 2L\Delta x}{2\Delta y} \quad (3.5)$$

$$\sin \theta = \frac{R\Delta x + L\Delta y}{R^2 + L^2} \quad (3.6)$$

これらの値を用いると、車輪駆動速度は以下のように計算できる。

$$v = R\omega \quad (3.7)$$

$$\omega = \frac{\theta}{\Delta t} \quad (3.8)$$

$\Delta y = 0$  の場合、車輪駆動速度は以下のように表される。

$$v = \frac{\Delta x}{\Delta t} \quad (3.9)$$



$$\omega = 0$$

(3.10)

微小変位  $\Delta x$  および  $\Delta y$  が負の場合も上式はそのまま成り立つ。

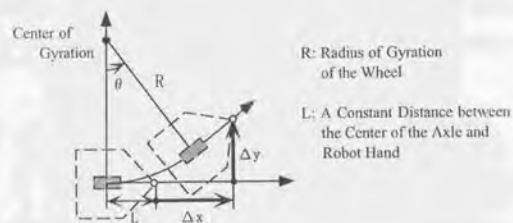


Fig. 3.16: How to derive Wheel Commands to the Experimental Robot

### 3.6 搬送機構の評価実験

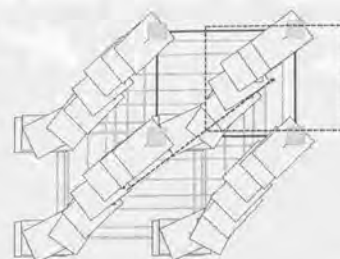
3.2節で設計した搬送機構の有効性検証のため、非ホロノミックな方向への持ち替えを伴わない協調ハンドリング実験を行った。

実際に製作した機構の有効性を検証するため、全ロボットの初期姿勢前方から  $45[\text{deg}]$  左の方向にハンドリングさせる実験を行った。Fig.3.17(a)に対象物とロボットの軌跡を示す。大きな長方形が対象物、小さな白い長方形がロボットの本体、灰色の正方形が把持機構を示す。点線の長方形が目標到達位置であり、太い実線は実際の到達位置である。本実験ではあらかじめシミュレーションにより計算した速度指令をロボットに与えた。

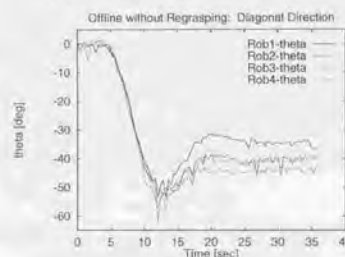
全ロボットが対象物を堅く把持していることから、ハンドリング中に対象物と把持機構の相対姿勢は変わらない。一方対象物は姿勢を変えことなく斜め前方のゴールへとハンドリングされるよう計画されているため、ロボット本体と把持機構の姿勢のずれだけが変化していくことになる (Fig.3.17(b))。協調ハンドリング作業の前半ではロボット本体の姿勢が徐々に変化していき、作業の後半ではロボットは対象物に対する相対的な姿勢を保って斜め方向に直進していることが分かる。対象物位置に関するフィードバックを行わなかったため、最終到達位置では絶対座標系  $x$  方向に  $282[\text{mm}]$ 、 $y$  方向に  $26[\text{mm}]$  のずれが生じた。

また Fig.3.17(c)の把持機構内部のコンプライアンスの変位を見ると、速度指令がオフラインで計算されておりコンプライアンス変位を減少させるフィードバックは行っていないため変位は徐々に溜っていく。しかしコンプライアンスのストロークに対し十分小さく、走行誤差による内力をコンプライアンスが吸収してハンドリング作業を完了した様子が分かる。

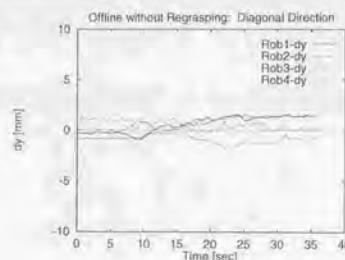
これらの結果から、回転の受動関節とともに搬送機構が有効に機能することが示された。



(a) Loci of Object and Robots



(b) Rotate Angle of Robot's Hand Coordinate



(c) Displacement along Y-axis on Robot's Hand Coordinate

Fig. 3.17: Experimental Results using Car-like Mobile Robots

### 3.7 持ち替え協調ハンドリング実験

#### 3.7.1 障害物の存在しない環境における協調ハンドリング実験

まず,

1. 動作計画アルゴリズムの実時間有効性
2. 評価関数の設計の妥当性
3. トルク限界を考慮した動作計画アルゴリズムの有効性

の検証を目的として、障害物の存在しない環境で、最適でない把持位置から始まる協調ハンドリング実験を行う。これは、障害物の影響で持ち替えを行った後、障害物が消失した時点での把持位置の計画の例にあたる。本実験では外部カメラシステムによる対象物位置のフィードバックを行っている。ここで本実験における総トルク限界は約12[N]である(ロボットの把持機構位置を中心に水平面内の全方位についてバネ計りにより押し/引き、一番弱い方向での測定値をとった)。ロボットの走行誤差が1計画周期で $\pm 5$ [mm]程度と見積ると走行誤差による内力は最大で10[N]となり、ハンドリング力限界は2[N]と設定する。

Fig.3.18の(a)に初期状態(実線)と目標位置(点線)を示す。初期状態でロボットは対象物の重心に対し正方形をなす位置を把持している。協調ハンドリング作業開始後、4~6秒にロボット4が(Fig.3.18(b))、10~12秒にロボット2が(Fig.3.18(c))、26~28秒にロボット1が、28~30秒にロボット3が(Fig.3.18(d))それぞれ持ち替えを行い、最終状態では対象物の各頂点を把持している。

まず(1)について、走行結果によって、計画周期として設定した単位時間内に計画、観測、制御が実現でき、かつ、目標位置と到達位置とのずれは絶対座標系 $x$ 方向に-23[mm]、 $y$ 方向に0.0[mm]と、おおむね計画通りに走行していることが確認できた。このことから、計画能力、搬送機構能力という観点からの、提案手法の実機への適用可能性が示された。

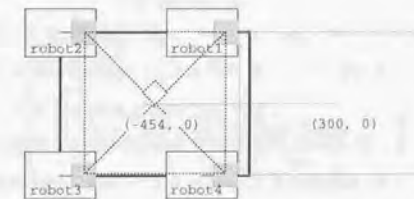
また本実験のように、周囲に障害物が存在しない環境において長方形の対象物を4台のロボットが把持する場合、2.5節で示した評価関数のうち、物体の安定把持および小エネルギーでのハンドリングを評価する第2項および第3項から、長方形の各頂点が最適な把持位置と言える。したがってFig.3.18の持ち替え結果から、評価関数の設計によって、実機システムにおいても、適当な初期配置に始まり“対象物の安定把持を維持”、“小エネルギーでのハンドリング”の両者を満たす配置に自動的に収束することが示された。障害



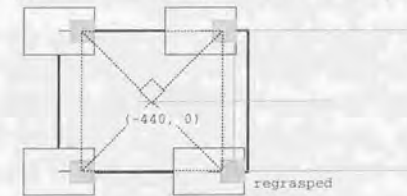
物存在下での回避動作についてはここでは評価していないが、ロボットの動作を生成する過程はほとんど同様であるため、障害物存在下においても同じ効果が得られると考えられる。したがって(2)の評価関数の設計の妥当性が示された。

また(3)について、協調ハンドリング作業中に把持機構の受けた最大トルクは、Fig.3.19に示すポテンシオメータの計測値から3.2[N]であった。本実験の場合、計画された対象物の運動に必要なハンドリング力は最大0.0074[N]と小さく、トルクのほとんどは走行誤差によるものではあったが、ハンドリング中の総トルク限界が満たされてタイヤの滑りは起こらず、振動も見られなかった。これにより、提案手法はトルク限界を考慮した有効なアルゴリズムであることが実機において示されたと言える。

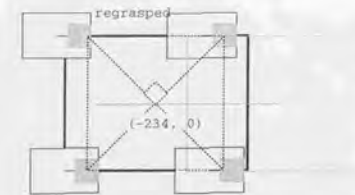
以上より、提案手法が持ち替えを伴う協調ハンドリング作業の実現のために有効であることが示された。



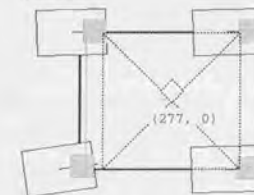
(a) Initial State(0.0 s) and Goal Position[mm]



(b) 6.0 s



(c) 12.0 s



(d) Final State(30.0 s)

Fig. 3.18: Experimental Results 2

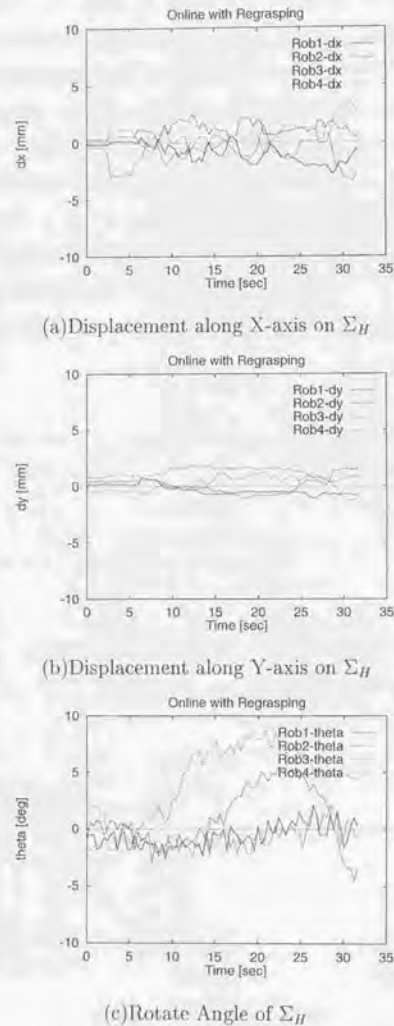


Fig. 3.19: Displacement Data of Each Compliance Unit

## 3.7.2 障害物の存在する環境での協調ハンドリングシミュレーション

次に、本章で提案した動作計画手法により、未知障害物の存在する環境において、非ホロノミック性を有する移動ロボットによる持ち替えを伴う協調ハンドリング動作が有効になされることを、シミュレーションによって示す。

シミュレーション環境（初期状態および対象物の目標位置）を Fig.3.20 に示す。

Fig.3.21に3.4節と同様の表記でシミュレーション結果を示す。ただし図中の黒く塗りつぶした多角形領域は障害物を表す。ハンドリング作業は160秒間で終了している。

0秒においてハンドリング作業が始まる (Fig.3.21-(a))。まず、進行方向に対し左前のロボットが障害物を発見し、障害物の回避のため持ち替えを行うと (Fig.3.21-(b))。他のロボットもこれに伴い安定把持を崩さないように持ち替えを行い (Fig.3.21-(c)) 障害物となっている通路の凸部のわきをほぼ通過する頃には、4台の把持する位置がちょうど頂点一つづつずれた所に変わっている (Fig.3.21-(d))。その後、対象物の後ろ側を把持しているロボットが障害物を回避する方向に把持位置を少し変え (Fig.3.21-(e))、最終的には初期状態と同様に障害物の存在しない場合に最適な状態に把持位置が収束している (Fig.3.21-(f))。

これらの結果から、非ホロノミック性を有するロボットを用いた場合も環境に応じた持ち替え計画がなされることが分かった。

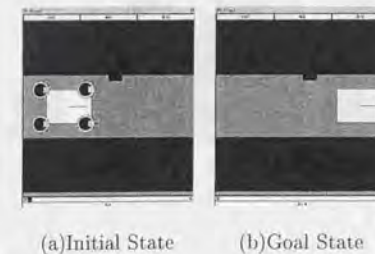


Fig. 3.20: Simulation Environment



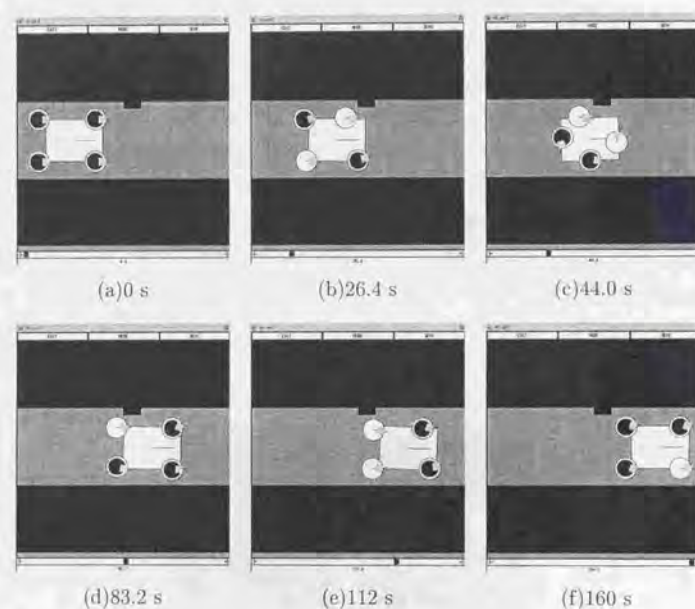


Fig. 3.21: Simulation Results 2

### 3.8 おわりに

本章では、物理層に関する課題解決として、未知環境における持ち替え動作を伴う協調ハンドリング動作を実ロボットで実現する際の課題のうち、力の干渉を伴う協調ハンドリング動作実現のためのものとして、次の3点に対する解決方法を扱った。すなわち、“トルク限界” “走行誤差による過大な内力” “非ホロノミック性” という3つの課題をとりあげ、機構および動作計画アルゴリズムの両面からこれを解決する手法を示した。

まず、移動ロボットの非ホロノミック性による搬送方向の制限および走行誤差に起因する過大な内力発生に対応するための機構として、車軸中心からオフセットを持つ位置に受動コンプライアンスを有する把持機構を設置する事を提案した。

次に、2章において提案したロボットの動作計画手法を拡張する形で、トルク限界を考慮した動作計画アルゴリズムを提案した。その際、環境に対する柔軟性を重視した計画となるよう、与えられた対象物の運動を実現するために各ロボットが及ぼすべきトルクを導出し、これがトルク限界を満たしている把持ロボットの組合せに対し、最適な把持位置を求め、これらの評価関数値を比較した。また、力の強いロボット群および弱いロボット群を用いた協調ハンドリングシミュレーションにより、ハンドリング力限界が計画にどのように影響するかを検証した。

これらにより協調ハンドリング作業が実現可能である事を示すため、協調搬送機構を製作して実ロボットに搭載し、協調ハンドリング実験を行った。まず、設計した搬送機構の有効性を示すため、非ホロノミックな方向への移動を必要とする協調ハンドリング実験を行った。実験より、車軸からオフセットを持った位置に設置した、受動コンプライアンスを含む搬送機構が、有効に機能していることが分かった。

また、障害物の存在しない環境における、評価関数値が最適化されていない把持位置からの持ち替え動作を伴う協調ハンドリング実験を行った。実験より、トルク限界を満たした持ち替えを伴うハンドリング作業が実際に実時間で計画および実現可能であることが分かった。

以上より、局所的に環境情報を取得する事が可能ならば、これに基づく最適な把持位置(持ち替え動作)を実時間で動的に計画可能であり、また、走行誤差が存在しハンドリングに費やすトルクに限界のある実ロボットにより実現可能であることが示された。

## Chapter 4

### 異種作業割り付け型協調搬送システムの計 画器アーキテクチャ

---

4.1	はじめに	92
4.2	タスク割り当てを含む行動計画器の課題	94
4.3	計画器アーキテクチャ	96
4.4	タスクの記述	102
4.5	優先度を用いたタスク割り当てアルゴリズム	107
4.6	タスクの実装	111
4.7	未知静的環境における協調搬送シミュレーション	123
4.8	未知動的環境を想定した作業への適用	137
4.9	提案手法の適用範囲の限界について	143
4.10	持ち替えを伴う協調搬送シミュレーション	146
4.11	おわりに	152

---



## 4.1 はじめに

本章では情報層に関する課題であるタスク割り当て決定過程の問題解決を図る。

2章および3章では、対象物周辺の環境情報を取得できるという仮定のもとに、その情報に基づき群ロボットが共通の物体をハンドリングするという、未知環境における協調搬送システムの中核となる動作について扱ってきた。しかし、実際には取得情報の処理に多かれ少なかれ時間がかかり、また各ロボットの観測可能な領域には限界がある。したがって取得可能であると想定した環境情報を実際に取得するためには、ロボット間の協調を考慮するなど、取得過程をうまく組み込んだ行動計画が必要となる。

そこで本章では、協調ハンドリングや障害物除去に加え、周辺監視など情報取得動作をそれぞれ作業（タスク）の1つと捉え、ロボットが必要に応じてタスクを切替える“異種作業割り付け型協調搬送システム”の計画器全体の構造を提案する。

計画器では、各ロボットへのタスク割り当てと、そのタスクを実現する具体的な動作を、ロボット群が周辺を観測した結果に基づき実時間で決定する必要がある。本論文で扱う搬送作業では、物体のハンドリングよりも先に軌道を塞ぐ障害物を除去する、物体のハンドリングと同時にあるいはそれより先に周辺を監視するなど、異種タスク間での実行の順序やタイミングに関する条件を考慮する必要がある。本論文ではこれを“タスク間の相互関係”と呼ぶ。

本章では、まず4.2節において計画器アーキテクチャが解決すべき課題を示し、4.3節で提案する計画器アーキテクチャの概要と課題の解決方法を示す。続いて、タスク割り当てのためのタスクの記述方法（4.4節）およびこれを用いた割り当てアルゴリズムの詳細を示す（4.5節）。提案した計画器アーキテクチャの有効性を検証するため、代表的なタスクを実装し（4.6節）、未知環境における協調搬送シミュレーションを行う（4.7節）、また2章において提案した持ち替え動作を伴う協調ハンドリング動作計画手法を組み込み、同様の環境における搬送シミュレーションを行うことで、本論文の目指す未知環境における搬送システムの全体としての有効性を検証する（4.10節）。

なお、以下の議論を進めるに当たり、

- ロボットは、各瞬間において、ある方向の限定された距離までの物体の相対位置および形状が認識可能
- ロボットは互いに自由に通信可能

という仮定をおく。

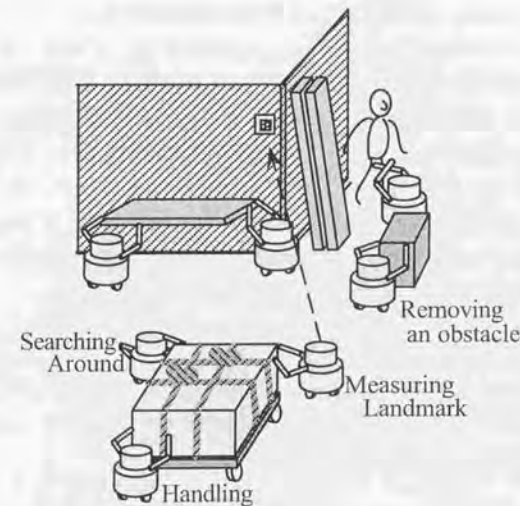


Fig. 4.1: Cooperative Transport in an Unknown Environment

## 4.2 タスク割り当てを含む行動計画器の課題

異種作業割り付け型協調作業のための行動計画器のうち、情報層で行われるタスク割り当て過程に着目すると、以下に示す3つの特徴を扱う能力が求められる。

まず、未知環境において作業を遂行するために必要なタスクの“性質”という観点から、以下の2つの特徴を扱う能力が求められる。

### (1) タスクの動的な変化

目的達成までにロボット群が満たすべき要求は複数種類存在する。しかし、それらの全てが常に必要というわけではない。例えば、走行精度が悪化した場合、誤差補正のためにランドマーク等を計測する必要性が生じる。また、搬送経路を占領する障害物を発見した場合、それを除去する必要性が生じる。このように、ロボット群の移動にしたがって必要なタスクの種類は変化する。また、各タスクは空間的に広がりを持って分布し、同じ種類のタスクでも、タスクの質、すなわちそのタスクがロボット群に対して持つ意味合いは異なる。例えば、それまでまったく観測を行っていない領域の監視とすでに観測を行った領域の監視では、ロボット群にとって情報が更新された場合の利得が異なる。このような変化を反映して必要なタスクを素早く抽出する必要がある。

### (2) ロボット台数を越える大量のタスク

(1)でとりあげたタスクの変化に伴い、実行すべきタスクの総量は変化し、場合によっては、ロボット群がその瞬間に実行可能な量以上のタスクが存在する。そのため、行動計画器は次の三点を解決する必要がある。

2-1 タスクの相互関係を考慮した適切な順序でのタスク選択・実行

2-2 割り当てられなかったタスクの解決

2-3 必要以上のタスク増加の抑制

2-1について、ロボット群の実行可能な量を越えてタスクが存在する場合、適切な順序で選択し実行する必要がある。まず、異種タスク間の実行順序に関する相互関係を守る必要がある。実行順序とは、例えば、物体のハンドリングよりも先に軌道を塞ぐ障害物を除去する、といったものである。また同種のタスクであっても、例えば搬送システムの核となる共通物体のハンドリング作業など、ロボット間の密な協調のうえに成り立つタスクが含まれるため、最低限必要なロボット台数を確実に確保する、といった考慮が必要となる。

次に2-2について、その瞬間に割り当てられず実行されなかったタスクでも、必要性が継続している場合、当然そのタスクを実行する必要がある。

また選択順序が適切な場合でも、タスクの増加に処理が追い付かない状態では計画器の破綻を招くため、2-3に挙げたようにタスクが必要以上に増加し溢れることを、可能な範囲で抑制する必要がある。

一方、これらのタスクはロボットが実際に環境内を移動しながら解決する必要がある。したがって移動しつつ作業を行うロボットのための計画器という観点から、三つ目の特徴として、

### (3) 実時間での動作決定

を実現する必要がある。



## 4.3 計画器アーキテクチャ

### 4.3.1 アーキテクチャの検討

#### 意図の矛盾による協調動作の破綻防止

未知環境で協調作業を行う場合、ロボットがそれまでに環境を観測して得られている情報に基づき、その場で行動を決定する必要がある。この時、各ロボットが単体で取得可能な情報は、自己周辺の限定された範囲に関するものであり、互いにその内容は異なる。しかし、本論文で扱う協調搬送システムでは、協調ハンドリング作業のように、ロボット間の力の干渉を伴い密な協調を必要とする作業が含まれる。この場合、ロボットが予告なしに突然対象物を離れるなど、ロボット同士で行動の意図に矛盾が生じると、物体が落下するなど協調動作が破綻し、非常に危険である。

ロボット間でのタスク割り当て問題に関する従来研究のうち、矛盾を防ぐものとして、[Noreils93]や[Asama92]といった、契約ネットを用いた交渉に基づく手法がある。これらは互いの合意のもとに最終的な計画がなされるため、計画のうえでの干渉はない。しかし交渉過程に時間がかかるため、行うべきタスクが頻繁に変化する場合には、適用が困難である。

行動決定の過程が高速なものとしては、[Parker98]のように、ビヘービヤバーストロボットの枠組の中で、各ロボットが自身の観測した情報のみに基づき行動を決定する方法がある。この手法ではロボット間の通信コストが低く、ロボット台数が多い場合、あるいはタスク同士が独立している場合には有効である。しかし、交渉過程を持たないため、同一の作業を行うロボット台数の確保、矛盾の無い障害物回避方向の決定などが保証できず、密な協調作業のためのタスク割り当て手法としての適用は困難である。

本研究では、ロボット同士の意図の矛盾による計画上的協調動作の破綻を防ぐため、行動を決定する要因となる環境情報を各ロボットから収集・統合した上で共有し、これに基づき、ロボット群全体の動作を同時に計画することにする。計画を行う主体により、

- ロボットのうち1台が群全体の計画を行い、結果を他のロボットに通信する

あるいは

- 全てのロボットが同質の計画アーキテクチャを持ち、それぞれが群全体の計画を行い、そのうち自分に対応する部分を実行する

という形態が考えられる。

#### タスクの単位と実行順序

上記のように統合した環境情報を用いて計画を行うが、得られている環境情報は対象物周辺という局所的なものであり、長期に渡ってのロボット群の行動を求めても、環境が変化した場合に無意味となる可能性が高い。そこで、2章で行ったように、一定単位時間毎に行動の再計画を行い、目標状態を達成するまでこれを繰り返すことを考える。

この時、タスク同士の実行順序を守ったタスク割り当て計画を行い、各タスクに必要なロボット台数を必ず確保可能な方法として、次の方法が考えられる。すなわち、単位時間毎に、最も優先度の高いものから順に少しずつ実行していく、というものである。しかし、ロボット群の一部が通行の障害となる物体を除去し、並行して残りのロボットが物体を協調ハンドリングするという割り当てで動作を行うなど、効率の良い計画を行うことができない。

ここで、割り当てるタスクを“1～数単位時間で1台のロボットが実行可能な具体的な動作”という単位（これをタスクインスタンスと呼ぶ）で捉えらる。また、この時、あるタスクインスタンスをあるロボットに、どの程度優先的に割り当てるべきかという評価（これをタスクの優先度と呼ぶ）が可能であると仮定する。その場合、ある状況におけるタスクインスタンスとロボットの最適な対応関係は、いわゆる“割当問題”という一種の線形計画問題として高速に計算可能となる。また、そのタスクを実行するロボットの具体的な動作も、1単位時間に限定すれば高速に計画可能であり、これらを積み重ねることで目標地点に至る事が期待できる。

上記の評価、つまり優先度は、状況によって不変な要素および変動する要素により決定可能であると考えられる。不変な要素とは、あるタスクを行う前に別のタスクを行っておく必要があるといった、タスクの性質に基づく実行順序により定まる要素である。

変動要素としては二つの側面が考えられる。

一つはタスクが協調動作であるための要素で、そのタスクにとって十分なロボットが確保されるかどうかという割り当て可能なロボット台数との関係である。これは、最低台数を確保できないならば、これを後回しにし他のタスクを先に実行した方が、作業全体を終えるまでの時間という観点から効率が良い場合がある、という考えに基づいている。

もう一つはそのロボット群およびタスク発生地のコンフィギュレーションの関係である。これは、ある領域の監視をしようとしても対象物が邪魔で観測できないなど、作業空間内のロボット群のコンフィギュレーションによって、各ロボットが効率良く実行可能なタス



クは異なる、という考えに基づいている。

また、群ロボットが実際に環境内を移動する過程でタスクが生成することから、群ロボットの中心である対象物の移動の抑制により、タスクの生成抑制という効果が期待できると考えられる。

### 4.3.2 提案アーキテクチャの概要

これらを踏まえ、本論文では前出の3つの課題(1)(2)(3)を解決する枠組みとして、Fig.4.2に示すアーキテクチャを提案する。

これは、

(Step 1) 割り当て可能なロボットの確認

(Step 2) 各ロボットの得た環境情報の融合

(Step 3) 環境情報に基づくテンプレートを用いた必要タスクインスタンスの生成

(Step 4) 優先度に基づくタスク割り当て

(Step 5) タスク毎の動作計画

(Step 6) 割り当てられなかったインスタンスの消去

(Step 7) 各ロボットによる実行

という一連の過程を一定単位時間毎に繰り返し行うものである。再計画を行う単位時間間隔を、“計画周期”と呼ぶ。必要なタスクインスタンスは次の計画周期で環境情報をもとに再度生成される。

設計者はタスク毎に次の二つを準備する。

一つは“タスクテンプレート”である。これは、環境情報に基づき次の計画周期にどのような動作（タスクの種類と優先度）がどの程度（ロボット台数）必要かを判断しロボット1台に対応した形態のタスクインスタンスを生成するために用いる。内容の詳細については4.4節で説明する。

もう一つは、各ロボットの具体的な動作を計画する“動作計画器”である。動作計画器は、対象物周辺という局所的な環境情報のみに基づき、1計画周期内のロボットの動作を実時間で計画可能である必要がある。本論文では2章において協調ハンドリング動作計画手法を提案したが、これは上記仕様を満たしており、物体の協調ハンドリングタスクの動作計画器として適用することができる。

なお、動作計画器には、必要以上に多数のタスクが生成するのを抑制する仕組みとして、他のタスクの進行・達成状況に応じて、移動を抑制する仕組みが備わっている。

上記処理に関する補足として、生成過程においては、ロボット台数やどのタスクを優先させるといった複雑な関係は考慮されず、必要なタスクはすべて生成される。

また、環境情報に関して、ロボット群および搬送対象物の現在位置や、ある領域に関する情報の古さなどの時間的な情報等をすべて、環境情報として扱う。

また、タスク割り当て器においては、各ロボットに必ずなんらかのタスクを割り当てるが、動作計画器において具体的な動作を計画した結果、割り当てられたロボットが必ずしも物理的效果を伴う行動をとるとは限らない。例えばハンドリングタスクの場合、周辺の監視状態等に基づきその場に停止待機が計画され、正の速度指令が計画されない場合がある、という事を意味している。

### 各層の計画・制御周期の関係

情報層におけるタスク割り当て計画の切り替わる周期（上記の計画周期）を  $T_{information}$ 、幾何層における動作計画の切り替わる周期（例えば2章の動作計画周期）を  $T_{geometric}$ 、物理層における各ロボットの動作制御周期を  $T_{physical}$  とする。動作計画の切り替わる周期を、参照する環境の幾何情報が切り替わる周期と考えると、本論文におけるこれらの周期は、

$$T_{information} = T_{geometric} \geq T_{physical} \quad (4.1)$$

という関係になっている。

$T_{information}$  および  $T_{geometric}$  の長さは、ロボットの最大移動速度やセンシングエリアといった仕様に基づき、環境の変化を検知した際に対処可能な時間間隔であるように決定される。 $T_{physical}$  については、ロボットの性能により決定されるものであり、一般に短いほど精度の高い制御が可能である。

なお、以降で示すシミュレーションおよび実機実験では、 $T_{information}$  および  $T_{geometric}$  は3秒、 $T_{physical}$  は0.1秒程度となっている。



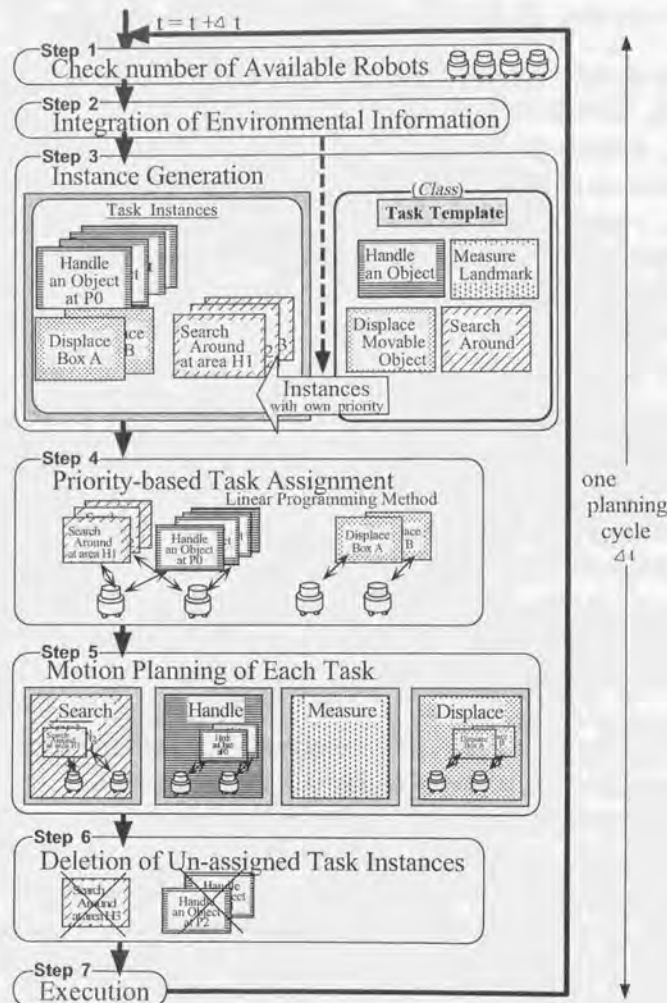


Fig. 4.2: Profile of the Planning Process

## 4.3.3 提案アーキテクチャによる問題解決

本アーキテクチャにより、3つの課題はそれぞれ以下のように解決される。

“(1) タスクの動的な変化”に対して：

一定計画周期毎に、タスク毎に準備されたタスクテンプレートに環境情報を入力しタスクインスタンスを生成するため、タスクの種類および質の変化が反映される。

“(2) ロボット台数を越える大量のタスク”に対して：

タスクインスタンスを生成する度に固有の実行順序や環境との関係を反映した優先度が評価され、未割り当てのタスクインスタンスも次の周期に反映されるため、実行可能なものから順次時系列で解決可能となっている。

また、一般にロボット群が移動した結果タスクが生じてくることを利用し、動作計画器においてタスクの達成状態に応じてロボットの移動を抑制することで、必要以上のタスク生成の抑制が可能である。

“(3) 実時間での行動決定”に対して：

必要な作業を、短時間に1台のロボットが行う作業であるタスクインスタンスとして生成するため、優先度というパラメータを用いた評価によりいわゆる“割当問題”として線形計画法を用いて解くことが可能となった。また動作計画も短期間の行動推定のみに基づいて行うため、計画が実時間で可能である。

## 4.4 タスクの記述

### 4.4.1 タスクとは

本論文では、搬送システムとしての目的、すなわち対象物の目標コンフィギュレーションの達成までに要求される、観測過程を含めた様々なロボットの行動をタスクとして扱う。

ここで、各ロボットは、各自を構成するセンサやアクチュエータといった資源を一定時間運用することで、ある種の作業を達成する“能力”を持つ。本論文では移動ロボットとして、多自由度のマニピュレータを持たず、周辺の限られた範囲内の障害物情報を観測可能なものを想定している。したがってその能力は、“移動”および“観測”と捉えることができる。

本論文では、上記能力のいずれかを短時間行使することで達成する作業のうち、観測データから抽出を望む情報など、実行する目的の異なる作業を異種タスクとみなす。例えば、 $T_1$ :搬送物体のハンドリング、 $T_2$ :既知ランドマークの計測による自己位置の同定、 $T_3$ :周辺の監視、 $T_4$ :可動障害物の除去、などは異種タスクとみなされる。

実行する目的が異なる場合、情報層のレベルでは、

- どのような状況で何台のロボットが必要か、といった生成・消滅の過程
- 他のタスクを実行する場合、あるいは実行するロボットが異なる場合など、そのタスクを割り当てた状態の評価

が異なり、幾何層のレベルでは、

- 能力の運用方法（動作計画方法）

が異なる。

### 4.4.2 タスクテンプレート

タスクテンプレートは、環境情報に基づき、次の計画周期に“どのような作業（タスクの種類と質）”が“どの程度（ロボット台数）”必要かを判断し、割り当てに用いる情報を持つタスクインスタンスを生成するためのものである。タスクインスタンスは、1台のロボットが1～数計画周期で成すべき作業に対応する。例えば、4台のロボットである物体をハンドリングする、という作業は、物体ハンドリングタスクインスタンス4つ分に相当する。

タスクテンプレートには、タスクの異種性を反映するため、大きく分けて

- タスクインスタンスの生成・消去に関する項目
- 割り当て状態を評価するための優先度に関する項目

の2種類を記述する必要がある。Fig.4.3にタスクテンプレートを示し、各項目の内容を以下で説明する。

なお、現在の計画周期の終りまでに、すでに割り当てられているタスクインスタンスを完了するロボットを、“割り当て可能なロボット”と呼ぶ。その台数  $n_{freerob}$  は各計画周期で異なり、タスクインスタンスの生成において参照される。

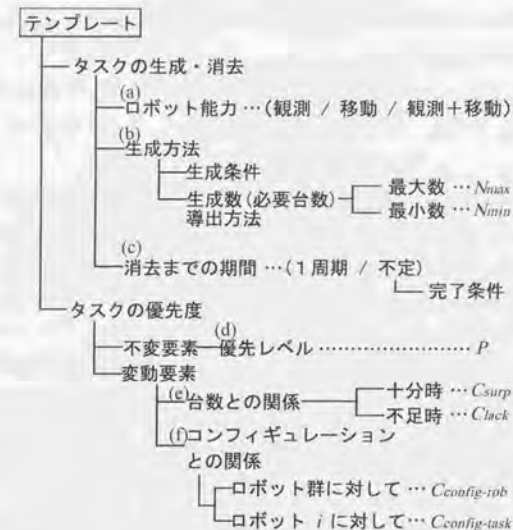


Fig. 4.3: Task Template

#### 4.4.2.1 タスクインスタンスの生成・消去に関する項目

まず、ある状況で必要とされる作業を抽出するためには、ロボットに対し何を要求するタスクインスタンスが、いつどのように生成され、またいつ完了したと見なされ消去され



るか、について規定する必要がある。そのため、(a)“ロボット能力”、(b)“タスクインスタンスの生成方法”、(c)“タスクインスタンスの消去方法”の3項目を定義する。

#### (a) “ロボット能力”

各タスクでは、ロボットがそれを実行するにあたって用いる能力が異なり、割り当てを決定する際に考慮する必要がある。本論文では、“移動”および“観測”をロボットの能力と考えた。したがって、ここではそのどちらかを用いるかを指定する。(ただしこの部分については他の能力を要素として考慮することも可能である。)

#### (b) “タスクインスタンスの生成方法”

どの情報がどのような条件を満たしたときにそのタスクが必要となるか(生成条件)を判断し、また、そのとき作業量はどれほどで何台のロボットを必要とするか(生成数)を見積もるための方法を定義する。

生成条件については、システムの保持する環境情報のうち用いる情報の種類と満たすべき条件を定義する。ただし、システムは環境情報として、ここに指定された情報を保持するものとする。

また生成数については、その作業量を達成するために確保すべき最低限の(あるいは適正な)台数  $N_{min}$  と、最大限の台数  $N_{max}$  の決定方法を定義する。これらの値を用い、生成過程では  $N_{max}$  個のタスクインスタンスが生成される。また、その計画周期における割り当て可能ロボットの台数から、そもそも  $N_{min}$  台より多くのロボットが確保可能かどうかという点が、タスクインスタンスの優先度決定の際に参照される。

#### (c) “タスクインスタンスの消去方法”

タスクインスタンスがロボットに割り当てられた後、消去可能とみなされるまでの期間と、その条件を定義する。期間には、“1計画周期”あるいは“不定”の場合が存在する。“1計画周期”と設定された場合には、1計画周期にあたる時間が経過することが条件となる。これに対し“不定”の場合には、完了条件として、システムの保持する環境情報のうち用いる情報の種類とそれが満たすべき条件を定義する。

#### 4.4.2.2 タスクインスタンスの優先度に関する項目

各計画周期においてロボットとタスクインスタンスとの最適な組合せを決定するためには、異種タスク間で守るべき実行順序を記述し、さらに、

- 各タスクに割り当てられるタスクインスタンスの数(Fig.4.4における大きな矢印)

- 同じタスク内での、ロボットとタスクインスタンスとの最適な組み合わせ(Fig.4.4における点線の矢印)

を決定するための割り当て状態の評価方法を記述する必要がある。

そこで、異種タスク間の実行順序に関して、タスクの本来持つ性質に基づいて設定され作業を通じて不変な要素である(d)“優先レベル”を定義する。また、割り当て状態の評価方法として、作業の進行に伴い変動する要素である(e)“台数との関係に基づく優先度”および(f)“コンフィギュレーションとの関係に基づく優先度”を定義する。

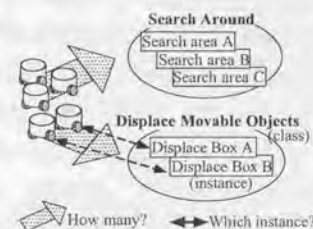


Fig. 4.4: Two aspects of Assignment

#### (d) “優先レベル: $P$ ”

優先レベルは、タスク本来の性質から定まり状況に依存しない不変要素であり、「あるタスクが他のタスクに先行して解決されるべきである」という異種タスク間の実行順序の相互関係に基づき設定する。優先レベルの最も高いものを1と設定し、以下高い順に2, 3, ...と設定する。したがって優先レベルの最も低いものが最も大きな数字をとることになる。

#### (e) “台数との関係に基づく優先度”: $C_{surp}$ , $C_{lack}$

タスクの優先度のうち、割り当て可能ロボット台数  $n_{freerob}$  と作業量の変化に伴い動的に変動する側面を評価する方法を定義する。タスクの種類に依存する側面であるため、同じ種類のタスクに属するタスクインスタンスでは同一の値をとることになる。

(b)に定義した  $N_{min}$  (最低限必要なロボット台数)と比較して、より多くのロボットの投入が好ましいかどうかは、 $n_{freerob}$  の値によって、そもそも  $N_{min}$  台以上のロボットを投入し得るかどうかによって変化する。例えば、最低でも3台揃わなければ全く実行できな

いタスクを考えると、その計画周期における割り当て可能ロボット台数が2台の場合、全てのロボットを割り当てたととしても結局作業はできず意味がない。

こうした関係を反映しロボット台数に応じた割り当て結果を可能とするため、 $C_{surp}$ 、 $C_{lack}$ の2つの定数を定義する。 $C_{surp}$ は割り当て可能台数が最低台数に対して十分大きい場合に、余剰分のロボットをどの程度必要とするかという度合を表す。 $C_{lack}$ は逆に不足している場合に、それでもなお割り当てを必要とする度合を表す。

(f) “コンフィギュレーションとの関係に基づく優先度”： $C_{config-task}$ 、 $C_{config-rob}$

タスクの優先度のうち、作業環境におけるロボット群のコンフィギュレーションの変化に伴い動的に変動する側面を評価するための方法を定義する。

$C_{config-task}$ は、それが実行されることのロボット群に対する意義であり、同じタスクに属するタスクインスタンスの間で質の違いを表現するための項である。

もう一つの $C_{config-rob}$ は、あるロボットがあるタスクインスタンスを実行する意義であり、

- そのタスクインスタンスのあるロボットが実行可能かどうか
- そのタスクインスタンスのあるロボットが実行するコスト

等を考慮する。これは、同じタスクに属するタスクインスタンスとロボットとの最適な組合せを決定するための項である。

## 4.5 優先度を用いたタスク割り当てアルゴリズム

テンプレートを用いて生成したタスクインスタンスを、それぞれの優先度を考慮してロボットに割り当てるアルゴリズムについて説明する。割り当てアルゴリズムでは各ロボットに必ず1つはタスクを割り当てるため、タスクインスタンス生成の過程で、最低でも割り当て可能ロボット台数分のタスクは必ず生成されている必要がある。

4.4.2項において、生成されたタスクインスタンスのあるロボットに割り当てる状態を評価する方法を定義した。したがって、タスクインスタンスの割り当て計画は、あるインスタンスをロボットに割り当てるかどうかで1または0の値をとる整数変数を用いると、いわゆる“割当問題”を拡張した形に定式化され、高速に解くことが可能である。

一方、4.4.2項で示したように、タスクによって必要なロボットの能力は異なり、必要とする能力によって、

- group 1 “移動”のみ（物体ハンドリングなど）
- group 2 “観測”のみ（周辺監視、ランドマーク計測など）
- group 3 “移動”および“観測”両方（障害物除去など）

の三つに分類する事ができる。したがって、1台のロボットは、互いに占有する能力が重複しない二つのタスク、すなわちgroup 1またはgroup 2に属するタスクからそれぞれ一つずつを同時に割り当て、実行することが可能であると言える。そこで、各ロボットの“観測”能力に対しては $S_{ij}$ 、“移動”能力に対しては $A_{ij}$ と別々の整数変数を用いる。これらの変数のとる値とその意味は以下の通りである。

$$A_{ij} = \begin{cases} 1 & \text{タスクインスタンス } i \text{ が} \\ & \text{ロボット } j \text{ の“移動”に割り当てられる場合} \\ 0 & \text{割り当てられない場合} \end{cases}$$

$$S_{ij} = \begin{cases} 1 & \text{タスクインスタンス } i \text{ が} \\ & \text{ロボット } j \text{ の“観測”に割り当てられる場合} \\ 0 & \text{割り当てられない場合} \end{cases}$$

なお、両能力を要するタスクの場合は、同じロボットがこれを行うという制約条件を付加し、必ず同じロボットに割り当てられるようにする（詳細は4.5.1項）。



割り当てアルゴリズムを以下に示す。

**Step 0** 初期問題を設定する。

優先レベル  $P = P_{lowest}$ 。

**Step 1** 問題が実行可能ならば、

優先レベル1から  $P$  に属するタスクに関する制約条件のもとで、

評価関数  $PI$  を最小化。

実行可能でなければ **Step 2** へ。

**Step 2** 優先レベル  $P$  に属するタスクに関する制約条件を緩和。

$P = P - 1$  として **Step 1** へ。

ここで  $P_{lowest}$  とは、実装タスクに対しそれぞれ準備された全てのタスクテンプレートに定義された優先レベルのうち最も低いものに相当する。ただし、優先レベルの数値は大きいほど低いものを表す。

また、初期問題とは、各タスクにおいて生成されたタスクインスタンスのうち、 $N_{min}$  個分のロボットの確保を制約条件とした問題を表す。制約条件の詳細は4.5.1項において説明する。 $PI$  はEq.(4.2)で表される評価関数である。

$$PI = \sum_i^{n_{instance}} \sum_j^{n_{freerob}} w_{ij} X_{ij} \quad (4.2)$$

$n_{instance}$ : タスクインスタンスの総数

$n_{freerob}$ : 割り当て可能ロボット台数

$X_{ij}$ :  $S_{ij}$  または  $A_{ij}$

この時、係数  $w_{ij}$  は、そのタスクインスタンス  $i$  をロボット  $j$  が行うという状態の評価値を意味し、テンプレートの優先度のうち変動要素によって、以下のように計算される。

$$w_{ij} = C_0 \cdot C_{num,i} \cdot C_{config-task,i} + C_1 \cdot C_{config-rob,j}$$

$$C_{num,i} = \begin{cases} C_{surp} & \text{when } n_{freerob} > N_{min} \\ C_{lack} & \text{otherwise} \end{cases}$$

ただし  $C_0, C_1$ : 重み係数 (本論文の以下のシミュレーションでは  $C_0 = 100.0, C_1 = 1.0$  としている)

ロボット台数に余剰分がある場合、制約条件の効果により必要最低の台数が確保されたのち、残りの部分については、変動要素のうち台数十分時の優先度  $C_{surp}$ 、あるいは、そのコンフィギュレーションにおける  $C_{config-task,i}$  の値が大ききものが、評価関数の効果により割り当てられる。

ロボット台数が十分でない場合、優先レベルの低い方から制約条件を緩和していくことで、優先レベルの高いタスクは制約条件の効果によって確実に割り当てられる。また優先レベルの低いタスクでも、変動要素のうち台数不足時の優先度  $C_{lack}$  やそのコンフィギュレーションにおける  $C_{config-task,i}$  の値が大ききものについては、評価関数の係数の効果により、なるべく多くのロボットに割り当てられる。

なお、 $n_{freerob} = N_{min}$  の時、この問題は一般に“割当問題”と呼ばれ、その解法については多くの研究がなされている ([伊理95] を参照)。ここでは分枝限定法を用いて解く ([大山93])。分枝限定法内部における線形計画計算については、本問題の場合、制約条件式の係数の関係で1回か2回のみ行われる。

#### 4.5.1 制約条件

定式化される制約条件には、大きくわけて以下の3種類の制約条件が存在する。

- (a) 各ロボットの“観測”能力および“移動”能力にタスクインスタンスを1つずつ必ず割り当てるための制約条件
- (b) 各タスクインスタンスに対し、ロボット1台分の“観測”能力あるいは“移動”能力を割り当てるとする制約条件
- (c) “観測”能力および“移動”能力を同時に1台のロボットが行うための制約条件

まず、(a)の各ロボットの“観測”能力および“移動”能力にタスクを1つずつ必ず割り当てるための制約条件は次のように記述される。

$$\sum_i^{n_{instance}} S_{ij} = 1 \quad j \in \{1, \dots, n_{freerob}\}$$

$$\sum_i^{n_{instance}} A_{ij} = 1 \quad j \in \{1, \dots, n_{freerob}\} \quad (4.3)$$

ただし、

$n_{instance}$ : “観測”能力を用いるタスクインスタンスの総数

$n_{a,instance}$ : “移動”能力を用いるタスクインスタンスの総数  
であり,  $n_{s,instance} + n_{a,instance} = n_{instance}$  という関係になっている。

次に, (b)に属する制約条件の記述方法を, あるタスクインスタンス $t_{ex}$ をどれかのロボット(1台分の“観測”能力および“移動”能力)に割り当てる, という例によって示す。必ず1台分に割り当てられる場合は以下のように等式条件として記述される。

$$\begin{aligned} \sum_j^{n_{freerob}} S_{lexj} &= 1 \\ \sum_j^{n_{freerob}} A_{lexj} &= 1 \end{aligned} \quad (4.4)$$

また, 割り当てる事が望ましく, 割り当てらばロボット1台分という場合には, 上記を緩和した不等式条件として記述される。したがって, 割り当てアルゴリズムにおいて緩和されるのは, 上記の部分となる。

$$\begin{aligned} \sum_j^{n_{freerob}} S_{lexj} &\leq 1 \\ \sum_j^{n_{freerob}} A_{lexj} &\leq 1 \end{aligned} \quad (4.5)$$

最後に, そのタスクインスタンスがロボットの能力のうち“観測”および“移動”の両者を必要とする場合, 同じロボットに割り当てられるよう(c)の制約条件式を次のように記述して加える。

$$S_{lexj} - A_{lexj} = 0 \quad j \in \{1, \dots, n_{freerob}\} \quad (4.6)$$

## 4.6 タスクの実装

本論文では搬送作業として,

「対象物は大型であり, 経路の大幅な変更はできるだけ避けることが好ましい」

「経路変更を防ぐために, 搬送経路の整備(環境への働きかけ)が可能」

という状況を想定する。そのために必要な最低限のタスクとして, “周辺監視”, “可動障害物除去”, “物体ハンドリング”の3種類を実装する。

このとき, 搬送経路上に存在する障害物の扱いに関する本論文の立場について言及しておく。障害物を発見した際にロボットがとる行動としては, “除去”および“迂回”の2種類が考えられるが, 上記より, 本論文では元の経路を最優先させる立場をとる。そのため, 除去可能な障害物に対しては除去を試み, 非可動な障害物に対してのみ搬送軌道の再計画を行ってこれを迂回する。非可動な障害物として, 下記を想定する。

- 壁など, 属性(重さ)に基づき, ロボットにより除去不可能と判断される固定障害物
- 属性(重さ)としては可動だが, 壁に接触するまで移動させるなど一通りの除去努力が施されたもの

各タスクに対するタスクテンプレート各項目の設定内容をTable 4.1にまとめる。

本節ではまず, 3つのタスク間の関係に関与するものとして, (a)“ロボット能力”の設定に基づく同時割り当て可能な関係と, (d)“優先レベル”および(e)“台数との関係に基づく優先度の決定方法”について説明する。さらに, タスクを実装するにあたっての前提条件を示し, タスクテンプレートの残りの項目および動作計画器について, タスク毎に分けて説明する。

### (a) “ロボット能力”の設定と同時割り当て可能な関係

各タスク実行に必要なロボット能力はTable 4.1に示す通りになっている。この時, “周辺監視タスク”および“物体ハンドリングタスク”は, それぞれ異なる能力を必要とするタスクであり, 1台のロボットがそれぞれから1つずつタスクインスタンスを選択し同時に実行することが可能な関係となっている。



Table 4.1: Contents of Implemented Tasks

タスク	周辺監視	障害物除去	ハンドリング
(a) ロボット能力	観測	観測+移動	移動
生成条件	対象物がゴール未到達	搬送経路内に可動障害物を検知	対象物がゴール未到達
(b) 生成数	$N_{max}$	障害物の個数	$n_{freerob}$
	$N_{min}$	条件を満たすセル集合を成す部分集合の数	2
(c) 消去までの期間	1 周期	不定	1 周期
完了条件	—	物体が搬送経路から出て壁に接触した時	—
(d) 優先レベル $P$	1	2	3
(e) 台数	(十分) $C_{surp}$	1.0	4.0
	(不足) $C_{lack}$	8.0	1.0
(f) コンフィギュレーション	$\alpha \times (\text{ロボット} i \text{ とタスク} j \text{ を生成する領域重心との距離})$	$\alpha \times (\text{ロボット} i \text{ とタスク} j \text{ を生成する障害物との距離})$	$\alpha \times (\text{ロボット} i \text{ とタスク} j \text{ を生成する把持位置との距離})$
$C_{config-task}$	領域の代表タイマー値	1.0	1.0

ただし  $\alpha = \begin{cases} 1: \text{タスクの発生した地点とロボットの間に他の障害物なし} \\ 10: \text{タスクの発生した地点とロボットの間に他の障害物あり} \end{cases}$

#### (d) “優先レベル”の決定方法

協調搬送システムのように、ロボット間の密な協調を中心とした協調作業システムでは、「タスクの実行によりシステムの安全を侵さない」順序で種々の作業を実行することが重要である。そのため優先レベルは、システム全体の安全維持に寄与するタスクほど高く、主目的作業ほど低く設定する必要がある (Fig.4.5)。

上記に基づいて実装タスクの優先レベルを考える。搬送作業において主目的に直接関係するのは“物体ハンドリング”である。しかし障害物に衝突せず安全に物体をハンドリングするためには、これより前に“可動障害物が除去”されている必要がある。さらにそのためには、その障害物が“周辺監視”により先に発見されている必要がある。したがって、システム全体の安全維持に最も寄与するタスクは“周辺監視”タスクであり、優先レベルは Table 4.1 に示す順序となる。



Fig. 4.5: Priority Level against the System Safety

#### (e) “ロボット台数との関係に基づく優先度”の決定方法

まず、各タスクに関する最低限必要な作業量を実行するのに必要なロボット台数 (タスクインスタンス数) に対し、現在の割り当て可能台数が十分である場合 ( $n_{freerob} > N_{min}$ ) に、余剰分のロボットの配分を決定する係数  $C_{surp}$  は次のように設定した。

周辺監視タスクについては、最低限の領域 (監視必要領域) を観測するロボットは確保されている状態であるため、余剰ロボットはそれほど必要としない。障害物除去に関しても、1 台のロボットで除去可能な物体の除去作業に余剰のロボットが投入されても、互いに動作の邪魔をするため、好ましくない。これらとは反対に、物体ハンドリングタスクについては、全体の台数が多ければ多いほど各ロボットに分配される荷重が小さくなり、好ましい状態と言える。したがって  $C_{surp}$  は、Table 4.1 に示す順に設定される。

次に、最低限必要な作業量に対し、現在の割り当て可能台数が不足である場合 ( $n_{freerob} < N_{min}$ ) のロボット台数の配分を決定する係数  $C_{lack}$  は次のように設定した。

周辺監視タスクに関して、監視必要領域のすべての観測が不可能な場合にも、なるべく多くのロボットを優先的に割り当て、少しでも広い領域に対して情報を更新する必要がある。また、障害物除去タスクに関しても同様に、複数の障害物が存在する場合、各障害物の除去は独立して実行可能であり、1 台でも多くのロボットが除去にあたれば、より早くその領域の通過が可能となる。物体ハンドリングタスクについては、余剰時とは異なり最低台数が集まらなければ実際に物体をハンドリングすることはできないため、他のタスクに対し割り当てる意義が低くなる。したがって  $C_{lack}$  は Table 4.1 に示す順に設定される。

### タスクの実装における前提条件

タスクの実装にあたって、下記の仮定をおく。

- 各ロボットの観測エリアは扇形にモデル化可能
- ロボットは観測エリア内に含まれる障害物の位置、動作（速度）、および属性（形状、重さ等）を認識可能

以後タスク毎に、上記で説明した(a)(d)(e)以外のテンプレートの内容および動作計画器について説明する。

### 周辺監視タスク

ロボット群が、与えられた搬送軌道上で障害物に衝突する事なく目標状態まで物体のハンドリングを行うためには、少なくとも対象物近辺の領域に関して、経路を塞ぐ障害物の有無を、見落としなく確実に認識する必要がある。

そこで周辺監視タスクとは、1計画周期内にロボット群による観測を必要とする領域を導出し、これを観測することであると捉える。タスクインスタンスは、その領域を1台のロボットで1計画周期内に観測可能な大きさに分割したものとなる。

ここでは、対象物近辺領域として、1計画周期経過後に到達する位置からどのような動作を指令されても、見落としが原因で障害物に衝突する事が無いよう、“監視対象領域”を定義する（Fig.4.6の薄灰色領域）。観測を行うセンサとしてカメラなど最短認識距離  $r_{min}$  が存在するものを用いる場合、見落としが起らないよう留意する必要がある。したがって図のように内側の領域を最短認識距離で決定する。

また、未知環境を移動しながら作業を行うロボット群にとっては、こうした環境に関する情報の信頼度は、その新しさに依存すると考えられる。そこで、領域の分割を簡単にし、それぞれの情報の質を表現するため、作業領域を矩形セルに区切り、各セルに、最後にその領域に関する観測情報を得てからの経過時間を表すタイマーを設置することにする（Fig.4.6）。なお、その領域に関して一度も観測を行っていない初期状態にはタイマー値として  $\infty$  を設定しておく。情報の古さを判断する目安として、タイマー値の閾値  $T_{old}$  を導入し、 $T_{old}$  を越えているセルの集合を特に“監視必要領域”と定義する（Fig.4.6における濃灰色の四角）。監視必要領域の観測が完了しているかどうかは、物体ハンドリングタスクにおける移動抑制の判断基準に用いられる（詳細は本節の物体ハンドリングタスクの説明の中で述べる）。

### タスクテンプレート

#### (b) “タスクインスタンスの生成方法”

生成条件は、対象物が目的地に到達していないことである。すなわち、作業を終了するまでは常に生成されることになる。

生成数について、最低でも監視必要領域の情報を更新する必要がある。そのため、監視必要領域に属するセルの集合を、1台のロボットで1計画周期内に監視可能な大きさに分割した部分集合の数が  $N_{min}$  となる（Fig.4.29）。分割の方法についての詳細は、本章末に



付録として記述した。

また割り当て計算において、各ロボットに割り当てうるタスクインスタンスを少なくとも1つ存在させる必要がある。そのため、監視対象領域のうち、現在のロボット配置で  $n_{freerob}$  台それぞれが観測可能な領域を考え、タスクインスタンスを生成する。したがって、タスクインスタンスの最大生成数  $N_{max}$  は  $N_{min}$  あるいは  $n_{freerob}$  のうちどちらか大きい方とする。

#### (e) “タスクインスタンスの消去方法”

指定された領域の全てを観測できたかどうかには関係なく、1計画周期で消去する。

ただし観測できなかった領域に関しては環境情報に反映させる。具体的には、観測したセルについてはタイマー値を0に更新し、それ以外の対象セルについては更新を行わないことで反映させる。

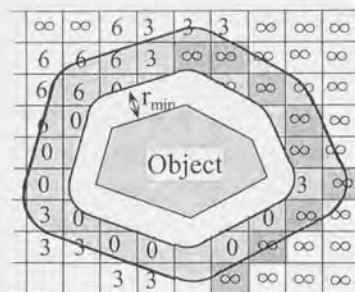


Fig. 4.6: Area to be searched

#### (f) “コンフィギュレーションとの関係に基づく優先度”

タスクインスタンス自体のロボット群に対する実行意義は、古い情報を更新する事になる監視必要領域の観測に関するものの方が、その他の領域を行うよりも高いと考えられる。したがって  $C_{config-task}$  は、その領域のタイマー値を反映したものと設定した。

また、対象物高さがロボットよりも高い場合、対象物を挟んで反対側の領域などはオクルージョンの発生により観測が不可能である。また、遠くの領域を観測するよりも、より近い領域の観測を行った方が精度が高いと考えられる。したがって、ロボットとタスクイ

ンスタンスとの組合せ関係の評価である  $C_{config-rob}$  には、表に示す導出式を設定した。

#### 動作計画器

上記で示したように、周辺監視タスクでは、タスク割り当て計画の段階で監視の必要な領域が導出されているため、動作計画器では、割り当ての決定したインスタンスに含まれるセルの重心位置を視野の中心に据えた場合を初期状態として、より多くのセル頂点を含む最適な視線方向を計画する。

## 障害物除去タスク

### タスクテンプレート

#### (b) “タスクインスタンスの生成方法”

タスクインスタンスの生成条件は、障害物が搬送軌道上に存在し、一定時間（シミュレーションでは10計画周期）以内にこれと衝突すると判断されることである。

ここでは各可動障害物が1台の移動ロボットで移動可能な場合のみを想定する。したがって1つの障害物に対し1つのタスクインスタンスが生成され、 $N_{min}$  も  $N_{max}$  も可動障害物の数に等しくなる。

#### (c) “タスクインスタンスの消去方法”

一旦ある障害物の除去に割り当てられたロボットはこれを完了するまで同じ物体を除去し続けるため、消去までの期間は不定となっており、そのロボットが「除去を完了した」と判断し通知するまで存在する。

完了条件は、軌道外まで移動されたと判断されることである。

#### (f) “コンフィギュレーションとの関係に基づく優先度”

あるロボット  $i$  が障害物  $j$  の除去を行う場合、ロボットの現在位置から障害物に辿り着くまでの距離が小さいほど早く除去作業を完了する事が可能であるため好ましい。すなわち、Fig.4.7における robot A の方が対象物を迂回して除去に向かう robot B よりも、このタスクを行う意義が大きい。また、よりロボット群に近い位置に存在する障害物の方が、遠くに位置するものよりも先に除去されることが望ましい。しかし、タスク割り当てのための見積りの時点で、各ロボットに対しこのように通るべき経路を算出するのは計算コストが高い。そのため  $C_{config-rob}$  としては、単純にロボットと障害物の代表点を結んだ線分が対象物と交差するか否かで、交差しない場合はその距離を直接、交差する場合はそれに重みをかけた値を用いることにする。

$C_{config-task}$  については、ゴールに到達するまでに衝突の予想されるものはすべて必要であるため、ここでは特に差をつくらず、1.0を設定しておく。

### 動作計画器

障害物を除去する経路の計画手法に関しては、[Faltings91] 他、種々の研究がなされているが ([Hwang92])、ここではタスク割り当て手法に焦点をおくため、動作計画としては非常に単純な戦略をとる。すなわち、まずは最も近い壁を除去目標接触壁として設定する。ロボットはまず除去対象物体に近寄り、障害物の壁と反対側に回り込んで、目標とした壁に到達するまで、搬送対象物の軌道に垂直にこれを押し続けるものとする。除去対象の障害物が複数ある場合は、壁付近の最終目的地が重ならないように経路を調整する (Fig.4.8)。

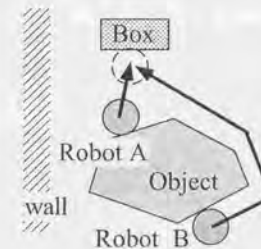


Fig. 4.7: Environmental Cost in the “Displace Movable Objects” Task

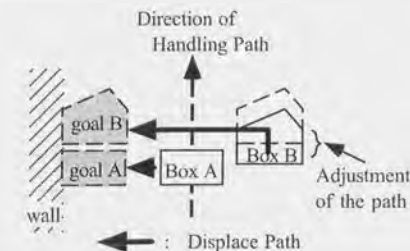


Fig. 4.8: Displacing path in the “Displace Movable Objects” Task



## 物体ハンドリングタスク

### タスクテンプレートの記述

#### (b) “タスクインスタンスの生成方法”

タスクインスタンスの生成条件は、対象物が目的地に到達していないことであり、到達するまでは常に生成しつづける。

またハンドリング動作に関して、ここでは

- 搬送物体が台車に載せられた状態でロボットはこれを押すまたは引くことでハンドリングを実現する
- 各ロボットの手先ではモーメントは発生させず平面内2次元の力のみを生成
- 各ロボットのトルク限界は十分高い

と仮定する。そのため、タスクインスタンス生成数のうち最低台数  $N_{\min} = 2$  となる。また、従事するロボット台数が多ければ多いほど各ロボットの対象物に及ぼすべきトルクが小さくなり望ましいため、 $N_{\max} = n_{\text{freerob}}$  である。

#### (c) “タスクインスタンスの消去方法”

1 計画周期毎に対象物が目的地に到達したかどうかを評価するため、各インスタンスは1 計画周期が経過した時点で完了するものとして扱う。

#### (f) “コンフィギュレーションとの関係に基づく優先度”

障害物の除去等で対象物を離れたロボットがおり、まだその他にも除去すべき障害物が残っている場合に、対象物付近に戻ってハンドリングを行うよりも、次の障害物の除去にうつった方が好ましいと考えられる。このような評価を反映するため、ロボット  $i$  とタスクインスタンス  $j$  との関係の評価方法である  $C$  は、現在のロボット  $i$  の位置と、タスクインスタンス  $j$  を生成する把持位置の距離を評価値として用いることにした。

$C_{\text{config-task}}$  については、タスクインスタンス間に特に差をつくらず、1.0 を設定しておく。

### 動作計画器

2 章で提案した協調ハンドリング動作計画器の概念設計に基づき、ここでも動作計画問題を、“対象物の動作計画問題” および “ロボットの動作計画問題” に分割し、シーケンシャ

ルに行う。以下、それぞれの計画器について説明する。

また4.10 節において、2 章で提案したロボットの持ち替え動作を伴う協調ハンドリング動作計画手法を本章での枠組と組み合わせた場合の協調搬送シミュレーションを行う。そのため、ロボット動作計画手法として、ここでは「持ち替えを伴うもの」、「伴わないもの」の2 種類を準備する。

#### 対象物の動作計画

原則として対象物は与えられた搬送軌道上を定速で移動するが、以下の条件を満たす場合には、その場で停止し待機するという移動の抑制を行う。

1. 他の“可動障害物の除去”中のロボットまたは障害物が、軌道上で動作している場合、
2. 1 周期前に監視必要と見なされた領域が監視されずに少しでも残った場合、

また、本節冒頭に示した障害物の扱いに関する立場に従い、非可動障害物を検知した場合はこれを迂回する動作を計画する。迂回方向については、対象物の重心と障害物位置を結んだ直線と垂直な方向とする。

#### ロボットの動作計画1—持ち替えを伴わない場合

この場合、対象物の予測位置に基づきロボットの予測把持位置を計算し、これをもとにロボット動作を計画する。2 章における把持ロボットの動作計画にあたる。

#### ロボットの動作計画2—持ち替えを伴う場合

2 章で提案した把持位置計画手法は、対象物周縁上に把持可能な位置が連続的に存在する場合に対するものであった。ここでは簡単のため、対象物周縁上に把持可能な位置が有限個、候補点として用意されている場合に、これらの中から最適な把持位置を選択する手法をとる。

ハンドリングに割り当てられるロボットには、

- 対象物周縁に位置し、実際にハンドリングを行う者
- 除去等で対象物から離れた場所に存在し、対象物周縁に戻ろうとする者

の2 種類が存在するが(Fig.4.9)、後者についてはタスクを割り当てられた際に暫定的に、どのロボットも把持していない把持候補点のうち最も近いものを把持位置として決めてお

く、ただしその把持位置は、実際にハンドリングする状態になるまで、最適化の対象とはみなさない。

対象物を実際にハンドリングを行うロボットの把持位置計画手法を以下のように行う。

**Step1** 各ロボットが次の周期の終りまでに速度限界内で到達可能な把持候補点を、それぞれ組合せの候補点として選択する。

**Step2** 物体を固定して把持するロボットを決め、それ以外の持ち替えロボットの把持位置をStep1で求めた把持候補点の中から組み合わせで選んで評価関数値を計算し、最適なものを持持位置として決定する。この時、対象物の目標速度が0の場合は最低1台を、0より大きい場合には最低2台のロボットを固定として計画する。

**Step3** 決定した把持位置から速度指令を計算する。

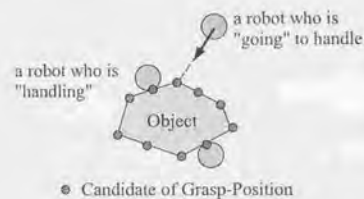


Fig. 4.9: Variation of Robots in Handling Task

## 4.7 未知静的環境における協調搬送シミュレーション

### 4.7.1 シミュレーション内容

提案した行動計画アーキテクチャの有効性を検証するため、未知環境における協調搬送シミュレーションを行う。

シミュレーションでは、環境内に存在する固定障害物を既知とし、固定障害物に衝突しない単純な直線状の搬送経路をあらかじめ与える。また、搬送経路上に未知障害物として、ロボットにより除去可能なもの（可動障害物）、除去不可能なもの（非可動障害物）の存在する、未知静的環境を想定する。

ここでは、ロボット群の構成台数および存在する未知障害物の種類を変化させた3種類のシミュレーションを行う。Table 4.2に各シミュレーション環境の特徴をまとめる。Simulation 1およびSimulation 2では、各種障害物の配置は同一であり、作業を行うロボット台数が異なる場合の比較となっている。

その他、各シミュレーションで共通の条件をTable 4.3に示す。

Table 4.2: Contents of Simulation

	the number of the robots	the number of the obstacles	
		removable	unremovable
Simulation 1	4	2	0
Simulation 2	2	2	0
Simulation 3	4	1	1



Table 4.3: Simulation Conditions (general)

the shape of the object	a $1.2 \times 0.6$ -[m] rectangle
the shape of the removable object	a $0.6 \times 0.4$ -[m] rectangle
the shape of the unremovable object	a $0.7 \times 0.8$ -[m] rectangle
the shape of the robots	circle with a $0.22$ -[m] radius
sensing area of the robots	a sector with a $2.0$ -[m] radius and a $20$ -degree interior angle
maximum speed while handling	$0.1$ [m/s]
maximum speed while running alone	$0.2$ [m/s]
division of work space for sensing	a $0.2 \times 0.2$ -[m] rectangle
planning cycle	$3.0$ [s]

## 4.7.2 シミュレーション結果（時間経過）

Simulation 1 から Simulation 3 の結果をそれぞれ Fig.4.11, Fig.4.13, Fig.4.15 に示す。また各シミュレーションに対し、全計画周期において生成されたタスクインスタンス数、各ロボットの実行したタスクインスタンスの種類、対象物のハンドリング状態（動作指令の種類）等を、Fig.4.12, Fig.4.14, Fig.4.16 に示す。

タスクの種別を示すため、図中では周辺監視タスクを S、障害物除去タスクを D、物体ハンドリングタスクを H とアルファベットを用いて表現してある。生成されたタスクインスタンス数については、最低限必要と判断されるもの（割り当てアルゴリズム上の初期問題において確保が制約条件と扱われるもの）と、それ以外のものに分けて示す。図中、タスク種別を表す S, D, H に円をついたものは、前者の制約条件として扱われるインスタンスであることを表している。

対象物の状態としては、ゴールに向かっての“直進(move)”，障害物の“回避(avoid)”，および“その場停止(stop)”の3種類が存在する。“その場停止”状態の場合には停止理由を図中に示した。停止理由としては、監視の不十分な領域の残留(U)，障害となる物体（固定，静止，ロボット）との衝突防止(C)，対象物から離れていたロボットが対象物ハンドリングに参加するための合流待ち(J)，およびロボット台数の不足(N)の4種類が存在する。

表記内容と計画周期（図中のtime）との関係は次のようになっている。タスクインスタンスについては、ある計画周期の動作を決定するために1つ前の計画周期において生成されたものを掲載している。また対象物の動作は、その周期のものを掲載している。

各図における扇形は各ロボットのセンシングエリアを、濃灰色の矩形は監視必要領域を表す。また図中の点列はロボットの移動した軌跡を表している。

各シミュレーション結果におけるロボット群の挙動を説明する。

## Simulation 1 (ロボット4台, 可動障害物2個)

搬送作業を開始し, しばらく周辺を監視しながら直進動作で物体をハンドリングした後 (Fig.4.11 (b)(c)), 33.0 秒において robot 1 および robot 2 の2台が対象物を離れ, “障害物除去” へと向かう (Fig.4.11 (d)(e)). これら2台が障害物を除去している間, 残りのロボットはハンドリング作業を続行している (Fig.4.11 (f)(g)). robot 1 が除去作業を終えて再び“ハンドリング”作業をすべく対象物近くへと戻すが (Fig.4.11 (h)), 除去作業が完了するまでは通過することができないためその場に止まって待機し, robot 2 が合流するのを待って (Fig.4.11 (i) (j)), 再びハンドリング作業を続行し (Fig.4.11 (k)), ゴールに到達している (Fig.4.11 (l)).

特徴的なタスク割り当ての切り替わり部分として, 27秒から33秒の割り当て結果の決まり方を説明する。

27秒および30秒においてすでに障害物除去のタスクインスタンスが生成されているが, 36秒になるまでの2周期に渡って割り当てられない。これは主に周辺監視タスクの  $C_{config-task}$  の効果である。すなわち, 監視必要領域の一部が未観測で残留したため, 必要最低限必要な作業として,  $C_{config-task}$  の値の大きい周辺監視タスクインスタンスが27秒および30秒で存在する。どちらもロボット台数分以上の観測能力を必要とすることからアルゴリズム上では優先レベル1まで制約条件が緩和され, 評価関数の重みである優先度の大きさのみで割り当てが決定する。そのため,  $C_{config-task}$  の効果で優先度の大きい周辺監視タスクインスタンスが選ばれている。

33秒で生成された最低限必要な作業は, ハンドリングに2台分の移動能力, 障害物除去に2台分の移動能力を要するのみとなっている。したがって, これに対する十分な台数(4台)のロボットが存在するため, これらのタスクに2台ずつのロボットが割り当てられ, その後, ハンドリングおよび障害物除去を並行して行う結果となっている。

## 移動抑制の効果

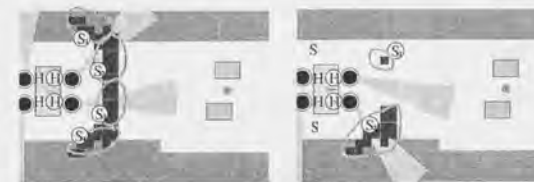
ある瞬間に監視必要領域を十分監視できなかった場合に, これを無視して物体のハンドリング(前進)を続けられ, 監視必要領域は広がる一方であり, これに伴ってタスクインスタンスも増加し続け, タスクの割り当て問題という観点から状況悪化を招く。また, 残留した未監視領域に障害物が存在した場合, これに衝突することになる。これに対し本論文では, 監視必要領域の監視の一部が未監視で残留した場合, 次の周期の物体ハンドリング動作計画では対象物を停止させる, という仕組みを実装した。そのため Fig.4.12 の対象物の動作の欄からも分かるように, 作業を通じて対象物が動いては止まる, という状態

が繰り返される。

例えば3.0秒では, それまで全く周辺に関する情報が得られていないため停止した状態で周辺の監視を行っている。この時, 全てのロボットに対し監視必要領域を観測するタスクインスタンスが割り当てられている (Fig.4.10(a))。しかし robot 3 および robot 4 は割り当てられたタスクインスタンス  $S_1$  および  $S_4$  の領域を実際には観測不可能であり, したがって全ての監視必要領域を覆うことができない。

これを受け, 6.0秒の周期では残留領域に対して, 新たに周辺監視タスクインスタンス2つが生成される (Fig.4.10(b))。この時, 4台のロボットが物体ハンドリングに割り当てられておりハンドリング動作の可能な最低台数は揃った状態であるが, 移動抑制の仕組みが働き, 動作指令としては  $0[m/s]$  すなわち停止指令が出されている。

3.0秒の周期で導出した監視必要領域の観測が3.0秒および6.0秒の2周期をかけ完了したことから, 9.0秒の周期では移動抑制が解除され, 対象物が前進をしている。同様の動作は, 126秒, 129秒, 138秒, 141秒他, 作業を通じて頻繁に見られた。その結果, 監視必要領域が増加してタスクが溢れる事はなく, また経路内に存在した障害物の見落としによる衝突等は起こらなかった。

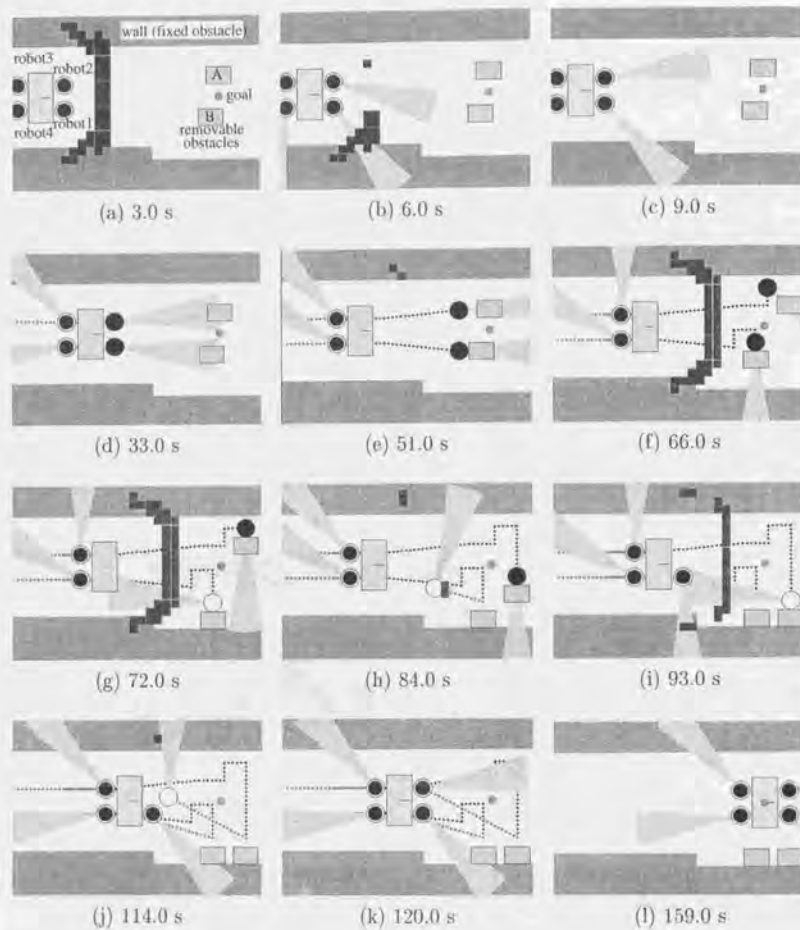


S: "Search Around" task  
D: "Displace Movable Objects" task  
H: "Handle an Object" task  
with  $\odot$  = instance that should be done at least

(1) generated for 3.0 s cycle (2) generated for 6.0 s cycle

Fig. 4.10: Task-generation Control by Handling Motion Planner





- : Displace Movable Object
- : Handle & Search (grasping the object)
- : Handle & Search (apart from the object)

Fig. 4.11: Result of Simulation 1 (4 robots, 2 removable obstacles)

time	the number of generated task instances				role of each robot				object motion	
[s]	S	D	H	I	rob1	rob2	rob3	rob4	status	why
0										
3	4	0	0	2	2	S	H	S	stop	U
6	3	1	0	2	2	S	H	S	stop	U
9	0	4	0	2	2	S	H	S	move	—
12	3	1	0	2	2	S	H	S	move	—
15	4	0	0	2	2	S	H	S	stop	U
18	2	2	0	2	2	S	H	S	stop	U
21	0	4	0	2	2	S	H	S	move	—
24	3	1	0	2	2	S	H	S	move	—
27	4	0	1	2	2	S	H	S	stop	U
30	2	2	2	2	2	S	H	S	stop	U
33	0	4	2	2	2	S	H	S	stop	C
36	0	2	0	2	0		S	H	stop	C
39	0	2	0	2	0		S	H	stop	C
42	0	2	0	2	0		S	H	stop	C
45	0	2	0	2	0		S	H	stop	C
48	0	2	0	2	0		S	H	stop	C
51	1	1	0	2	0		S	H	move	—
54	3	0	0	2	0		S	H	stop	U
57	3	0	0	2	0		S	H	stop	U
60	3	0	0	2	0		S	H	stop	U
63	3	0	0	2	0		S	H	stop	U
66	3	0	0	2	0		S	H	stop	U
69	3	0	0	2	0		S	H	stop	U
72	3	0	0	2	1	S	H	S	stop	U
75	3	0	0	2	1	S	H	S	stop	U
78	3	0	0	2	1	S	H	S	stop	U
81	3	0	0	2	1	S	H	S	stop	U
84	2	1	0	2	1	S	H	S	stop	U
87	1	2	0	2	1	S	H	S	stop	U

(time)	S	D	H	I	rob1	rob2	rob3	rob4	status	why
90	0	3	0	2	1	S	H	S	move	—
93	4	0	0	2	2	S	H	S	move	—
96	3	1	0	2	2	S	H	S	stop	U,C
99	1	3	0	2	2	S	H	S	stop	U,C
102	1	3	0	2	2	S	H	S	stop	U,C
105	0	4	0	2	2	S	H	S	stop	C
108	0	4	0	2	2	S	H	S	stop	C
111	0	4	0	2	2	S	H	S	stop	C
114	1	3	0	2	2	S	H	S	stop	C
117	1	3	0	2	2	S	H	S	stop	C
120	0	4	0	2	2	S	H	S	move	—
123	4	0	0	2	2	S	H	S	move	—
126	3	1	0	2	2	S	H	S	stop	U
129	1	3	0	2	2	S	H	S	stop	U
132	1	3	0	2	2	S	H	S	move	—
135	4	0	0	2	2	S	H	S	move	—
138	3	1	0	2	2	S	H	S	stop	U
141	1	3	0	2	2	S	H	S	stop	U
144	0	4	0	2	2	S	H	S	move	—
147	4	0	0	2	2	S	H	S	move	—
150	2	2	0	2	2	S	H	S	stop	U
153	0	4	0	2	2	S	H	S	move	—
156	2	2	0	2	2	S	H	S	move	—

reason to stop

- U: Unsearched area
- C: Collision with obstacles
- J: waiting for robots to Join
- N: lack of robot Number

S: "Search Around" task  
D: "Displace Movable Objects" task  
H: "Handle an Object" task  
h: when the assigned robot is apart from the object  
○: instance that should be done at least

Fig. 4.12: Generated Task-instances during Simulation 1

## Simulation 2 (ロボット 2 台、可動障害物 2 個)

搬送作業を開始し、しばらく周辺を監視しながら物体のハンドリングを行った後 (Fig.4.13 (b)(c)(d)(e)), 36.0 秒において robot 1 および robot 2 の両方が対象物を離れ“障害物除去”へと向かう (Fig.4.13 (f)(g)(h)(i)). その後, robot 1 が 72 秒に除去作業を終え, 再度ハンドリングに割り当てられ, 対象物を把持するために戻るが (Fig.4.13 (j)(k)(l)), 1 台だけでは実際に動くことができないためその場に待機している. 132 秒でもう 1 台の robot 2 が戻るため, 再びハンドリング動作を再開し (Fig.4.13 (m)(n)), ゴールに到達している (Fig.4.13 (o)).

特徴的なタスク割り当ての切り替わり部分として, 30 秒から 36 秒の割り当て結果の決まり方を説明する.

30 秒および 33 秒においてすでに障害物除去のタスクインスタンスが生成されているが, 36 秒になるまでの 2 周期に渡って割り当てられない. これは主に周辺監視タスクの  $C_{config-task}$  の効果である. すなわち, 監視必要領域の一部が未観測で残留したため, 必要最低限必要な作業として,  $C_{config-task}$  の値の大きい周辺監視タスクインスタンスが 30 秒および 33 秒で存在する. どちらもロボット台数以上の観測能力を必要とすることからアルゴリズム上では優先レベル 1 まで制約条件が緩和され, 評価関数の重みである優先度の大きさのみで割り当てが決定する. そのため,  $C_{config-task}$  の効果で優先度の大きい周辺監視タスクインスタンスが選ばれている.

次に 36 秒において最低限必要として生成されたタスクインスタンスは物体ハンドリングおよび障害物除去であり, どちらも 2 台分の移動能力を必要とする. しかし 2 台のロボットしか存在しないため, アルゴリズム上では優先レベル 2 まで制約条件が緩和され, 評価関数の重みである優先度の大きさのみで割り当てが決定することになる. ここでは, 優先度決定項のうち, 主に  $C_{num}$  の効果により割り当てが決定している. 物体ハンドリングタスクのインスタンスには不足時の  $C_{lack}(=1.0)$  が設定され値が低い. また周辺監視タスクでは  $C_{num}$  は十分時の  $C_{surp}(=1.0)$  が設定される. これに対し障害物除去タスクは  $C_{num}$  に不足時の  $C_{lack}(=8.0)$  が設定され値が高い. したがって除去タスクが選ばれ, 割り当てられる結果となっている.

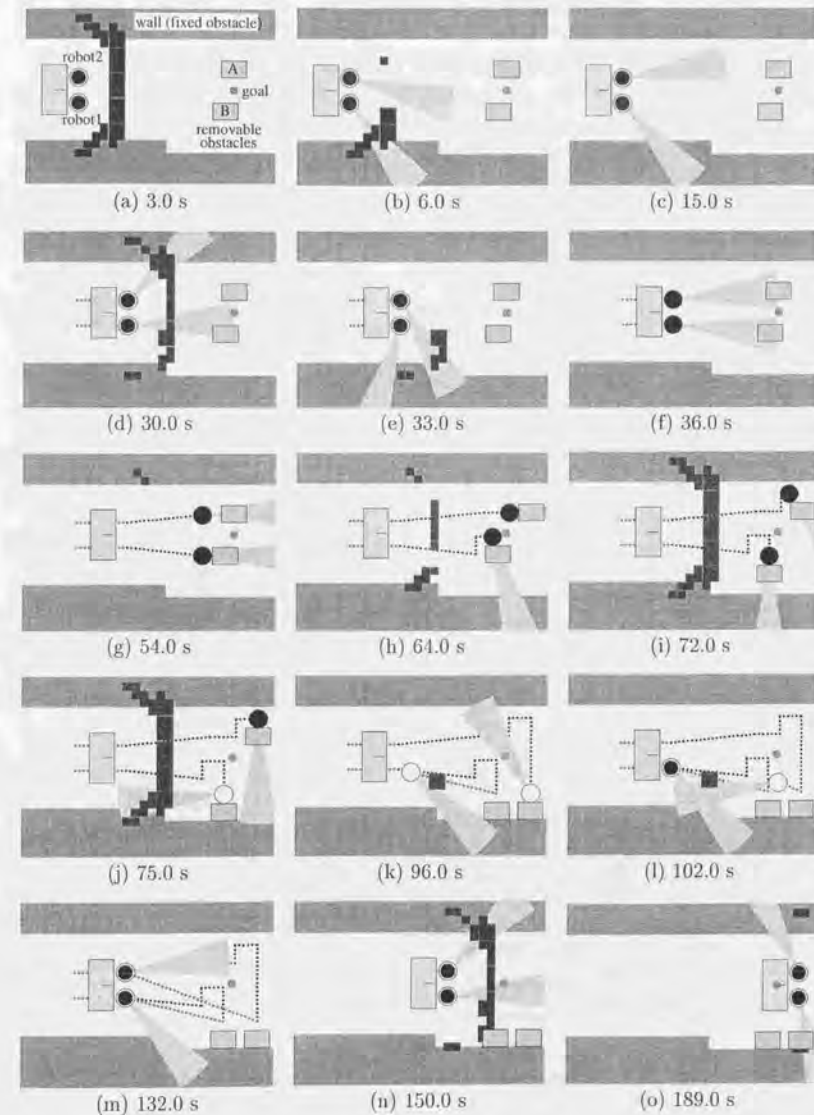


Fig. 4.13: Result of Simulation 2 (2 robots, 2 removable obstacles)



time	the number of generated task instances				role of each robot		object motion	
[s]	S	D	H		rob1	rob2	status	why
0								
3	4	0	0	2	2	2	stop	U
6	3	1	0	2	2	2	stop	U
9	1	3	0	2	2	2	stop	U
12	0	4	0	2	2	2	move	—
15	3	0	0	2	2	2	move	—
18	4	0	0	2	2	2	stop	U
21	2	2	0	2	2	2	stop	U
24	0	4	0	2	2	2	move	—
27	3	3	0	2	2	2	move	—
30	4	2	1	2	2	2	stop	U
33	2	3	2	2	2	2	stop	U
36	0	2	2	2	2	2	stop	N,C
39								
42								
45								
48								
51								
54								
57								
60								
63								
66								
69								
72								
75	4	1	0	1	0	0	stop	U,N
78	3	0	0	1	0	0	stop	U,N
81	3	0	0	1	0	0	stop	U,N
84	3	0	0	1	0	0	stop	U,N
87	2	0	0	1	0	0	stop	U,N

S: "Search Around" task  
D: "Displace Movable Objects" task  
H: "Handle an Object" task  
h: when the assigned robot is apart from the object  
○: instance that should be done at least

(time	S	D	H	rob1	rob2	status	why
90	1	0	0	1	0	stop	U,N
93	1	0	0	1	0	stop	U,N
96	1	1	0	2	0	stop	U,N
99	1	1	0	2	0	stop	U,N
102	1	1	0	2	0	stop	U,N
105	1	1	0	2	0	stop	U,N
108	1	1	0	2	0	stop	U,N
111	2	0	0	2	0	stop	C,N
114	1	1	0	2	0	stop	C,N
117	2	0	0	2	0	stop	C,N
120	2	0	0	2	0	stop	C,N
123	1	1	0	2	0	stop	C,N
126	2	0	0	2	0	stop	C,N
129	1	1	0	2	0	stop	C,N
132	0	0	0	2	0	move	—
135	3	0	0	2	0	move	—
138	4	0	0	2	0	stop	U
141	2	0	0	2	0	stop	U
144	0	2	0	2	0	move	—
147	3	0	0	2	0	move	—
150	4	0	0	2	0	stop	U
153	2	0	0	2	0	stop	U
156	0	2	0	2	0	move	—
159	3	0	0	2	0	move	—
162	4	0	0	2	0	stop	U
165	2	0	0	2	0	stop	U
168	0	2	0	2	0	move	—
171	3	0	0	2	0	move	—
174	4	0	0	2	0	stop	U
177	2	0	0	2	0	stop	U
180	0	2	0	2	0	move	—
183	2	0	0	2	0	move	—
186	2	0	0	2	0	move	—

reason to stop  
U: Unsearched area  
C: Collision with obstacles  
J: waiting for robots to join  
N: lack of robot Number

Fig. 4.14: Generated Task-instances during Simulation 2

## Simulation 3 (ロボット4台、非可動障害物1個、可動障害物1個)

0.0秒で搬送作業を開始すると、その場に停止したまま3周期をかけて監視必要領域の観測を行う。この間に非可動障害物の存在を発見している。その後、元の搬送経路のまま直進した場合、進行方向左前方の非可動障害物と衝突することから、12秒から30秒にかけてこれを回避する方向に物体のハンドリングを行う (Fig.4.15 (b))。30秒からは、ゴールに向かって直進してもこの非可動障害物に衝突しないと判断されたため、周辺を観測しながら通常のゴールに向かっての直進動作で物体のハンドリングを継続する (Fig.4.15 (c))。48秒の周期で可動障害物を発見し、54秒にrobot 2がこの除去に向かう (Fig.4.15 (d)(e)(f))。robot 2は87秒で除去動作を終えると、再び物体のハンドリングを割り当てられ (Fig.4.15 (g))。108秒で対象物を把持する状態に戻る (Fig.4.15 (i))。この間、残りのロボットは物体ハンドリング動作を継続しているが、robot 2にしか観測できない領域が未監視として残っているため、実際に対象物のハンドリングを行ったのは81秒の周期のみとなっている。robot 2が合流したのち、周辺監視およびハンドリングを行い (Fig.4.15 (j)(k))。168秒にゴールに到達している (Fig.4.15 (l))。

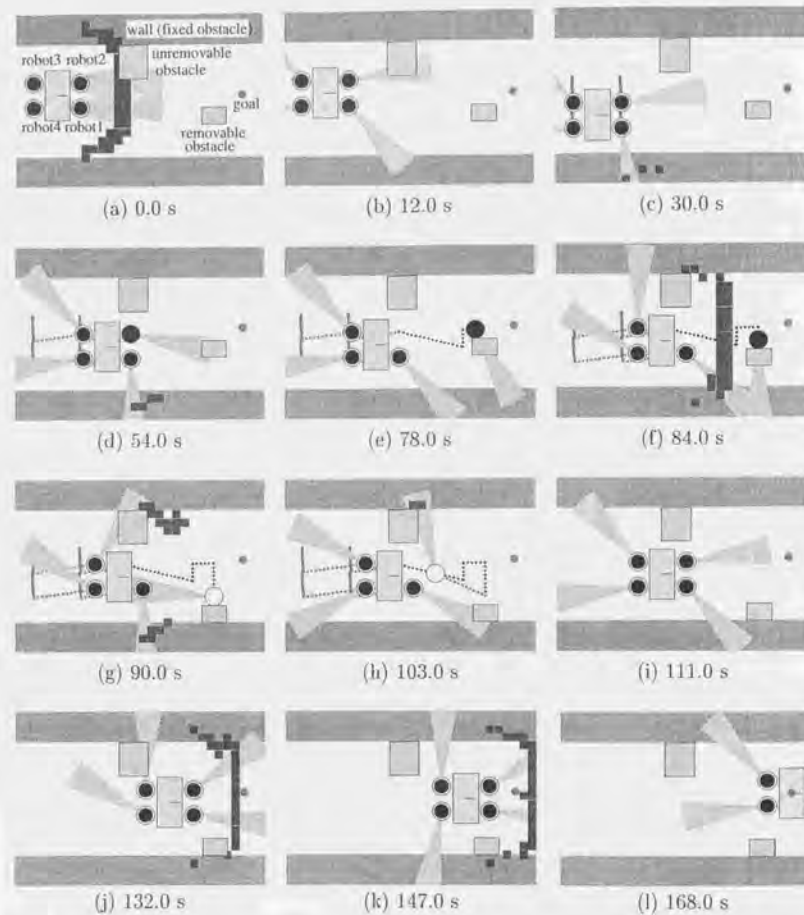


Fig. 4.15: Result of Simulation 3 (4 robots, Unremovable obstacle and removable obstacle)

time	the number of generated task instances	role of each robot	object motion
[s]	S D H	rob1 rob2 rob3 rob4	status why
0	⊙ S ⊙ ⊙ ⊙ ⊙		
3	4 0 0 2 2	⊙ H ⊙ H ⊙ H ⊙ H	stop U
6	4 0 0 2 2	⊙ H ⊙ H ⊙ H ⊙ H	stop U
9	1 3 0 2 2	S H ⊙ H S H S H	stop U
12	0 4 0 2 2	S H S H S H S H	avoid —
15	2 2 0 2 2	⊙ H ⊙ H S H S H	avoid —
18	1 3 0 2 2	⊙ H S H S H S H	avoid —
21	2 2 0 2 2	⊙ H ⊙ H S H S H	avoid —
24	1 3 0 2 2	⊙ H S H S H S H	avoid —
27	2 2 0 2 2	⊙ H ⊙ H S H S H	avoid —
30	1 3 0 2 2	⊙ H S H S H S H	move —
33	4 0 0 2 2	⊙ H ⊙ H ⊙ H ⊙ H	move —
36	3 1 0 2 2	⊙ H ⊙ H S H ⊙ H	stop U
39	2 2 0 2 2	⊙ H ⊙ H S H S H	stop U
42	0 4 0 2 2	S H S H S H S H	move —
45	5 0 0 2 2	⊙ H ⊙ H ⊙ H ⊙ H	move —
48	4 0 0 2 2	⊙ H ⊙ H ⊙ H ⊙ H	stop U
51	4 0 1 2 2	⊙ H ⊙ H ⊙ H ⊙ H	stop U
54	1 3 1 2 2	⊙ H ⊙ S H S H	stop U,C
57	0 3 0 2 1	S H S H S H	stop C
60	0 3 0 2 1	S H S H S H	stop C
63	0 3 0 2 1	S H S H S H	stop C
66	0 3 0 2 1	S H S H S H	stop C
69	1 2 0 2 1	⊙ H S H S H	stop C
72	1 2 0 2 1	⊙ H S H S H	stop C
75	0 3 0 2 1	S H S H S H	stop C
78	0 3 0 2 1	S H S H S H	stop C
81	1 2 0 2 1	⊙ H S H S H	move —
84	5 0 0 2 1	⊙ H ⊙ H ⊙ H	stop C
87	5 0 0 2 1	⊙ H ⊙ H ⊙ H	stop U,C

S: "Search Around" task  
 D: "Displace Movable Objects" task  
 H: "Handle an Object" task  
 —h: when the assigned robot is apart from the object  
 ⊙: instance that should be done at least

(time ⊙ S ⊙ ⊙ ⊙ ⊙ rob1 rob2 rob3 rob4 status why)  
 90 4 0 0 2 2 ⊙ H ⊙ H ⊙ H ⊙ H stop U,C  
 93 2 2 0 2 2 S H ⊙ H S H S H stop U,C  
 96 2 2 0 2 2 S H ⊙ H S H S H stop U,C  
 99 1 3 0 2 2 ⊙ H S H S H S H stop U,C  
 102 1 3 0 2 2 ⊙ H S H S H S H stop U,C  
 105 0 4 0 2 2 S H S H S H S H stop C  
 108 0 4 0 2 2 S H S H S H S H stop J  
 111 0 4 0 2 2 S H S H S H S H move —  
 114 4 0 0 2 2 ⊙ H ⊙ H ⊙ H ⊙ H move —  
 117 5 0 0 2 2 ⊙ H ⊙ H ⊙ H ⊙ H stop U  
 120 5 0 0 2 2 ⊙ H ⊙ H ⊙ H ⊙ H stop U  
 123 2 2 0 2 2 ⊙ H ⊙ H S H S H stop U  
 126 0 4 0 2 2 S H S H S H S H move —  
 129 4 0 0 2 2 ⊙ H ⊙ H ⊙ H ⊙ H move —  
 132 5 0 0 2 2 ⊙ H ⊙ H ⊙ H ⊙ H stop U  
 135 4 0 0 2 2 ⊙ H ⊙ H ⊙ H ⊙ H stop U  
 138 2 2 0 2 2 ⊙ H ⊙ H S H S H stop U  
 141 0 4 0 2 2 S H S H S H S H move —  
 144 4 0 0 2 2 ⊙ H ⊙ H ⊙ H ⊙ H move —  
 147 5 0 0 2 2 ⊙ H ⊙ H ⊙ H ⊙ H stop U  
 150 3 1 0 2 2 ⊙ H ⊙ H S H S H stop U  
 153 1 3 0 2 2 S H ⊙ H S H S H stop U  
 156 0 4 0 2 2 S H S H S H S H move —  
 159 2 2 0 2 2 ⊙ H ⊙ H S H S H move —  
 162 1 3 0 2 2 ⊙ H S H S H S H move —  
 165 2 2 0 2 2 ⊙ H ⊙ H S H S H move —

reason to stop  
 U: Unsearched area  
 C: Collision with obstacles  
 J: waiting for robots to join  
 N: lack of robot Number

Fig. 4.16: Generated Task-instances during Simulation 3



## 4.7.3 シミュレーション結果のまとめ

シミュレーション結果より次のことが分かった。

- 各シミュレーションでは、作業開始前に固定障害物以外の障害物の存在位置を知らされていない。そのため、作業の進行に伴い除去対象となる障害物が発見され、行すべきタスクの種類が作業途中で変化している。また、周辺監視タスクインスタンスは作業を通じて常に生成されているが、監視すべき領域が変化することから、それぞれの質は異なっている。しかし3つ全てのシミュレーションにおいて、ロボットによる環境の観測結果に基づき、静止障害物を回避あるいは除去することで、これらと衝突する事なくゴールへ到達している。すなわち、タスクが動的に変化する環境に対応可能であることが分かった。
- **Simulation 1** および **Simulation 2** では目的達成までの時間は異なるが、同じ環境に対し、異なる台数のロボット群の場合にも目的を達成する行動が計画されている。このことから、提案手法がロボット台数には無関係の構造であり、割り当て可能な台数に応じた計画が可能である事が示された。
- **Simulation 2** では、全般に渡り必要な作業に対しロボット台数が少ない事から、ロボット台数との関係に基づく項の影響する様子が顕著に表れた。
- 想定した未知静的環境の場合、対象物の移動抑制によりタスクの生成を抑制する仕組みが効果的に機能することが分かった。
- 計算時間は、SPARCStation20を用いて計算を行ったところ、ロボット台数が2台の場合、各ステップ平均で約0.07秒、4台の場合約0.11秒であった。

以上より、短時間の計画周期で区切って計画を繰り返す提案アーキテクチャにより、未知静的環境において搬送システムの主目的である対象物のゴール到達を実現する行動が実時間で計画可能であることが示された。

## 4.8 未知動的環境を想定した作業への適用

## 4.8.1 動的環境への提案手法の適用

前節までは、搬送対象物が大型で軌道変更を極力避けることが好ましい場合の搬送作業を想定し、未知静的環境における搬送システムとして、タスクの実装およびシミュレーションによる検証を行ってきた。上記の未知静的環境では、環境とロボット群との関係はロボット群の移動によってのみ変化する。したがって、ロボット群が移動を抑制（停止）する事で、環境との関係の変化が起らず、搬送作業の続行を妨げる障害物との衝突などの危険性を回避することが可能である。

ここで、より一般的な環境として、移動障害物の存在する未知動的環境を考える。環境内に移動障害物が存在することは、ロボット群の移動とは独立に環境との関係が変化することを意味している。すなわち、ロボット群が移動を抑制するだけでは危険性を回避する事が不可能であり、これを回避する行動をとる必要がある。

未知動的環境における作業システムとして、提案手法の枠組の適応性を確認するため、移動障害物の存在する環境における搬送作業シミュレーション (**Simulation 4**) を行う。移動物体については、未知静的環境における“非可動障害物”と同様に、発見するとこれを回避する行動をとるものとする。ただし移動速度は一定（速度  $0.05[m/s]$ ）とし、ロボットはその速度を測定可能とする。

**Simulation 4** (ロボット4台、移動障害物1個、可動障害物1個)

**Simulation 4**の様子をFig.4.17に示す。またFig.4.18に、シミュレーションの各計画周期における生成タスクインスタンス数、各ロボットへ割り当てられたタスクインスタンスの種類、計画された対象物の動作状態を示す。図の表記方法は4.7.2項と同様である。ただし、持ち替えるなど把持位置変更の状態が良く分かるように、ハンドリングタスクに割り当てられ、なおかつ持ち替えを行っている場合には、 $h_e$ と表記した。

0.0秒で搬送作業を開始すると、移動抑制の効果でその場に停止したまま3周期をかけて監視必要領域の観測を行う。21秒で移動障害物の存在を発見するが、監視必要領域の観測が残っているため1周期はその場に停止して観測を終え、24秒から移動障害物を回避する方向に物体のハンドリングを行う (Fig.4.17 (b))。その後48秒になるまでは、監視領域を毎周期毎に消化するため移動抑制により停止する事はなく、回避動作を続ける。

48秒からは、ゴールに向かって直進してもこの非可動障害物に衝突しないと判断されたため、周辺を観測しながら通常のゴールに向かっての直進動作で物体のハンドリングを継続する (Fig.4.17 (c)). 次に54秒の周期で可動障害物を発見し、60秒に robot 1 がこの除去に向かう (Fig.4.17 (d)). robot 1 は87秒で除去動作を終えると、再び物体のハンドリングを割り当てられ (Fig.4.17 (e)), 108秒で対象物を把持する状態に戻る (Fig.4.17 (g)). この間、残りのロボットは物体ハンドリング動作を継続しているが、robot 1 にしか観測できない領域が未監視として残っているため、実際に対象物のハンドリングを行ったのは105秒の周期のみとなっている。robot 1 が合流したのち、周辺監視およびハンドリングを行い (Fig.4.17 (h)), 168秒にゴールに到達している (Fig.4.17 (i)).

上記より、低速度で移動し運動予測の可能な障害物の存在する未知動的環境では、前節までと同様の提案アーキテクチャにより、障害物に衝突せず搬送システムの主目的である対象物のゴール到達を実現する行動が計画可能であることが示された。

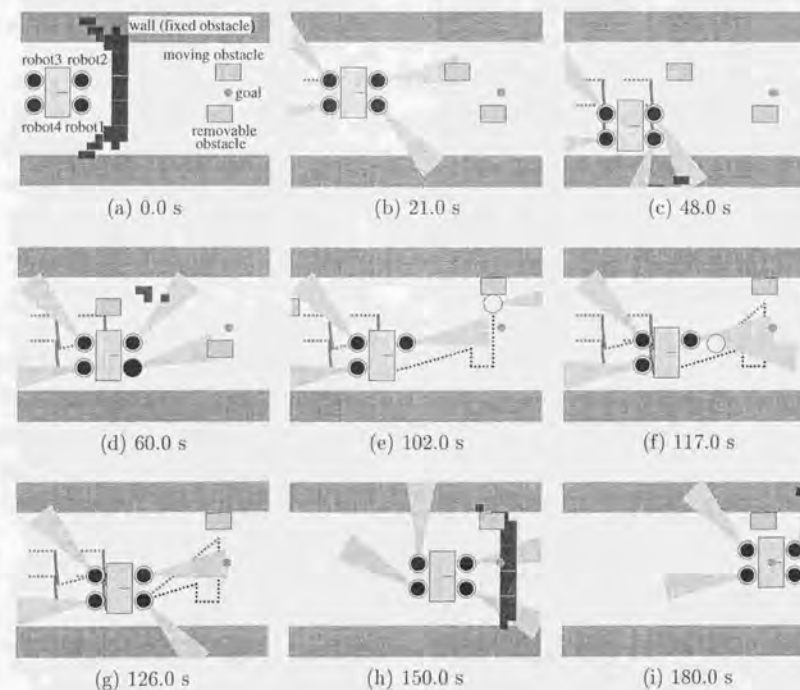


Fig. 4.17: Result of Simulation 4 (4 robots, a moving obstacle and a removable obstacle)



time	the number of generated task instances	role of each robot	object motion	
[s]	S D H	rob1 rob2 rob3 rob4	status why	
0	S S D H			
3	S 0 0 2	S H S H S H	stop U	
6	S 1 0 2	S H S H S H	stop U	
9	S 3 0 2	S H S H S H	stop U	
12	S 4 0 2	S H S H S H	move	
15	S 4 0 2	S H S H S H	move	
18	S 0 0 2	S H S H S H	stop U	
21	S 2 0 2	S H S H S H	stop U	
24	S 4 0 2	S H S H S H	avoid	Moving obstacle is found
27	S 1 3 0 2	S H S H S H	avoid	
30	S 2 0 2	S H S H S H	avoid	
33	S 1 3 0 2	S H S H S H	avoid	
36	S 2 0 2	S H S H S H	avoid	
39	S 2 0 2	S H S H S H	avoid	
42	S 2 0 2	S H S H S H	avoid	
45	S 1 3 0 2	S H S H S H	avoid	
48	S 2 0 2	S H S H S H	move	
51	S 4 0 2	S H S H S H	move	
54	S 0 0 2	S H S H S H	stop U	
57	S 4 0 1 2	S H S H S H	stop U	Removable obstacle is found
60	S 1 3 1 2	S H S H S H	stop U.C	
63	S 0 3 0 2	S H S H S H	stop C	
66	S 0 3 0 2	S H S H S H	stop C	
69	S 0 3 0 2	S H S H S H	stop C	
72	S 0 3 0 2	S H S H S H	stop C	
75	S 0 3 0 2	S H S H S H	stop C	
78	S 1 2 0 2	S H S H S H	stop C	
81	S 1 2 0 2	S H S H S H	stop C	
84	S 1 2 0 2	S H S H S H	stop C	
87	S 4 0 2	S H S H S H	stop U	

(time)	S	D	H	rob1	rob2	rob3	rob4	status	why
90	S	0	0	2	1			stop	U
93	S	0	0	2	1			stop	U
96	S	0	0	2	1			stop	U
99	S	1	0	2	1			stop	U
102	S	1	2	0	2	1		stop	U
105	S	0	4	0	2	1	S h	move	
108	S	4	0	0	2	1	S h	stop	C
111	S	3	1	0	2	1	S h	stop	U.C
114	S	1	3	0	2	1	S h	stop	U.C
117	S	0	4	0	2	1	S h	stop	C
120	S	0	4	0	2	1	S h	stop	C
123	S	0	4	0	2	1	S h	stop	C
126	S	1	3	0	2	1	S h	stop	C
129	S	0	4	0	2	2	S H	move	
132	S	0	0	0	2	2	S H	move	
135	S	4	0	0	2	2	S H	stop	U
138	S	4	0	0	2	2	S H	stop	U
141	S	1	3	0	2	2	S H	stop	U
144	S	1	3	0	2	2	S H	move	
147	S	5	0	0	2	2	S H	move	
150	S	4	0	0	2	2	S H	stop	U
153	S	4	0	0	2	2	S H	stop	U
156	S	1	3	0	2	2	S H	stop	U
159	S	0	4	0	2	2	S H	move	
162	S	5	0	0	2	2	S H	move	
165	S	4	0	0	2	2	S H	stop	U
168	S	4	0	0	2	2	S H	stop	U
171	S	1	3	0	2	2	S H	stop	U
174	S	0	4	0	2	2	S H	move	
177	S	2	2	0	2	2	S H	move	
180	S	2	2	0	2	2	S H	move	

S: "Search Around" task  
 D: "Displace Movable Objects" task  
 H: "Handle an Object" task  
 h: when the assigned robot is apart from the object  
 O: instance that should be done at least

reason to stop  
 U: Unsearched area  
 C: Collision with obstacles  
 J: waiting for robots to join  
 N: lack of robot Number

Fig. 4.18: Generated Task-instances during Simulation 4

## 4.8.2 動的環境への対応のための今後の課題

シミュレーションにより、低速度の移動障害物に対しては提案手法によりそのまま対応可能である事が分かったが、対応が不十分な場合もある。より一般的な状況に適用可能な搬送システムとするためには、次のような改良が課題となる。

## ● 移動抑制効果の起動条件の改良

例えば上記では、監視が不十分である場合にロボット群の移動を抑制しているが、非常に速度の高い物体が近付いてきた場合、移動の抑制が原因で回避しきれずに衝突する、という状況が考えられる。

これに対しては、移動障害物のように緊急の回避行動の必要が生じた場合、監視が確実に実行されている方向へ回避行動をとる、といった動作計画レベルでの対応により、対応可能であると考えられる。

## ● 状況に応じた回避行動の優先度の変更による改良

Fig.4.19のように、3台のロボットから成る協調搬送システムが、移動障害物1つと静止可動障害物2つを同時に発見する、という状況が考えられる。現在の実装方法では可動障害物の除去タスクの優先レベルの方が物体ハンドリングタスクよりも高いため、この場合、2台のロボットが除去タスクに向かうことになる。その結果、残りのロボット3だけでは回避動作を実現できず、やはり衝突を招いてしまう。

このような状況に対応するためには、移動障害物のように危険度の高い障害物を“回避する”という状態を、静止可動障害物の除去よりも優先度の高い状態として扱う必要がある。改良方法として、以下のような方法が考えられる。

## (1) コンフィギュレーションとの関係に基づく優先度の定義の改良

前節までの実装方法では、物体ハンドリングタスクにおけるコンフィギュレーションとの関係に基づく優先度は1.0という固定成分としていた。この部分を、例えば

$$C_{\text{config-task}} = \begin{cases} \left( \frac{r_{\text{max}}}{d_{\text{movingobjs}}} \right)^2 & \text{移動障害物が存在する場合} \\ 1.0 & \text{存在しない場合} \end{cases}$$

ただし、

$r_{\text{max}}$ : 回避行動を開始すべき距離（観測可能な最大距離など）

$d_{movingobj}$ : 対象物から移動障害物までの最短距離

とすることで、移動障害物が近付くにしたがって危険度が増し優先度が高くなる、という構造となる。この値が障害物除去タスクの場合のものよりも大きくなれば、同時に静止障害物を発見している場合にも回避行動をとるためのハンドリング作業が優先される結果になると考えられる。

## (2) 優先レベルの変更

“物体ハンドリング”という作業は、その動作の意図から、“目標位置を達成するためのハンドリング作業”という側面と“衝突回避のためのハンドリング作業”という側面から成ると捉える事ができる。そこで衝突回避のためのハンドリングタスクを、静止障害物除去タスクよりも一つ高い優先レベルに設定することで、移動障害物が接近した場合に、回避行動を確実に実行することが可能になると考えられる。

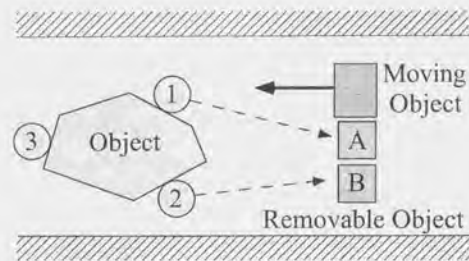


Fig. 4.19: Problem caused by Moving Obstacle

## 4.9 提案手法の適用範囲の限界について

提案した行動計画アーキテクチャでは対応不可能な場合について説明する。

### 動作計画手法に起因する限界

本論文では除去動作の実装にあたり、タスク割り当て手法に焦点を当てるため、動作計画器としては、搬送経路に垂直方向に移動させるという、非常に単純な戦略を採用している。複数の障害物が存在する場合に、他の障害物との関係は、除去完了時の目標位置における干渉についてのみ考慮している。

これに対し、Fig.4.20のような状況が考えられる。これは、始めに発見した可動障害物AおよびBの除去経路に、さらに障害物Cが存在する状況である。この場合、除去経路を変更する必要があるが、実装した動作計画器では除去作業が始まった後の経路の変更までは対応していないため、この状況では衝突を起こすことになる。

ここであげた限界はこのような動作計画器の不完全さに起因するものである。したがって、動作計画器を、“除去目標地までの経路の確認”、“経路が空いていない場合には別の除去目標地の再計画”の二点を行うものに改善することにより対処可能と考えられる。

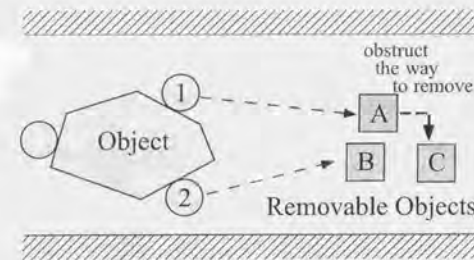


Fig. 4.20: Limit of the Proposed Architecture – Case 1



### タスク割り当て手法の枠組に起因する限界

最後に、本論文のタスク割り当てアーキテクチャでは、1あるいは数計画周期という比較的短い期間に1台のロボットが実行可能な形態をタスクの単位として捉えた、シミュレーションにより、障害物除去のように数計画周期に渡るタスクの扱いも可能であることが示されている。

これに対し、Fig.4.21のような状況が考えられる。これは、3台のロボットから成る協調搬送システムが、停止している可動障害物2つをまず発見し、除去タスクに割り当てられた後に移動障害物を発見した、という状況である。残りのロボット3だけでは回避動作を実現できず、やはり衝突を招いてしまう。

この例は、数計画周期に渡るタスクが多く存在し、それらの実行手順を考えることが重要な場合への適用の難しさを表している。すなわち、提案手法では1計画周期毎に割り当てを繰り返し行うため、その計画周期における割り当て結果の妥当性は保証されるが、数計画周期に渡る場合には保証されない。したがって、基本的にはタスクを1計画周期で分断可能であることが前提であり、上記の問題はこのタスク割り当て手法の枠組に起因する限界と言える。

この問題に対応するためには、一旦、数周期に渡るタスクインスタンスの実行を開始したロボットへの割り当てを、完了を待たずに強制的に解除し中断という状態で保存する機構が必要となる。

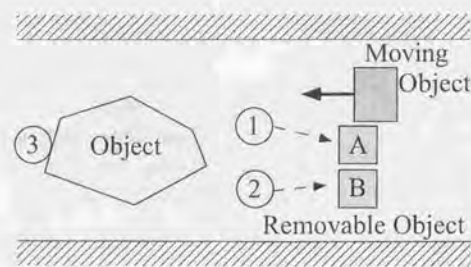


Fig. 4.21: Limit of the Proposed Architecture – Case 2

ここで、本論文では未知環境として、

「ロボットが走行することを認知している作業者が数名、また作業を妨害する物体が数

個存在する環境」

を想定した。この場合、除去すべき物体はロボット台数をはるかに越えては存在しないため、除去順序などの作業手順はある程度単純であると考えられる。また移動障害物としては、作業者自身あるいは作業者の運転するフォークリフト等が考えられる。しかしこれに関しても、ロボットが走行することを認知しているため、故意にロボット群に高速度で突入することは無いと考えられる。また、比較的低速度で動作方向の急激な変更は少なく、その動作は推測可能と考えられる。

以上より、

- 回避などの動作予測・対応が可能な動作で移動する障害物が存在
- 移動可能な障害物が散在

する未知環境において、目標状態を達成する協調行動の計画に、提案アーキテクチャは有効であると言える。

## 4.10 持ち替えを伴う協調搬送シミュレーション

2章で提案した持ち替え動作を伴う協調ハンドリング動作計画手法と、本章で提案した行動計画アーキテクチャとを組み合わせた統合システムが搬送システムとして機能することを示すため、未知障害物存在下での協調搬送シミュレーションを行う。

## 4.10.1 シミュレーション設定

ハンドリングタスクの割り当てに関する実装面については4.7節と共通とする、ただし、ロボットの動作計画手法については2章の計画をそのままは適用せず、4.6節で説明した持ち替えを伴う動作計画手法を用いる。

ロボットおよび対象物の形状等に関するシミュレーション条件をTable 4.4に、持ち替えに関する評価関数の重み係数をTable 4.5に示す。またFig.4.22に、ロボット・対象物・可動障害物の初期位置、対象物のゴール位置、およびロボットが把持可能な位置（全14箇所）を示す。

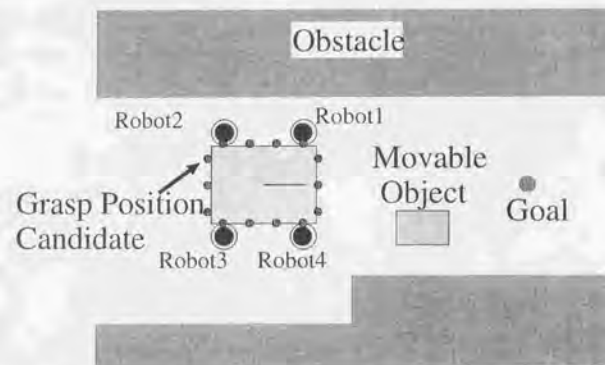


Fig. 4.22: Initial State of the Transport simulation

Table 4.4: Simulation Conditions

the shape of the object	a $1.8 \times 0.9$ -[m] rectangle
the number of the robots	4
the shape of the robots	a circle with a 0.15-[m] radius
sensing area of the robots	a sector with a 2.5-[m] radius and a 45-degree interior angle
maximum speed while handling	0.1 [m/s]
maximum speed while running alone	0.2 [m/s]
division of work space for sensing	a $0.2 \times 0.2$ -[m] rectangle
planning cycle	3.0 [s]

Table 4.5: Coefficients of Performance Index for Regrasping

Weight Coefficient		
$\alpha$ (regrasping)	$\beta$	$\gamma$
1.0 (o)	5.0	10.0

## 4.10.2 シミュレーション結果（時間経過）

Fig.4.23に、シミュレーション結果として、時間経過に沿ったロボットおよび対象物の動作を示す。またこの時のハンドリング動作計画の中で把持位置を決定する各評価項目値の遷移をFig.4.25 Fig.4.26 Fig.4.27に示す。

作業を開始した後(Fig.4.23 (a)) ロボット群は可動障害物を発見し、robot 4が“障害物除去”タスクに割り当てられて対象物を離れ、障害物へと向かう(Fig.4.23 (b))。この時、評価関数のうち第2項および第3項( $d_2$ および $d_3$ )の項の影響で、robot 1が持ち替え把持位置を変更している(Fig.4.23 (c))。robot 4が障害物を除去している間、残り3台のロボットはハンドリング作業を続行している(Fig.4.23 (d)(e)(f))。ただし除去作業が完了するまでは通過することはできないため、残りの3台は0.6[m]進んだところで停



止待機し, robot 4 が除去作業を終えて再び“ハンドリング”作業に合流するのを待つ (Fig.4.23 (g)). 再びハンドリング作業を続行する. 対象物を離れたロボットの把持位置は合流するまでは最適化されないため, 合流直後の Fig.4.23 (g) では robot 1 と robot 4 が非常に近い位置を把持し評価項目の第2項が低くなっている. その後 robot 4 を含めた把持位置の最適化が行われ, 初期状態と同じ配置に戻る (Fig.4.23 (g) から (h)). 除去し終えた可動障害物の横を通過する際に, あらかじめ与えられた軌道を守った場合, 初期の配置のままではロボットの大きさが邪魔となり通過ができない. しかしここでは評価項目の第1項 (障害物回避) の影響に基づく持ち替えが行われ, 対象物の軌道を変更すること無くこの箇所を通過することが可能であった. Fig.4.23 (i) から (j) では, robot 2 および robot 4 が, Fig.4.23 (k) から (l) では, robot 1 および robot 3 が, Fig.4.23 (l) から (m) では, 再び robot 2 および robot 4 が, 持ち替えを行っている.

通過後, 評価項目の第1項の影響は減少し, Fig.4.23 (n) から (o) では, robot 1 および robot 3 が再びロボット群は初期の配置に戻り, 対象物はゴールに到達している.

以上により, 2章で提案した持ち替えを伴うハンドリング計画を本章で提案したタスク割り当てを伴う行動計画アーキテクチャと組み合わせることで, 未知環境における柔軟な行動計画がなされることが分かった.

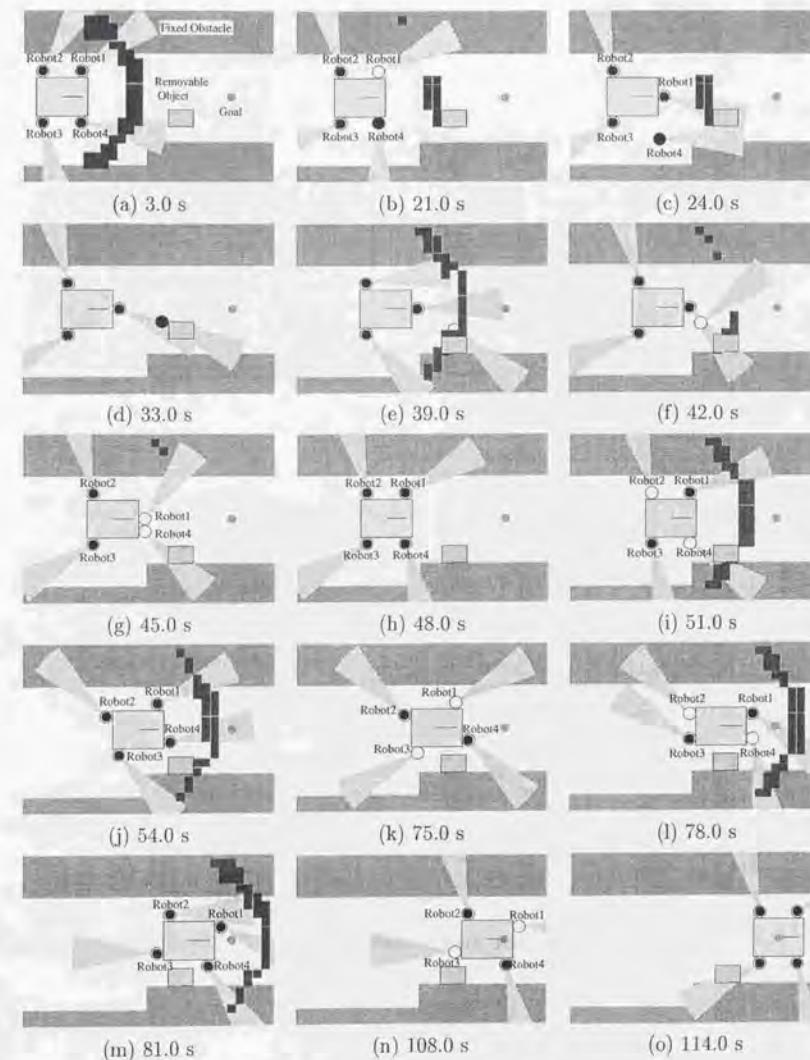


Fig. 4.23: Simulation Results-with Regrasping

time	the number of generated task instances				role of each robot				object motion	
[s]	S	D	H		rob1	rob2	rob3	rob4	status	why
0										
3	4	0	0	2	2	S	H	S	H	stop U
6	2	2	0	2	2	S	H	S	H	stop U
9	1	3	0	2	2	S	H	S	H	stop U
12	0	4	0	2	2	S	H	S	H	move —
15	3	1	0	2	2	S	H	S	H	move —
18	3	1	1	2	2	S	H	S	H	stop U
21	2	2	1	2	2	S	H	S	H	stop U,C
24	1	2	0	2	2	S	H	S	H	stop U,C
27	1	2	0	2	2	S	H	S	H	stop U,C
30	0	3	0	2	2	S	H	S	H	stop C
33	0	3	0	2	2	S	H	S	H	move —
36	3	0	0	2	1	S	H	S	H	move —
39	3	1	0	2	2	S	H	S	h	stop U,C
42	3	1	0	2	2	S	H	S	h	stop U,J
45	1	3	0	2	2	S	hr	S	H	stop U
48	0	4	0	2	2	S	H	S	H	move —
51	3	1	0	2	2	S	hr	S	H	move —
54	3	1	0	2	2	S	H	S	H	stop U
57	1	3	0	2	2	S	H	S	H	stop U
60	0	4	0	2	2	S	H	S	H	move —
63	3	1	0	2	2	S	H	S	H	move —
66	3	1	0	2	2	S	H	S	H	stop U
69	2	2	0	2	2	S	H	S	H	stop U
72	1	3	0	2	2	S	H	S	H	stop U
75	0	4	0	2	2	S	hr	S	H	move —
78	3	1	0	2	2	S	h	S	H	move —
81	3	1	0	2	2	S	H	S	H	stop U
84	2	2	0	2	2	S	H	S	H	stop U
87	1	3	0	2	2	S	H	S	H	stop U

Obstacle is found

reason to stop

- U: Unsearched area
- C: Collision with obstacles
- J: waiting for robots to join
- N: lack of robot Number

S: "Search Around" task  
D: "Displace Movable Objects" task  
H: "Handle an Object" task  
h: when the assigned robot is apart from the object  
hr: when the robot is regrasping the object  
○: instance that should be done at least

Fig. 4.24: Generated Task-instances during the Simulation

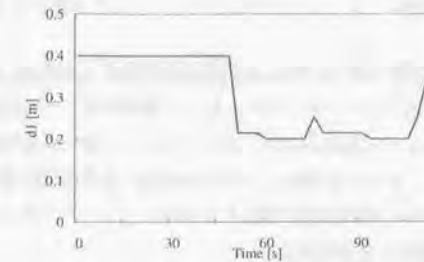


Fig. 4.25: Transition of  $d_1$

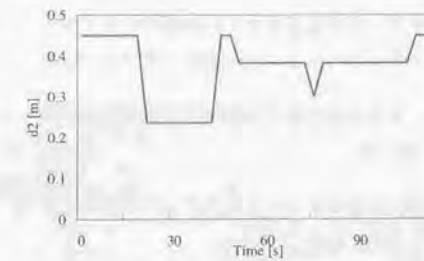


Fig. 4.26: Transition of  $d_2$

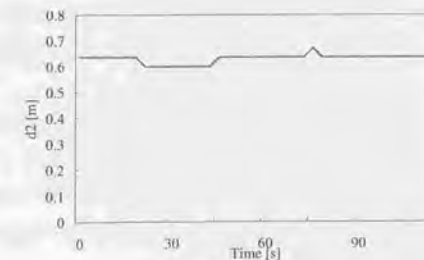


Fig. 4.27: Transition of  $d_3$



## 4.11 おわりに

本章では、異種作業割り付け型の協調搬送作業を可能とするため、情報層に関する課題解決にあたる“タスク割り当て手法”を中心とした行動計画アーキテクチャを構築した。

このタスクには次のような特徴がある。まず、ロボット群による解決が必要なタスクは作業進行に伴い変化し、ロボット群によってその瞬間に実行可能な量を越えて存在する場合がある。また、ロボットが環境内を移動するがゆえにタスクが生じるため、ロボットの移動に伴い実時間で解決する必要がある。

これに対し本章では、

- 短期間に1台のロボットにより実行可能な作業をタスクの単位と捉えること（タスクインスタンス）
- あるタスクインスタンスをあるロボットが実行する状態を、ロボットの台数との関係およびコンフィギュレーションとの関係に基づき評価すること（優先度）

を考えた。これにより、タスク割り当て計画問題が線形計画問題の一つである“割当問題”として定式化される。そこで、

- テンプレートを用いた必要タスクインスタンスの生成
- 優先度に基づくタスク割り当て計画
- タスク毎の協調動作計画

を一定計画周期で繰り返し実行するアーキテクチャを提案した。これは、分枝限定法を用いたタスク割り当て計画手法と、短期間の推定に基づく動作計画器と組合せることで実時間計画を可能とするものである。また、1計画周期毎にタスクインスタンスの生成および優先度評価を繰り返し行うことで、動的に変化する環境において実行順序など相互関係の深いタスクの割り当てを可能とするものである。

提案手法の有効性を検証するため、未知障害物の存在する環境での協調搬送シミュレーションを行った。その結果、回避などの対応が可能な動作で移動する障害物や、移動可能な障害物が散在する環境において、除去順序などの作業手順がある程度単純な場合に、提案アーキテクチャは有効であり、目標状態を達成する協調行動の計画が可能であることが分かった。

また、2章で提案した持ち替え動作を伴う協調ハンドリング動作計画器を組み込んだシミュレーションを行った。その結果、2章で提案した動作計画器が、本章で提案した異種

作業割り付け型の協調搬送システムのための動作計画器として適用可能であることが確認できた。

以上より、異種作業割り付け型の協調搬送システムのための行動計画アーキテクチャを構築することができた。

## 付録 — 周辺監視タスクにおける領域分割方法について

実装タスクの内、周辺監視タスクのインスタンスの生成方法について、補足説明を行う。

監視必要領域を表す濃灰色セル集合を部分集合に分割するにあたり、セルが直接隣接してつながった部分集合以外を別の部分集合とみなすと、タスクインスタンスの数が無駄に多くなる恐れがある。そこで、直接隣接しなくとも比較的近くに存在する領域をひとまとめにして扱うための処理を行うことにする。これにより、無駄に多い部分集合生成が抑制される。

まず、対象物重心を中心としてその近辺の領域を放射状に分割する。そして扇形の各放射領域に対し、含まれるセルの数に関するヒストグラムを作成する (Fig.4.29 (a))。

次に近い距離にある放射領域のつながりを考慮するための処理を行う。まず作成されたヒストグラムを、含むセルが1つ以上存在するかしないかで、1または0の数列に変換する。この数列に対し“膨張”と“収縮”という画像処理で用いる手法を適用し領域のつながりを再構成する(Fig.4.29 (b))。これにより、例えばひとつの放射領域を挟んで隣合うセルの監視をひとまとめに扱うことが可能となる。

その際、分割された領域の大きさを、ロボット1台が1計画周期中に監視可能な広さに収まるようにしておく。これは、一度の観測操作（情報の取得と処理）に必要な時間から1計画周期に何度この処理が可能かによって求まる（Fig.4.28）。

図の例の場合、ロボットは1計画周期にFig.4.29 (a)における放射領域2つ分を監視可能とみなされたため、4放射領域が連続したところが2つに分割され、全部で3つのタスクインスタンスが生成される (Fig.4.29 (c)).

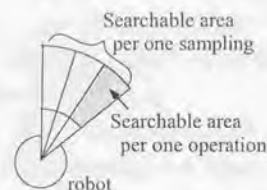
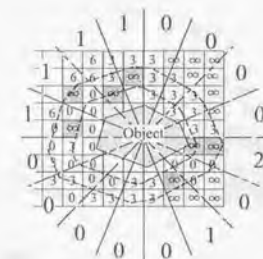


Fig. 4.28: Searchable area during one cycle



(a) Histogram Generation

0000111100000102 ← convert into 1 or 0  
0000111100000101 ← (one or nothing)  
100111110001111 ← extraction  
0000111100000111 ← contraction

(b) Histogram Contraction and Extraction



(c) Adjusted to Robots Searchable Area

Fig. 4.29: How to generate Instances in the “Search Around” Task



## Chapter 5

### 実ロボットによる異種作業割り付け型協調 搬送作業の実現

---

5.1 はじめに	158
5.2 実験システムの全体の構成	159
5.3 各ロボットの構成	161
5.4 物体ハンドリング部	163
5.5 環境認識部	165
5.6 未知環境における協調搬送実験	171
5.7 実ロボットによる環境認識過程についての考察	178
5.8 おわりに	180

---

## 5.1 はじめに

本章で提案した異種作業割り付け型協調搬送システムが実ロボットを用いて実現可能である事を示すため、4章で提案した行動計画アーキテクチャを実装した実ロボットシステムを構築して協調搬送実験を行う。

ここでは以下の方針に従ってシステムを構築する。

- 環境情報の融合、タスク割り当て、タスク毎の動作計画はホストコンピュータに行わせる。
- 周辺環境情報の取得、実際の速度指令導出、および駆動等の制御部分は各ロボットに行わせる。

本章では、まずこの方針に基づいて構成した実験システムの全体構成を示し(5.2節)、各ロボットの構成の概要を説明する(5.3節)。さらに、各ロボットの構成要素として、物体の協調ハンドリングに関する部分(5.4節)および環境情報の認識部分(5.5節)について詳しく説明する。

後半ではこのシステムを用い未知障害物存在下での協調搬送実験を行いロボットの観測結果に基づくタスク割り当て過程について評価を行う(5.6節)。

また、実機による環境情報の観測過程の実現方法や観測結果の誤差がアーキテクチャに与える影響とその対処について、5.7節において議論する。

## 5.2 実験システムの全体の構成

4.1節で示した方針に基づき構築した協調搬送実験システムの全体の外観とその構成をFig.5.1およびFig.5.2に示す。また、Fig.5.3に、ホストおよびロボットが1計画周期の中での処理、通信時期およびその内容を示す。

本実験では、2台の全方向移動ロボットを用いる。搬送対象物にはキャスタ付きの机(1.2 × 1.6[m])を用い、これを押すまたは引くことでハンドリングを実現する。各ロボットは有線イーサネットを介してホストコンピュータ(SUN製のSparcStation20)と接続されている。

ロボットの構成について述べたのち、ハンドリング、および外界認識に関して、さらに詳しく述べる。

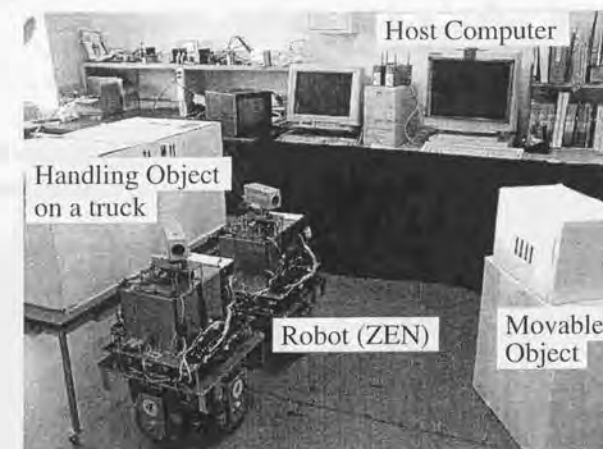


Fig. 5.1: Experimental Setup



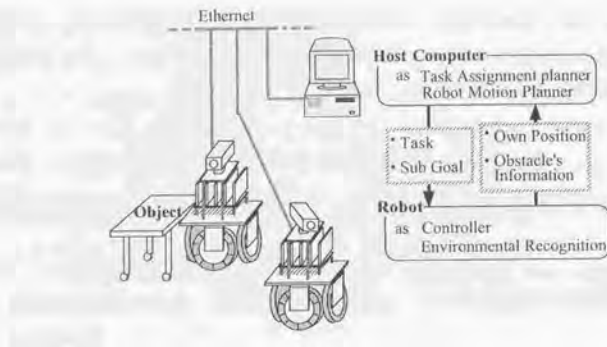


Fig. 5.2: Data flow in Whole Experimental System

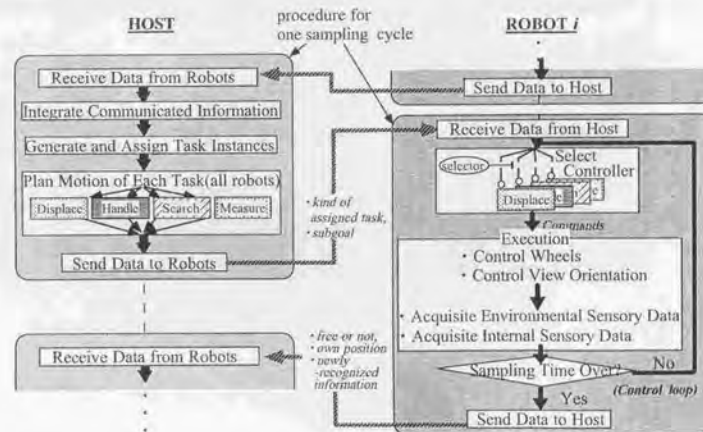


Fig. 5.3: Procedure in one planning cycle for Host and Robots

### 5.3 各ロボットの構成

実験に用いたロボットは、理化学研究所製ZENをもとに、ハンドリングや外界認識のためのシステムを付加して構築したものである。

移動ロボット本体の外観およびハードウェア構成図を Fig.5.4に、制御系を Fig.5.5に、主な仕様を Table 5.1に示す。

各ロボットにはIntel製CPU 486DX4(100MHz)を用いたマザーボードを搭載しており、ISAデータバスを通じて接続された各種ボード類を用いて制御を行う。各ロボットは、ホストからの指令（タスクと目標位置）に基づいて

- 自己位置
- 外界センサであるCCDカメラの姿勢

を制御し、CCDカメラ画像の処理により得られた障害物位置などの環境情報をホストに通信する。ただし、目標位置については、各ロボットの走行誤差やセンシング誤差を考慮したものではないため、各ロボットは、ホストに与えられた値を自己のロボット座標系における値に変換した後、内力回避などの必要に応じてリアクティブに変更していく。

次節からは、物体のハンドリングおよび外界認識に関する部分について、さらに詳しく述べる。

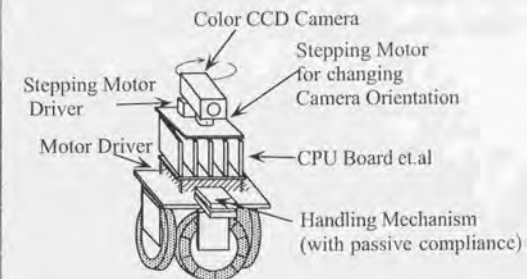
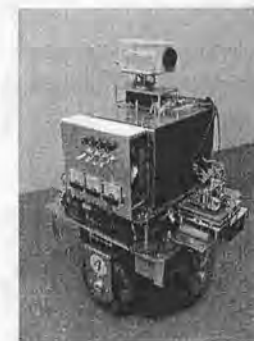


Fig. 5.4: Profile of Each Robot

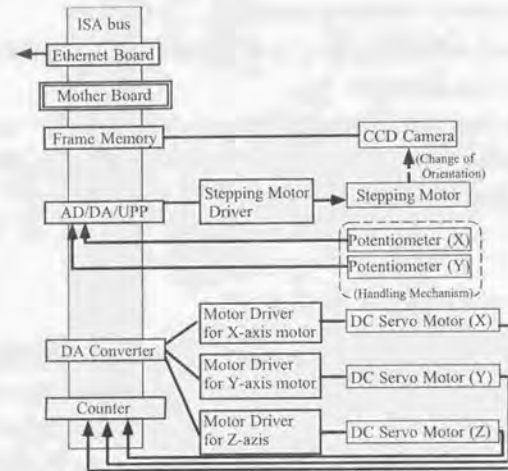


Fig. 5.5: Hardware Structure of Each Robot

Table 5.1: Main Specifications of Robot

Size	approximately $0.4(W) \times 0.5(D) \times 0.7(H)$ -[m]
Weight	approximately 24.0 [kg]
Communication(with Host)	Wire Ethernet(TCP/IP Protocol)

## 5.4 物体ハンドリング部

本論文では3章において、移動ロボットによる力の干渉を伴う協調ハンドリング作業実現のために考慮すべき課題のうち、

- (1) 車輪型移動機構に特有の非ホロノミック性
- (2) 移動誤差の存在下での共通物体の搬送

の二つの課題に対し、機構面から解決する方法としてFig.5.6（再掲）に示す機構を用いる事を提案し、その有効性を検証した。

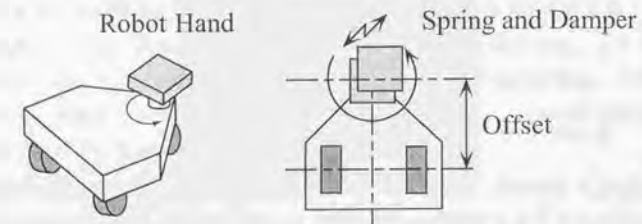


Fig. 5.6: Schematic view of the Handling Mechanism

本章で行う実験においては全方向移動ロボットを用いるため、上記のうち(2)に対する設計議論を踏襲し、移動ロボットの走行誤差に起因する過大な内力の回避のための受動コンプライアンス機構を導入する。さらに

- 物体を固定把持できること
- 対象物から離れる、合流する、といった作業の実装・実現が容易であること

という点を考慮して、Fig.5.7に示す搬送機構を試作し用いた。

この機構は、対象物を固定把持または解放するための脱着部分を、対象物の運動する水平面内のコンプライアンスを実現する $xy$ スプリングユニット部に載せる形態となっている。 $xy$ スプリングユニットについては、ロボットの大きさに合わせ、バネ定数は約 $1[\text{kgf/cm}]$ 、ストロークは $\pm 25[\text{mm}]$ となっている。



脱着部分に関して、各ロボットは突起部を対象物側の穴に抜き挿しすることで対象物を掴む、離すという状態を実現する (Fig.5.8)。このように棒状の突起を対象物に引っかけて搬送するという手法は、現在実際に実用化されている自動搬送車 (AGV) による運搬作業でも良く用いられる手法である。応用例としては病院における使用済みシート自動回収作業などがあり、シートを入れたキャスタ付きの箱の運搬にこの手法を用いている。

また棒状の突起部を対象物側の円形の穴に挿入するため、各ロボットの手先で過大なモーメントは発生せず水平面内2次元の力のみ発生するようになっている。そのため、ハンドリングには最低2台のロボットが必要とされる。



Fig. 5.7: Mechanism for Handling an Object

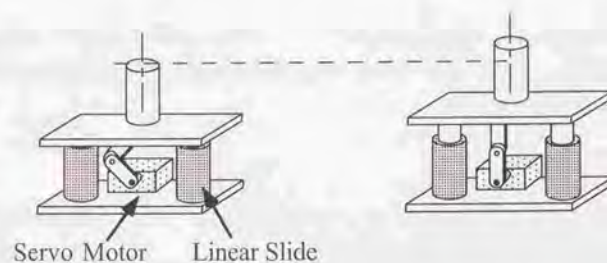


Fig. 5.8: Mechanism for Handling an Object 2

## 5.5 環境認識部

### 5.5.1 認識支援装置を用いた環境認識システム

環境に存在する障害物の認識は、ロボットに搭載した CCD カメラによって行う。ロボットが認識すべき内容は

- 障害物の位置姿勢
- 障害物の属性 (形状や重さなど除去タスクに必要な情報)

である。しかしこれらの情報を一般的な画像処理のみによって行うには実装および実行の手間がかかる。そのため、ここでは認識の支援装置を導入することを考える。

支援装置としては、大きく分けて、3次元的なマーカを用いる手法と、2次元的なマーカを用いる手法がある (例えば [新井94])。認識の精度という観点からは、一般に3次元的なマーカが望ましいが、製作および取り付けの簡便さから、ここでは2次元的なマーカを貼付することにする。

2次元マークを用いた環境認識支援に関する研究としては、[Kim92] や [太田99] 等がある。[Kim92] は1次元バーコードを用いる手法であり、計測アルゴリズムが非常に単純であるために高速な計測が可能となっている。また、マーカをロボットの行動支援のために積極的に利用した研究として [太田99] が行われている。この研究では2次元バーコードを組み込んだ正方形のマーカを用いており、マーカ形状の計測により相対位置の計測を、2次元バーコード部による物体の形状やその操作方法等の記述により、ロボットがあらかじめ環境に関して持つべきモデルを減らすことが可能となっている。

ここでは、これらの研究を参考に、

- 製作が簡便であること
- 認識速度が速いこと
- 通常の照明条件下で認識可能なこと

という要件を満たすマークとして、Fig.5.9に示すものを作成し用いた。

これは、属性を識別するための色情報部分と、その位置姿勢を計測するためのランドマーク部分から成る。ロボットにはあらかじめ、各色に対応した物体の属性モデルを持たせ、色の識別により、対応するモデルを認識させることにする。また、バーコードのよう

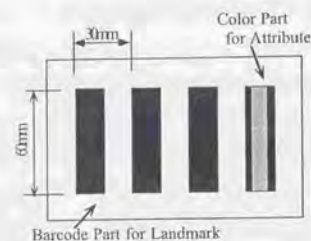


Fig. 5.9: Profile of the 2D mark

に一定間隔で黒から白に変化するものを用いることで、周辺の雑音からのマーク検出が容易になる、というメリットがある。

### 5.5.2 相対位置計測アルゴリズム

マークは、ロボットのカメラと同じ高さを中心に、地面と垂直な面に貼付すると仮定すると、このマークを用いた環境認識処理の流れは以下の通りとなる。

Step0 計画された初期姿勢にカメラを向ける。

Step1 残り時間があれば画像を取得。

Step2 白黒の切り替わる連続パターンがないか横方向に1ライン分の画素を検索する (Fig.5.10の scan1)

見つかった場合は  $x_{p1} \sim x_{p4}$  の座標値を導出し、Step3へ進む。

見つからない場合は Step5へ進む。

Step3  $x_{p1} \sim x_{p4}$  の座標値をもとに、縦方向に1ラインずつ検索し (Fig.5.10の scan2, scan3)、輝度の切り替わりから、 $P_1 \sim P_4$  の座標を特定。

Step4 四角形の4頂点である  $P_1 \sim P_4$  の座標値から、[高橋96a] [高橋96b] のアルゴリズムを用い相対位置姿勢を計算。

Step5 カメラの向きを  $\Delta\theta$  だけずらし、Step1へ戻る。

Step6 終了。

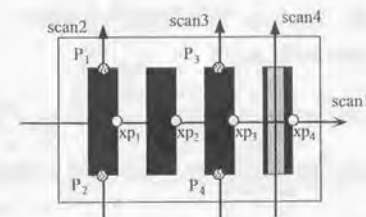


Fig. 5.10: 2D-mark Recognition Procedure

### 5.5.3 マークの認識精度評価実験とロボットのセンシングエリアの特定

本システムの精度実験を行った。本測定実験における CCD カメラとマークの位置関係を Fig.5.11 に示す。CCD カメラの光学中心とマークの中心を結ぶ線を視線方向と呼ぶ。視線方向と CCD カメラの中心軸とのなす角を  $\alpha$ 、視線方向とマークの法線ベクトルのなす角を  $\beta$  とすると、マークの相対姿勢  $\theta = \beta - \alpha$  と表される。またマークの相対距離  $d$  は CCD カメラの光学中心とマークの中心点との距離である。

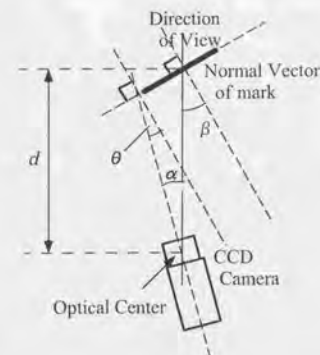


Fig. 5.11: Camera and Mark — Variables in Recognition

距離、角度ともに10回の計測値の平均値をグラフにしたものをそれぞれ Fig.5.12 および Fig.5.13 に示す。両図とも横軸には真の  $d$  の値をとる。縦軸には、Fig.5.12 では  $d$  の測



定値の真の値からの誤差を, Fig.5.13では $\theta$ の測定値の真の値からの誤差をとる.

精度実験の結果, 次の傾向が見られた.

- 距離については近いほど全体的に誤差が小さくなる
- 姿勢については $0[\text{deg}]$ に付近が最も悪く, 角度が大きくなるに従って真の値に近づく.

またロボットのセンシングエリアについて, マークが画像に占める大きさから最小認識可能距離は $500[\text{mm}]$ であり,  $2000[\text{mm}]$ を越えた地点での計測は角度計測に平均5回以上失敗することからこれを最大認識可能距離と設定した. また視野角は $20[\text{deg}]$ であった.

#### 5.5.4 色識別精度評価実験

今回採用したマークにより, 物体の位置姿勢および属性の認識がロボットの走行中に可能であることを確認するため, 単体ロボットを用いたマーク認識に基づく走行実験を行った.

実験では, Table 5.2に示すように色情報の異なる2種類のマークを貼付した障害物を用意し, ロボットの目的地までの経路中にこれがおかれた時に, そのモデルにしたがって適応行動をとることが可能であるかどうかを見る.

実験は各マークにつき3回ずつ行った. マークの色情報についてはマーク手前 $800[\text{mm}]$ で認識した色情報を採用しているため, 物体の識別を間違えることはなかった.

Table 5.2: Experiment Conditions

Model No.	Color	Weight of an Obstacle	Robot Action
1	red	100 [kg]	avoid
2	green	1 [kg]	displace

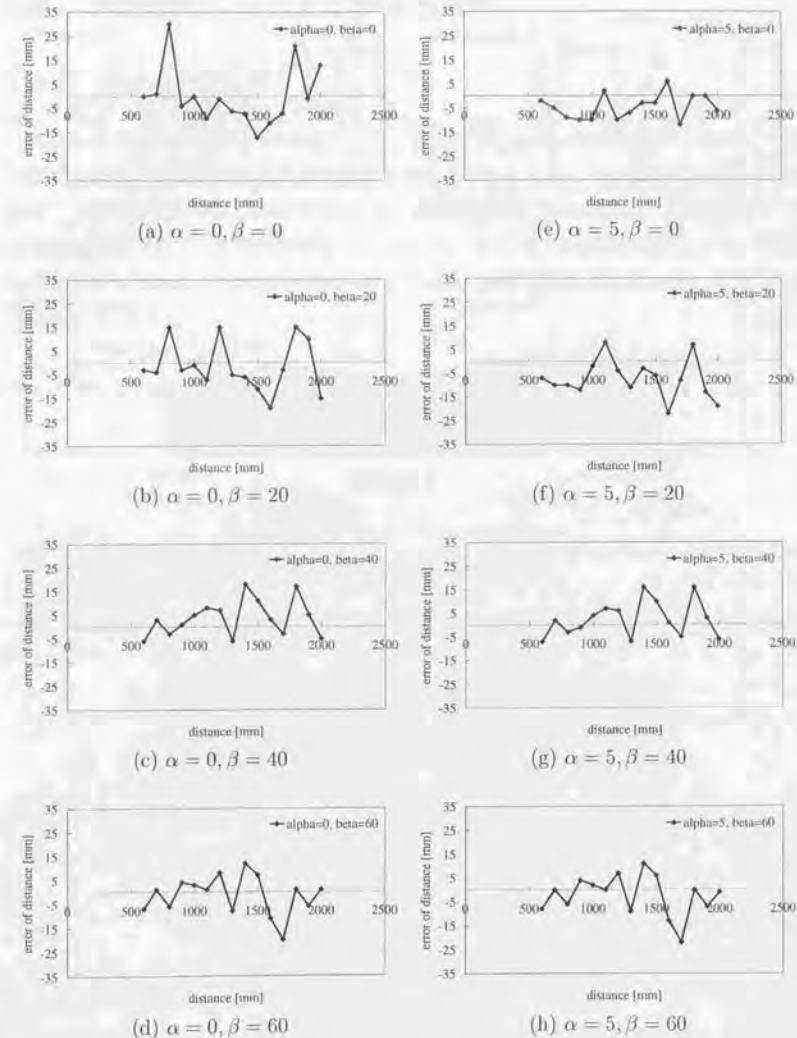


Fig. 5.12: Error in Recognized Distance

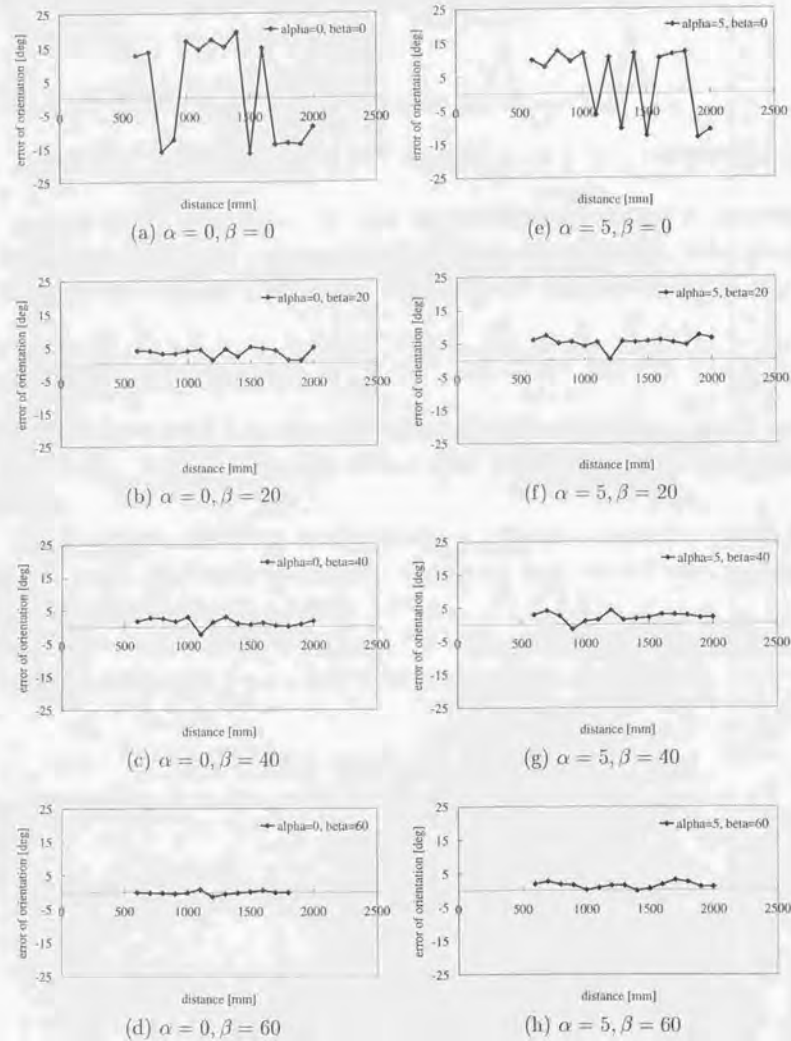


Fig. 5.13: Error in Recognized Orientation

## 5.6 未知環境における協調搬送実験

### 5.6.1 実験内容と目的

提案した手法の実機適応性の検証として、実ロボットによる観測に基づきロボット群の行動が実時間で計画され作業が実現可能であることを示すため、搬送軌道に可動障害物を設置した環境において協調搬送実験を行う。実験開始時点では、ロボット群に障害物の存在位置は与えられていない未知環境となっている。ただし、5.5節に示したように経路上に存在し得る障害物のモデルは事前に持っている。実験環境を Fig.5.14 に、実験条件を Table 5.3 に示す。

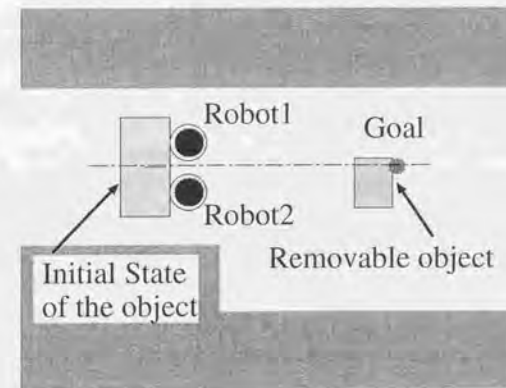


Fig. 5.14: Initial States of the Robots, the Object, and the Movable Object([m])

### 5.6.2 実験結果（時系列）

実験におけるロボット群の作業実行の様子を Fig.5.15 に、生成されたタスクインスタンスと割り当て結果を Fig.5.16 に示す。Fig.5.16 における小さい円は生成されたタスクインスタンスであり、下向きの矢印および数字の 1 (または 2) の描かれたものは、robot1(ま



Table 5.3: Experiment Conditions

the shape of the object	a $1.2 \times 0.6$ -[m] rectangle
the shape of the movable object	a $0.6(w) \times 0.4(d) \times 0.8(h)$ -[m] box
the size of the robots	$0.4(w) \times 0.5(d) \times 0.7(h)$ -[m]
sensing area of the robots	a sector with a $2.0$ -[m] radius and a $20$ -[degree] interior angle
maximum velocity of the robots	$0.1$ [m/s]
division of work space for sensing	a $0.2 \times 0.2$ -[m] rectangle
planning cycle of the host computer	$3.0$ [s]

たは *robot2*) に割り当てられたことを示す。矢印の長さは、割り当てられたインスタンスが実行され完了されるまでの期間を示す。

作業が開始した時点では、ロボットは周辺に関しての情報を持っていないため、周辺監視タスクインスタンスおよびハンドリングタスクインスタンスのみが生成され、3.0秒からの計画周期では2台のロボット両方に周辺監視およびハンドリングが割り当てられる (Fig.5.15 (a))。ただし、この時点では周辺の監視不足のため、動作計画器の判断で移動が抑制され実際に動くことはせず、十分に監視が行われたと判断された12.0秒から2計画周期 (6秒) の間、ロボット群は実際に前進して物体のハンドリングを行う。

3.0秒からの計画周期において *robot1* により発見された障害物は、12.0秒の計画周期において、5計画周期以内に対象物をハンドリングするロボット群と衝突すると判断され、それを除去するためのタスクインスタンスが生成される。

12.0秒から4計画周期の間、“物体ハンドリング” “周辺監視” “障害物除去” の全種類のタスクのインスタンスがロボット台数を越えて生成される。このうち、初めの3計画周期の間は高い優先度を持つ周辺監視タスクインスタンスが複数存在するためこれが優先された割り当てとなる。4計画周期目の21.0秒では、周辺監視タスクの優先度が減少し除去タスクの優先度が相対的に高くなるため、観測の結果障害物により近いと判断された *robot2* が対象物を離れ“障害物の除去”に向かう。

その間 (Fig.5.15(c)-(g)), *robot1* は1台で物体をハンドリングすることはできないため、その場に停止待機する計画がなされ、周辺監視を行いながら *robot2* が戻ってくるのを待つ

ている。なおこの間対象物は停止しているが、時間経過に伴い監視必要領域のタイマー値が閾値を越え、その領域に関する環境情報の更新必要性が生じるのに対応して、優先度の高い周辺監視タスクが生成されている。

72.0秒において *robot2* が障害物除去作業を完了し、それを宣言すると、75.0秒で *robot2* は対象物付近まで戻って周辺監視およびハンドリングを続行するよう計画された、ここで実験は終了している。

Fig.5.17に、上記で説明したロボット群および対象物の動作軌跡と、可動障害物の最終的なコンフィギュレーションを示す。図中の点線で描かれた円はロボットの初期位置を表す。さらに小さな円は *robot2* の移動した軌跡 (デッドレコニングによる値) であり、その横に付された数字は通過した時刻を表している。

### 5.6.3 実験結果の評価

#### 観測結果の誤差

Fig.5.18に、ロボットにより観測された障害物のコンフィギュレーションを示す。ロボットのセンシング誤差のため、観測された除去対象障害物のコンフィギュレーションと実際のものとは“ずれ”が見られた。しかし、発生した障害物除去タスクインスタンスが、*robot1* でなく実際の位置が障害物に近い *robot2* に割り当てられたこと、またその除去タスクが完了されたことから、作業遂行上問題のない程度に十分小さかったと言える。

#### 実時間処理性

タスクの割り当て器および動作計画器についてはホストコンピュータ (SPARCstation20) 上に実装したが、ホストコンピュータにおける計算時間は、各ステップ平均で0.03秒、ホストコンピュータとロボットの間の通信時間は各ステップ平均0.03秒であった。これらの和は1計画周期の3秒に対し十分小さい。

1計画周期の長さは、ロボットのセンサであるCCDカメラの視野角が $20[deg]$ と小さく一度の処理で監視可能な領域が狭い事と、CCDカメラの姿勢変更に要する操作時間 (回転の最大角速度 $??[rad/s]$ ) とを考慮して決定した。この時、ロボットの最大並進速度が約0.1 (平均0.07) [m/s]で、環境内に移動障害物が存在しないため、1計画周期で生じるロボット群と環境との相対的位置関係の変化量が最大の場合でも、次の計画周期で十分対処可能であると考えられる。

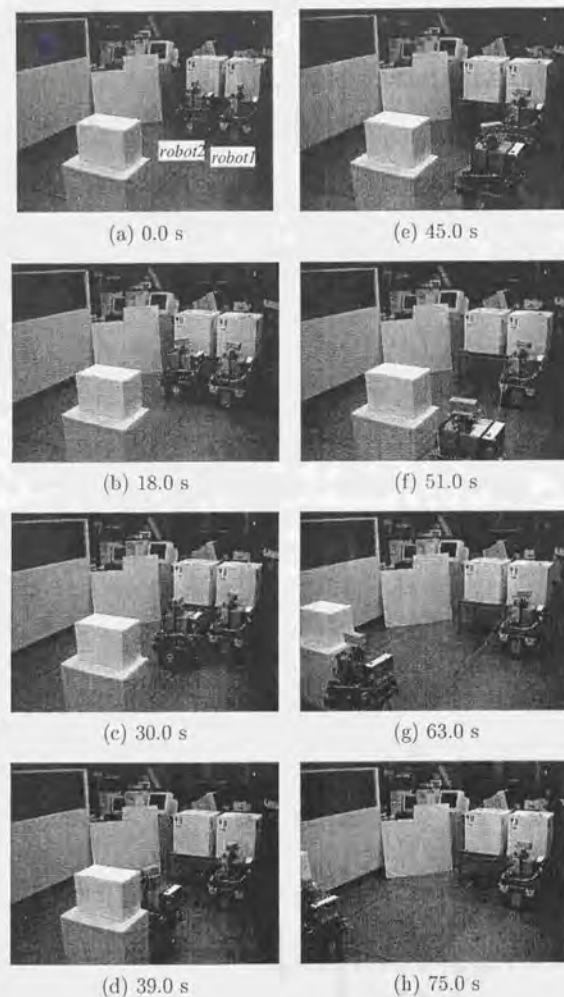


Fig. 5.15: Experimental Results (time transition)

したがって、提案手法により実時間での計画が可能であると言える。

また、今回は、ロボット群に与えられた搬送対象物の目標コンフィギュレーションに到達していない75.0秒の段階、すなわち主目的を達成していない段階で実験が終了している。しかし、ロボットによる実際の観測結果に基づき、異種作業割り付け型協調搬送作業を行うロボット群の行動が計画可能であるということを示すには、十分な内容であったと考えられる。

以上より、提案アーキテクチャにより、ロボットが実際に環境を観測した結果に基づき、群ロボットの動作を実時間で計画し、実行することが可能であることが分かった。



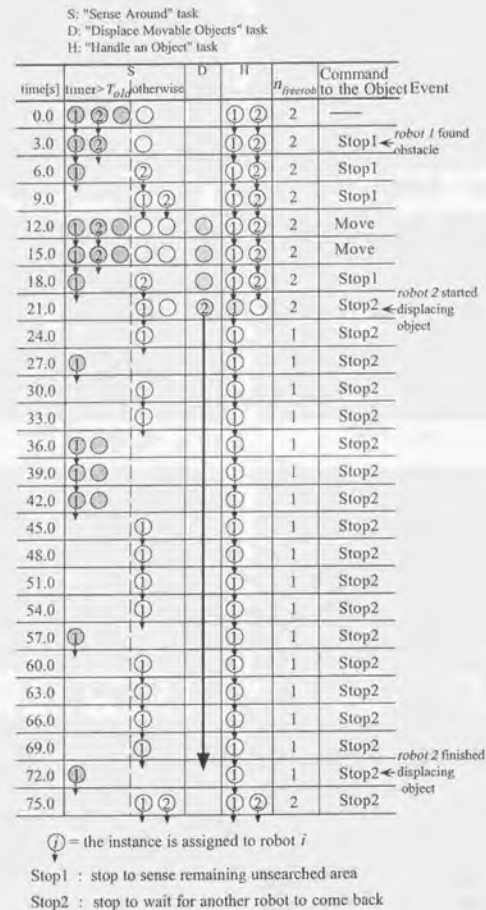


Fig. 5.16: Generated Task-instances during the Experiment

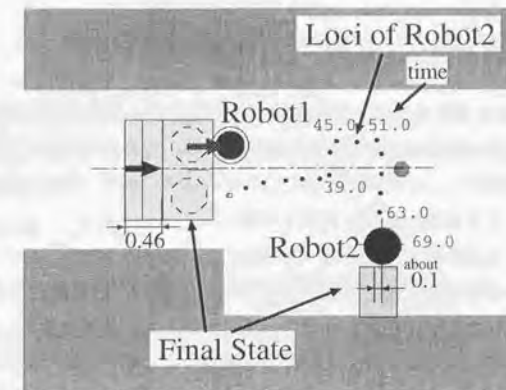


Fig. 5.17: Loci of the Robots and the Object ([m])

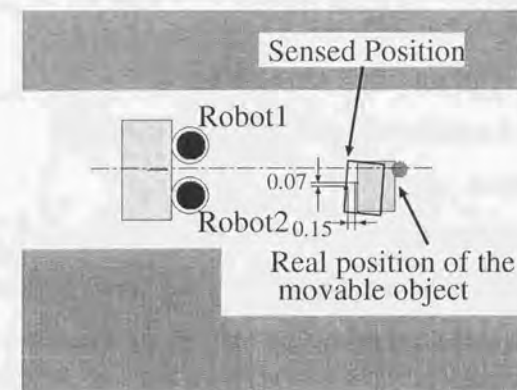


Fig. 5.18: Sensed position of the Removable Object([m])

## 5.7 実ロボットによる環境認識過程についての考察

### 5.7.1 観測動作の組み込み方について

CCDカメラによって得られる物体の画像のみをもとに、その物体の形状から属性まで高速に認識する事は、現状の技術では困難である。そこで本章における実験では、障害物に2次元マークを貼付し、マークの計測によりその物体の属性(形状や重さなど)も認識可能なシステムにより環境を認識し作業を実現した。

このような認識支援システムを用い、環境側に完全に情報を埋め込むことが出来る場合には、提案手法をそのまま適用することができる。しかし、適用範囲を広げるためには、形状等の未知な対象物体が存在する場合にも対応することが重要となる。このような場合には、例えば[佐々木99]のように、未知情報を取得するための行動をタスクとして組み込む事で対応可能であると考えられる。

### 5.7.2 タスク割り当てにおけるセンシング誤差の影響

実ロボットにより環境を観測する場合、その誤差を避けることはできない。ここで、タスク割り当ての枠組みに対するセンシング誤差の影響を考える。

センシング誤差には次の二種類が存在する。

(error1) 対象となる障害物の位置・姿勢の誤認識

(error2) 対象となる障害物の属性(形状、重さ、性質など)の誤認識

(error1)の影響としては、

- 当初計画された除去軌道が、本来あるべき軌道からずれる
- 異なるロボットが同一の物体を発見した時に、異なる物体として認識する場合がある

といったものがある。しかし前者については、割り当てられたロボットがタスクを実行する過程で得られた情報をもとに、各ロボットが動作計画器のレベルで対応することが可能である。また、後者についても、環境情報地図上で情報を融合する際に、近すぎる物体は同一とみなす、といった処理が可能である。今回実験結果として掲載した例では生じていないが、実際他の試行において、除去に向かわずに残っていたロボットが、現在除去の対象となっている物体を視野内に捉え、その位置情報をホストに通信するケースが存在し

た。しかし、発見した物体は、物体間の距離の計算により、既に発見済みの物体と同一とみなされ、インスタンスの生成には至らなかった。

(error2)について、本実験では環境認識の支援装置として物体の位置・姿勢および属性の計測可能なマークを利用したため、マークの識別さえ成功すればロボット群は障害物の正確なモデルを取得することが可能であり、(error2)は理論的には生じない。しかし、一般的な作業環境においてすべての物体を誤差なく認識できるわけではなく、その影響を考慮する必要がある。

(error2)の影響としては、

- 除去可能であると見積もったにも関わらず除去不可能であった
- 1台で除去可能であると見積もったが1台では不可能であった

といった割り当て結果の無効性あるいは不適切性が考えられ、この場合、割り当てのやり直しが必要となる。今回、割り当て結果の有効・無効に関する評価器の部分までは実装していないが、提案アーキテクチャは一定計画周期毎に常にタスク割り当ての再計画を行うという枠組であるため、容易に拡張可能であると考えられる。



## 5.8 おわりに

本章では、実際のロボットシステムを構築し、事前に設置位置が未知な障害物の存在する環境における搬送実験を行うことで、提案手法の実現性に関する検証を行った。

まず、実ロボットシステムとして、全方向移動ロボット2台およびホストコンピュータからなる系を構築した。力の干渉を伴う協調ハンドリング作業を行うため、3章において扱った実機問題のうち、走行誤差の存在するロボット間での協調を可能とするための機構設計に関する議論を踏襲した機構を試作した。すなわち受動コンプライアンスを含む搬送機構を用い、さらに対象物からの乖離および合流を容易に実現する脱着機構をその上部に搭載した。またCCDカメラおよび認識支援のための色情報つきバーコード状2次元マークを用いた環境認識システムを構築した。

上記システムを用い、事前に設置位置が未知な障害物の存在する環境下における搬送実験を行ったところ、実際に、搬送経路上での物体のハンドリングおよび搬送経路を塞ぐ物体の除去という行動を得る事ができた。このことから、ロボットが観測によって得た環境情報に基づき、タスクの動的な生成・割り当て計画・動作計画という一連の行動計画および計画に基づく行動が、実時間で実現可能であることが分かった。

したがって、本論文で提案した異種作業割り付け型協調作業のための行動計画アーキテクチャが、実ロボットシステムに適用可能であることが示された。

## Chapter 6

### 結論

6.1 結論	182
6.2 今後の展望	184

## 6.1 結論

本論文では、複数台の移動ロボットによる未知環境における協調搬送システムの構築を目的として行った。

動的な変化を伴う未知環境において、複数台の移動ロボットにより協調搬送作業を行うためには、障害物の発見、衝突回避、軌道確保のための障害物の移動など、必要情報の観測過程を含む種々の動作を行う必要がある。しかし従来の協調搬送に関する研究は、制御問題としての側面のみを扱い外部環境との関係に基づくロボット群の動的な行動計画を扱っていないものがほとんどであった。そこで本研究では、ロボットが必要に応じて実行するタスクを切替える“異種作業割り付け型協調搬送システム”を提案した。実現のための具体的な課題として、“タスク間の相互関係を考慮した協調作業のための実時間タスク割り当て手法の構築”“持ち替えを伴う協調ハンドリング動作計画手法の構築”“実機問題の考慮に基づく協調ハンドリング作業実現”の3つを設定し研究を行った。

環境との幾何的な関係に基づく“持ち替えを伴う協調ハンドリング動作計画手法の構築”という課題では、以下の解決手法を示した。まず、目標コンフィギュレーション達成までの動作を短時間で分断して捉え、周期的に到達可能な範囲内での動作計画を局所情報に基づき行う枠組を考えた。また、各周期の動作計画を実時間で可能とするため、問題を“対象物”および“ロボット”で分離し、局所動作計画器および制約条件付非線形計画手法を適用することを提案した。シミュレーションにより、通過可能な軌道上に小型の未知障害物がある程度離れて存在する単純な環境において、障害物との衝突や対象物の落下を起こさず目標コンフィギュレーションを達成するロボット群の動作が計画可能である事が分かった。すなわち、上記環境における把持位置変更を伴うロボット群協調ハンドリング動作計画手法を構築できた。

“実機問題の考慮に基づく協調ハンドリング作業実現”という課題では、移動機構に起因する問題を、周辺環境との幾何的な関係に基づく上記動作計画問題と分離することを考え、機構設計問題および動作計画結果の採用条件という形で扱った。4台の実ロボットを用いた実験から、ロボットのトルク限界を見積もることで、走行誤差を有する非ホロノミックな移動ロボットによる協調ハンドリング作業が実現可能であることが示された。

“タスク間の相互関係を考慮した協調作業のための実時間タスク割り当て手法の構築”という課題に対しては以下の解決手法を示した。まず環境の観測過程を含めた種々のタスクを、1台のロボットにより一定短時間で実行可能な作業という単位（タスクインスタンス）で捉えた。また、あるロボットがあるタスクインスタンスを実行する状態を“優先度”

として評価する枠組を準備した。優先度は、タスク同士の先行関係に加え、協調動作を行うのに必要な最低限のロボット台数およびロボット群を含めた作業環境のコンフィギュレーションとの関係に基づき決定される。これにより、一定短時間内のタスク割り当てが実時間で決定可能な形に定式化された。上記タスク割り当て器と、1周期分の動作を計画する動作計画器と組み合わせ周期的に計画を行う行動計画器を提案し、シミュレーションを行った。シミュレーションでは、小型で除去可能な障害物や移動障害物が少数存在する環境を想定した。その結果、対象物周辺の観測を行いながら協調して物体をハンドリングし、発見した障害物を除去あるいは回避することで軌道を確保し、目標コンフィギュレーションを達成する、という協調搬送動作が計画可能であることが分かった。すなわち、数台の確保を要する協調ハンドリング作業、あるいは協調ハンドリング作業に先行した静止障害物の除去作業といった、相互関係の強く決まったタスクを実行するロボットの行動計画が可能であることが分かった。

また、協調ハンドリング動作実現のための機構を備えた全方向移動ロボット2台から成る搬送システムを構築し、存在位置が未知である除去可能な障害物の存在する環境において協調搬送実験を行った。これにより、実ロボットによる実際の観測を元にした異種作業割り付け型協調搬送作業が実時間で実際に可能であることが示された。

各課題に対する解決策を統合したシステムが全体として調和的に機能し、障害物の存在する未知環境において、協調ハンドリング・障害物の観測・軌道確保のための障害物の移動という行動により、協調搬送作業を実現可能であることが、シミュレーションおよび実験によって示された。

以上より、はじめに対象として想定した未知環境における複数台の移動ロボットによる協調搬送システムが構築できた。



## 6.2 今後の展望

未知環境における複数台の移動ロボットによる協調作業の実現化に向け、本論文を通じて明らかになった課題を以下に示す。

### 割り当て後の状況変化を考慮したタスク割り当て手法

本論文で提案した手法では、短時間に分断して捉えられるタスクに対し、実行順序等のタスク間の相互関係を満たした割り当てが可能である。しかし、障害物除去のように数周期に渡って実行を続けるタスクにロボット群のうちの大多数が割り当てられた後に、放置した対象物に移動障害物が近付くなど状況が変化する場合が存在する。この時、軌道変更といった別の作業を現在のものよりも先に解決する必要が生じる。こうした状況は、ロボット群を構成する台数が少なく、環境内に動的に変化する要素が多い場合に起こりやすい。このような状況変化に起因する作業手順の変更に対応するためには、割り当て状態の監視および解除機構を導入する必要がある。これが今後の課題である。

### 異質なロボット群による協調作業への適用

本論文では、ロボットの能力が同質な場合について、必要な作業量を満たし得るロボット台数を見積もり、タスクインスタンスの生成・割り当てを行っている。実際の導入時には、トルク限界などの異なるロボットが存在すると考えられ、その場合にも、最も弱いロボットに合わせて作業を見積もることで提案手法を適用可能である。しかし、力の弱いロボットならば2台で、強いロボットならば1台確保など、それぞれの能力を反映した割り当てが可能になれば、作業の実行効率が向上すると考えられる。

### 協調維持のための上位協調／下位協調

本論文では、有線通信を行うなど与えられた指令を全体が十分同期して実行可能な状況において、実ロボットを用いた作業の実現性が検証された。しかし、無線通信を行った場合に起こりうる通信遅れにより同期がずれた場合や、協調動作を行っていたロボットがなんらかの理由で緊急停止した場合などには、残りのロボットがこうした状態をいち早く検知し、これに追従して振舞うなど反射的な行動により最低限の協調状態を満たす下位協調の仕組みが必要である。

また、現在の群を構成するロボットだけでは、時間に関係なく解決が不可能なタスクが発生した場合には他からの助けを呼ぶ必要があるが、こうしたロボット群間の調整を行う上位の機構も必要となる。

### 人間の介在した協調作業の実現

環境に対する柔軟な対応を行うためには、状況を適切に認識する能力は不可欠であるが、現状ではその状況認識能力の低さが、ロボットによる作業実用化における難関の一つでもある。

また、どのような状況においても最適とみなされる群行動を設計することは難しく、作業進行に伴って、その環境に合わせた行動になるよう自己を調整していく必要があるが、現在の状態が好ましいかどうかという判断を自動的に下すのは難しい。

このような状況に対応するため、状況認識能力に長けた人間を積極的に介在させた協調作業を可能とする枠組が有効である。この場合、通信を行う事のできない人間の意図や指示をいかに提示し、また認識するか、という点が課題であるが、限定的な認識問題となるため、あらゆるものの認識を目指すよりも実現性の高いシステムになると期待される。

本書的出版，首先要感謝我的家人，特別是我的母親，她一直支持我完成這項工作。其次要感謝我的老師，他們在學術上的指導對我來說至關重要。最後要感謝我的同學和朋友，他們的鼓勵和幫助讓我能夠堅持下來。

本書的內容主要分為三個部分。第一部分是關於理論的探討，第二部分是關於實踐的應用，第三部分是關於未來的展望。在寫作過程中，我參考了大量的文獻資料，並與多位專家進行了交流。本書的出版得到了許多人的支持，特別是我的導師和同學們。他們的批評和建議對我來說非常有價值。本書的出版也得到了一些資助機構的支持，這對我來說是一個巨大的鼓勵。

本書的出版是一個艱辛的過程，但也是一個充滿意義的過程。我希望本書能夠為讀者提供一些啟發和幫助。本書的出版也得到了一些資助機構的支持，這對我來說是一個巨大的鼓勵。本書的出版也得到了一些資助機構的支持，這對我來說是一個巨大的鼓勵。

本書的出版是一個艱辛的過程，但也是一個充滿意義的過程。我希望本書能夠為讀者提供一些啟發和幫助。本書的出版也得到了一些資助機構的支持，這對我來說是一個巨大的鼓勵。本書的出版也得到了一些資助機構的支持，這對我來說是一個巨大的鼓勵。

## 謝辭

本書的出版，首先要感謝我的家人，特別是我的母親，她一直支持我完成這項工作。其次要感謝我的老師，他們在學術上的指導對我來說至關重要。最後要感謝我的同學和朋友，他們的鼓勵和幫助讓我能夠堅持下來。

本書的內容主要分為三個部分。第一部分是關於理論的探討，第二部分是關於實踐的應用，第三部分是關於未來的展望。在寫作過程中，我參考了大量的文獻資料，並與多位專家進行了交流。本書的出版得到了許多人的支持，特別是我的導師和同學們。他們的批評和建議對我來說非常有價值。本書的出版也得到了一些資助機構的支持，這對我來說是一個巨大的鼓勵。

本書的出版是一個艱辛的過程，但也是一個充滿意義的過程。我希望本書能夠為讀者提供一些啟發和幫助。本書的出版也得到了一些資助機構的支持，這對我來說是一個巨大的鼓勵。本書的出版也得到了一些資助機構的支持，這對我來說是一個巨大的鼓勵。

本書的出版是一個艱辛的過程，但也是一個充滿意義的過程。我希望本書能夠為讀者提供一些啟發和幫助。本書的出版也得到了一些資助機構的支持，這對我來說是一個巨大的鼓勵。本書的出版也得到了一些資助機構的支持，這對我來說是一個巨大的鼓勵。



本論文は、私が東京大学大学院 工学系研究科 新井・湯浅・太田研究室において、6年をかけて行った研究をまとめたものです。長きに渡り御指導を頂きました

東京大学大学院 工学系研究科 精密機械工学専攻 教授 新井 民夫 先生  
に、心から感謝いたします。先生とのディスカッションでは、豊富な知識に基づく多様な視点からの含み深い御助言を頂き、本論文をまとめるにあたり大変参考になりました。この6年を頑張ることができたのも、温かくフェアな精神の持ち主である先生のものであったからこそです。先生の教えを糧に、これからも研究者として成長すべく精進してゆきたいと思います。

また、本論文をまとめるにあたり、副査をお引受け下さいました

東京大学大学院 工学系研究科 精密機械工学専攻 教授 木村 文彦 先生

東京大学大学院 工学系研究科 精密機械工学専攻 教授 小林 郁太郎 先生

東京大学大学院 工学系研究科 機械情報工学専攻 教授 佐藤 知正 先生

東京大学 生産技術研究所 助教授 藤井 輝夫 先生

には、論文全体に渡って建設的な御指導を頂きました。ここに謹んで感謝の意を表します。

東京大学大学院 工学系研究科 精密機械工学専攻 助教授 太田 順先生には、研究内容だけでなく、初歩的な作業から、研究の進め方、研究者としての在り方まで、多岐にわたるきめ細やかな御指導を頂きました。本当にありがとうございました。再び元気なお姿で現場にお戻りになれることを、今はただ、心からお祈りしております。

東京大学大学院 工学系研究科 精密機械工学専攻 助教授 湯浅 秀男先生、そして本研究室の助手であり同期でもある前田雄介氏には、本論文の理論をまとめるにあたり的確な御指摘を頂き、大変参考になりました。ありがとうございました。

理化学研究所の浅間 一先生には、ロボットシステムの実現だけでなく、理論構築においても多くの適切な御助言を頂きました。また、東北大学 工学系研究科 教授 小菅 一弘先生、中央大学 理工学部 助教授 大隅 久先生には、研究分野が近いことから、学会など、折に触れて有益な御指摘を頂きました。ここに改めて感謝いたします。

本研究室の元助手である相山康道氏（現 筑波大学講師）には、文字通り“手”のかかる複数の実ロボットを扱うにあたり、その豊かなハードウェアをもって多大な御助力を頂きました。また研究室の先輩方である、山口博明氏（現 機械情報工学専攻 井上・稲葉研究室）、吉田英一氏（現 通商産業省機械技術研究所）、倉林大輔氏（現 理化学研究所）、佐々木順氏（現 富士通研究所）には、在学中より、未熟な私を根気良く御指導頂きました。あ

りがとうございました。

本論文の理論展開から、シミュレーション、実験装置まで、ここまでの形にすることができたのは、本研究室の横井真浩氏（現 修士課程1年）、今西順一氏（現 修士課程2年）、そして山下淳氏（現 博士課程2年）の多大なる御尽力のお蔭です。大変感謝しております。

新井研に配属となった年に先輩として温かく迎えて下さった森素子（現 株式会社 東芝）さん、そして新井研と一緒に卒業することになる池田ひなたさん（現 修士課程2年）には、数少ない同性の同僚として、様々な相談に載って頂きました。彼女達と、同じ研究室で同じ時間を共有することができたことは、私にとって大変な幸運でした。

研究室の元秘書の岡田直子さん、篠崎久美子さん、牧野奈緒さん、高井美和さん、佐々木麻子さん、土方純子さん、そして現秘書の小坂井順子さん、井口幸葉さん、米岡道江さんには、煩雑な事務手続きなどで大変お世話になりました。試作室の斎治男さんには、部品加工のサポートだけでなく職人技も伝授頂き、モノがなくては始まらない私の研究を影で支えて頂きました。また、新井雅世様には、家族のように明るく温かく接して頂き、大事な場面毎に、心の籠った励ましのお言葉を頂きました。改めて、厚く御礼申し上げます。

最後に、我儘を通してこの道を選んだ私を、常に温かく見守り精神的に支えてくれた親愛なる家族に、心より感謝いたします。

2000年2月14日

宮田 なつき

1. Introduction  
The purpose of this study is to investigate the effects of the proposed method on the performance of the system. The results show that the proposed method significantly improves the performance of the system compared to the baseline method.

2. Methodology  
The proposed method is based on the combination of the existing method and the new method. The new method is designed to improve the performance of the system by reducing the error rate.

3. Results  
The results of the experiment show that the proposed method significantly improves the performance of the system. The error rate is reduced by 10% compared to the baseline method.

4. Conclusion  
The proposed method is effective in improving the performance of the system. The results of the experiment show that the proposed method significantly improves the performance of the system compared to the baseline method.

5. Future Work  
In the future, we will investigate the effects of the proposed method on the performance of the system in different environments. We will also investigate the effects of the proposed method on the performance of the system in different scenarios.

6. References  
[1] Smith, J. D. (2010). The effects of the proposed method on the performance of the system. *Journal of the American Statistical Association*, 105(488), 1234-1245.

## 参考文献

[2] Jones, A. B. (2011). The effects of the proposed method on the performance of the system. *Journal of the American Statistical Association*, 106(490), 1234-1245.

[3] Brown, C. D. (2012). The effects of the proposed method on the performance of the system. *Journal of the American Statistical Association*, 107(500), 1234-1245.

[4] White, E. F. (2013). The effects of the proposed method on the performance of the system. *Journal of the American Statistical Association*, 108(510), 1234-1245.

[5] Black, G. H. (2014). The effects of the proposed method on the performance of the system. *Journal of the American Statistical Association*, 109(520), 1234-1245.

[6] Green, I. J. (2015). The effects of the proposed method on the performance of the system. *Journal of the American Statistical Association*, 110(530), 1234-1245.



## 和文文献

- [木村97] 木村浩, 梶浦吾一, “動作理解に基づく操縦ロボットと自律ロボットの協調作業”  
日本ロボット学会誌, Vol. 15, No. 4, pp.624-633 (1997).
- [橋本92] 橋本雅文, 大場史憲, 中原宏章, “複数の移動ロボットによる階層型協調搬送制御”, 第6回知能移動ロボットシンポジウム予稿集, pp.75-80 (1992).
- [銭谷93] 銭谷 哲, 橋本雅文, 江口透, 大場文憲, “並進リンク機構を有する連結器を用いた群構成移動ロボットシステムの協調搬送制御”, 第11回日本ロボット学会学術講演会予稿集, pp.99-102 (1993).
- [橋本95] 橋本雅文, 大場史憲, 江口透, “複数の移動ロボットによる動的搬送制御法”, 日本ロボット学会誌, Vol. 13, No. 6, pp.886-893 (1995).
- [小菅98] 小菅一弘, 大住智宏, 千葉晋彦, “単一物体を操る複数移動ロボットの分散協調制御”, 日本ロボット学会誌, Vol. 16, No. 1, pp.87-95 (1998).
- [安井94] 安井裕司, 原文雄, 有竹俊之, “2 移動体の協調性の運動学的解析”, 日本ロボット学会誌 Vol.12, No. 7, pp.1013-1020 (1994).
- [篠島97] 篠島賢治, 原文雄, “正弦波路面における2 移動ロボットの特異点解析”, 日本ロボット学会誌, Vol.15, No.1, pp.124-130(1997).
- [小俣94] 小俣透, 永田和之, 岩槻重伸, 柿倉正義, “持ち替えによる多指ハンドの物体回転操作の計画”, 日本ロボット学会誌, Vol. 12, No. 5, pp.751-758 (1994).
- [木村93] 木村浩, 王志東, “複数移動ロボットによる宇宙構造物の操り(第1報, 操りの基礎理論)”, 日本機械学会論文集(C編) 59巻 559号 pp.227-233 (1993).
- [王93] 王志東, 木村浩, “複数移動ロボットによる宇宙構造物の操り(第2報, 操りの基礎実験)”, 日本機械学会論文集(C編) 59巻 559号 pp.234-239 (1993).
- [新井93] 新井民夫, 太田順, “仮想的なインピーダンスを用いた複数移動ロボット系の動作計画”, 日本ロボット学会誌, Vol.11, No.7, pp.1039-1046 (1993).

- [太田93] 太田順, “仮想的なインピーダンスを用いた複数移動ロボット系の動作計画”, 東京大学博士論文 (1993).
- [佐々木99] 佐々木順, “複数移動ロボットによる搬送作業のためのセンシング計画”, 東京大学博士論文 (1999).
- [福地99] 福地正樹, 山下淳, 河野功, 太田順, 新井民夫: “3次元環境下での持ち替え動作を伴う複数移動ロボットによる物体搬送経路計画”, 日本機械学会ロボティクス・メカトロニクス講演会'99 講演論文集, 1P1-30-035, pp.1-2 (1999).
- [和田97] 和田正義, 森俊二: “ホロノミック全方向移動ロボットの開発”, 日本ロボット学会誌, Vol. 15, No. 8, pp.1139-1146 (1997).
- [高橋96a] 高橋章, 石井郁夫, 牧野秀夫, 中静真: “人工現実感インターフェイスのための単眼画像からのマーカーの位置と姿勢の計測”, 電子情報通信学会論文誌A, Vol. J79-A, No. 3, pp.804-812 (1996).
- [高橋96b] 高橋章, 石井郁夫, 牧野秀夫, 中静真: “VR インターフェイスのための単眼による長方形マーカー位置精度実時間推定法”, 三次元画像コンファレンス1996 講演論文集, pp.162-172 (1996).
- [新井94] 新井 民夫, 木村 久幸, 前田 健太郎, 太田 順, 梅田和昇: “移動ロボット相互の位置・姿勢実時間計測システムの開発”, 日本ロボット学会誌, Vol. 12, No. 3, pp.472-478 (1994).
- [太田99] 太田 順, 山本 正和, 池田 和生, 相山 康道, 新井 民夫: “記憶機能付き画像マーカーによる作業移動ロボットのための環境整備”, 日本ロボット学会誌, Vol. 17, No. 5, pp.670-676 (1999).

## 欧文文献

- [Desai96] J. Desai, C.-C. Wang, M. Zefran, and V. Kumar: "Motion Planning for Multiple Mobile Manipulators", *Proc. 1996 IEEE Int. Conf. Robotics and Automation*, pp.2073-2078 (1996).
- [Khatib99] O. Khatib: "Mobile manipulation: The robotic assistant", *Robotics and Autonomous Systems* 26, pp.175-183 (1999).
- [Wang94] Z.-D. Wang, E. Nakano, T. Matsukawa, "Cooperating Multiple Behavior-Based Robots for Object Manipulation" *Proc. 1994 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp.1524-1531 (1994).
- [Ahmadabadi97] A. M. Nili, E. Nakano: "Task allocation and distributed cooperation strategies in a group of object transferring robots", *Proc. 1997 IEEE Int. Conf. on Intel. Robots and Systems*, pp.435-440 (1997).
- [Hara99] M. Hara, M. Fukuda, H. Nishibayashi, Y. Aiyama, J. Ota, and T. Arai, "Motion Control of Cooperative Transportation System by Quadruped Robots Based on Vibration Model in Walking", *Proc. 1999 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Vol.3, pp.1651-1656 (1999).
- [Stilwell93] D. J. Stilwell, J. S. Bay, "Toward the Development of a Material Transport System using Swarms of Ant-like Robots", *Proc. IEEE Int. Conf. Robotics and Automation*, Vol.1, pp.766-771 (1993).
- [Stilwell94] D. J. Stilwell, J. S. Bay, "Optimal Control for Cooperating Mobile Robots Bearing a common Load", *Proc. IEEE Int. Conf. Robotics and Automation*, Vol.1, pp.58-63 (1994).
- [Kosuge95] K. Kosuge, D. Taguchi, T. Fukuda, M. Sakai and K. Kanitani, "Decentralized Control of Robots for Dynamic Coordination", *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Vol.1, pp.76-79 (1995).
- [Kosuge96] K. Kosuge and T. Oosumi, "Decentralized Control of Multiple Robots Hand-

- dling and Object", *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Vol.1, pp.318-323 (1996).
- [Kosuge99] K. Kosuge and M. Sato: "Transportation of a Single Object by Multiple Decentralized-Controlled Nonholonomic Mobile Robots", *Proc. 1999 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Vol.3, pp.1681-1686 (1999).
- [Koga92] Y. Koga, J.-C. Latombe, "Experiments in Dual-Arm Manipulation Planning", *Proc. 1992 IEEE Int. Conf. Robotics and Automation*, pp.2238-2245 (1992).
- [Koga94] Y. Koga, J.-C. Latombe, "On Multi-Arm Manipulation Planning", *Proc. 1994 IEEE Int. Conf. Robotics and Automation*, pp.945-952 (1994).
- [Ogasawara92] G. Ogasawara, T. Omata, T. Saito, "Multiple Movers Using Distributed Decision-Theoretic Control", *Proc. of JAPAN/USA Symposium on Flexible Automation*, Vol.1, pp.623-630 (1992).
- [Yoshikawa96] T. Yoshikawa, M. Kurisu, T. Kozuka: "Deciding Grasping Positions and Regrasping Action by Cooperating Multiple Mobile Robots", *Proc. 1996 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp.118-124 (1996).
- [Osumi96] H. Osumi: "Cooperative Strategy for Multiple Position-Controlled Mobile Robots", *Distributed Autonomous Robotics Systems* 2, pp.374-385, Springer (1996).
- [Borenstein93] J. Borenstein: "Multi-Layered Control of a Four-Degree-of-Freedom Mobile Robot with Compliant Linkage", *Proc. IEEE Robotic and Automation*, pp. 7-12 (1993).
- [Yuta92] S. Yuta and S. Premvuti: "Coordinating Autonomous and Centralized Decision Making to Achieve Cooperative Behaviors Between Multiple Mobile Robots", *Proc. 1992 IEEE/RSJ Int. Conf. on Intel. Robots and Systems*, pp.1566-1574 (1992).
- [Noreils92] F. R. Noreils: "Multi-Robot Coordination for Battlefield Strategies", *Proc. 1992 IEEE/RSJ Int. Conf. on Intel. Robots and Systems*, pp.1777-1784 (1992).
- [Noreils93] F. R. Noreils: "Toward a Robot Architecture Integrating Cooperation between



- Mobile Robots: Application to Indoor Environment", *Int. J. of Robotics Research*, Vol.12, No.1, pp.79-98 (1993).
- [Caloud90] P. Caloud, W. Choi, J.-C. Latombe, C. L. Pape, and M. Yim: "Indoor Automation with many mobile robots", *Proc. 1990 IEEE Int. Workshop on Intell. Robots and Systems*, pp.67-72 (1990).
- [Shibata96] T. Shibata, K. Ohkawa, and K. Tanie: "Spontaneous Behavior of Robots for Cooperation -Emotionally Intelligent Robot System-", *Proc. 1996 IEEE Int. Conf. on Robotics and Automation*, pp.2426-2431 (1996).
- [Jung99] D. Jung, A. Zelinsky: "An architecture for distributed cooperative planning in a behaviour-based multi-robot system", *Robotics and Autonomous Systems* 26, pp.149-174 (1999).
- [Kuniyoshi94] Y. Kuniyoshi, J. Rieki, M. Ishii, S. Rougeaux, N. Kita, S. Sakane, and M. Kakikura: "Vision-Based Behaviors for Multi-Robot Cooperation", *Proc. 1994 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp.925-932 (1994).
- [Mitsumoto97] N. Mitsumoto, T. Fukuda, F. Arai, and H. Ishihara: "Control of the Distributed Autonomous Robotic System based on the Biologically Inspired Immunological Architecture", *Proc. 1997 IEEE Int. Conf. on Robotics and Automation*, pp.3551-3556 (1997).
- [Fukuda97] T. Fukuda and T. Kaga: "Distributed Decision Making of Dynamically Reconfigurable Robotic System", *Proc. 1997 IEEE Int. Conf. on Intel. Robots and Systems*, pp.1604-1609 (1997).
- [Genovese92] V. Genovese, P. Dario, R. Magni, L. Odetti: "Organizing Behavior and Swarm Intelligence in a Pack of Mobile Miniature Robots in Search of Pollution", *Proc. 1992 IEEE/RSJ Int. Conf. on Intel. Robots and Systems*, pp.1575-1582 (1992).
- [Asama92] H. Asama, K. Ozaki, A. Matsumoto, Y. Ishida, and I. Endo: "Development of Task Assignment System Using Communication for Multiple Autonomous Robots", *J. of Robotics and Mechatronics*, Vol.4, No.2, pp.122-127 (1992).

- [Vidal96] T. Vidal, M. Gallab, and R. Alami: "Incremental Mission Allocation to a Large Team of Robots", *Proc. 1996 IEEE Int. Conf. on Robotics and Automation*, pp.1620-1625 (1996).
- [Brooks96] R. A. Brooks: "A Robust Layered Control System For A Mobile Robot", *IEEE J. of Robotics and Automation*, pp.14-23 (1986).
- [Parker96] L. E. Parker: "On the design of behavior-based multi-robot teams", *Advanced Robotics*, Vol.10, No.6, pp.547-578 (1996).
- [Parker97] L. E. Parker: "L-ALLIANCE: Task-oriented multi-robot learning in behavior-based systems", *Advanced Robotics*, Vol.11, No.4, pp.305-322 (1997).
- [Parker98] L. E. Parker: "ALLIANCE: An Architecture for Fault Tolerant Multirobot Cooperation", *IEEE Trans. Robotics and Automation*, Vol.14, No.2, pp.220-240 (1997).
- [Faltings91] B. Faltings and P. Pu: "Applying Means-ends Analysis to Spatial Planning", *Proc. 1991 IEEE/RSJ Int. Conf. on Intel. Robots and Systems*, pp.80-85 (1991).
- [Hwang92] Y. K. Hwang, and N. Ahuja: "Gross Motion Planning - A Survey", *ACM Computing Surveys*, Vol.24, No.3, pp.219-291 (1994).
- [Yamamoto94] Y. Yamamoto and X. Yun: "Coordinating Locomotion and Manipulation of a Mobile Manipulator", *Trans. Automatic Control*, Vol. 39, pp. 1326-1332 (1994).
- [Kim92] J. H. Kim, H. S. Cho: "Real-time determination of a mobile robot's position by linear scanning of a landmark", *Robotica*, Vol. 10, pp. 309-319 (1992).

## 書籍

- [石田96] 石田亨, 片桐恭弘, 桑原和宏: “分散人工知能”, コロナ社 (1996).
- [美多86] 美多勉: “デジタル制御理論”, 昭見堂 (1986).
- [今野78] 今野浩, 山下浩: “非線形計画法”, 日科技連 (1978).
- [中山94] 中山弘隆, 谷野哲三: “多目的計画法の理論と応用”, コロナ社 (1994).
- [伊理95] 伊理正夫, 今野浩, 刀根薫: “最適化ハンドブック”, 朝倉書店 (1995).
- [大山93] 大山達雄: “最適化モデル分析”, 日科技連 (1993).
- [村上90] 村上 伸一: “画像処理工学”, 東京電機大学出版局 (1990).
- [Latombe91] J.-C. Latombe: “Robot Motion Planning”, Kluwer Academic Publishers (1991).
- [園川96] 園川隆夫, 伊藤謙治: “生産マネジメントの手法”, 朝倉書店 (1996).

## 研究業績



## 学位論文

- 宮田 なつき: “持ち替え動作を伴う複数台の移動ロボットによる大型対象物協調搬送”, 東京大学卒業論文 (1995).
- 宮田 なつき: “移動ロボットによる持ち替え協調搬送”, 東京大学修士論文 (1997).

## 学術論文

- 太田 順, 宮田 なつき, 新井 民夫, 吉田 英一, 倉林 大輔, 佐々木 順: “持ち替え動作を導入した移動ロボット群による大型対象物の搬送制御”, 日本機械学会論文集C編, 63巻, 605号, pp.174-181 (1997).
- 宮田 なつき, 太田 順, 新井 民夫, 相山 康道, 佐々木 順: “車輪型移動ロボット群による持ち替え協調搬送の実現”, 日本機械学会論文集C編, 65巻, 631号, pp.1021-1028 (1999).
- Jun OTA, Tamio ARAI, Yuji YOSHIMURA, Natsuki MIYATA, Eiichi YOSHIDA, Daisuke KURABAYASHI, Jun SASAKI: “Motion Planning of Multiple Mobile Robots by Combination of Learned Visibility Graph and Virtual Impedance”, *Advanced Robotics*, Vol.10, No.6, pp.605-620 (1996).

## 査読付き講演論文

- Jun OTA, Natsuki MIYATA, Tamio ARAI, Eiichi YOSHIDA, Daisuke KURABAYASHI, Jun SASAKI: “Transferring and Regrasping a Large Object by Cooperation of Multiple Mobile Robots”, *Proc. 1995 IEEE/RSJ Int. Conf. on Intel. Robots and Systems*, Vol.3, pp.543-548 (1995).
- Natsuki MIYATA, Jun OTA, Tamio ARAI, Eiichi YOSHIDA, Daisuke KURABAYASHI, Jun SASAKI, Yasumichi AIYAMA: “Cooperative Transport with Regrasping of Torque-Limited Mobile Robots”, *Proc. 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.304-309 (1996).

- Natsuki MIYATA, Jun OTA, Yasumichi AIYAMA, Jun SASAKI, Tamio ARAI: “Co-operative Transport System with Regrasping Car-like Mobile Robots”, *1997 IEEE/RSJ International Conference on Intelligent Robots and Systems* pp.1754-1761 (1997).
- Natsuki MIYATA, Jun OTA, Yasumichi AIYAMA, Tamio ARAI: “Real-time Task Assignment for Multi-agent Robots - Application to Cooperative Transportation Task -”, *Proc. 1999 IEEE Int. Conf. on Systems, Man and Cybernetics*, Vol.4, pp.761-768 (CD-ROM) (1999).
- Natsuki MIYATA, Jun OTA, Yasumichi AIYAMA, Tamio ARAI: “Real-time Task Assignment for Cooperative Transportation by Multiple Mobile Robots”, *Proc. 1999 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.1167-1174 (1999).
- Jun Sasaki, Atsushi Yamashita, Natsuki Miyata, Yasumichi Aiyama, Jun Ota and Tamio Arai: “Constraint of Contacting Points in Cooperative Handling”, *Proc. 1998 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.1425-1430 (1998).

## 口頭発表

- 太田 順, 宮田 なつき, 佐々木 順, 吉田 英一, 倉林 大輔, 新井 民夫: “持ち替え動作を伴う複数台の移動ロボットによる大型対象物協調搬送制御”, 1995年度精密工学会春季大会学術講演論文集, pp.593-594 (1995).
- 宮田 なつき, 太田 順, 佐々木 順, 吉田 英一, 倉林 大輔, 新井 民夫: “持ち替え動作を伴う複数台の移動ロボットによる大型対象物の協調搬送”, 日本機械学会ロボティクス・メカトロニクス講演会'95講演論文集, pp.800-803 (1995).
- 宮田 なつき, 太田 順, 佐々木 順, 吉田 英一, 倉林 大輔, 新井 民夫: “複数の移動ロボットの協調による搬送作業の実現 (第4報: ロボットの速度・加速度を考慮した持ち替え戦略)”, 第13回 日本ロボット学会学術講演会, pp.851-852 (1995).
- 宮田 なつき, 太田 順, 相山 康道, 佐々木 順, 新井 民夫: “移動ロボットによる持ち

替え協調搬送 第1報: 動作計画アーキテクチャ”, 1997年度精密工学会秋季大会学術講演論文集, pp.447 (1997).

- 宮田 なつき, 太田 順, 相山 康道, 佐々木 順, 山下 淳, 新井 民夫: “移動ロボットによる持ち替え協調搬送 第2報: 協調搬送実験”, 1997年度精密工学会秋季大会学術講演論文集, pp.448 (1997).
- 宮田 なつき, 太田 順, 相山 康道, 新井 民夫: “機能分化を伴う協調搬送”, 日本機械学会ロボティクス・メカトロニクス講演会'98講演論文集, (by CD-ROM) 2A12-6 (1998).
- 宮田 なつき, 太田 順, 今西 順一, 相山 康道, 新井 民夫: “機能分化を伴う協調搬送計画”, 第16回 日本ロボット学会学術講演会予稿集, pp.111-112 (1998).
- 宮田 なつき, 川端 邦明, 浅間 一, 小菅 一弘, 新井 民夫: “異方性を利用した可変コンプライアンス型協調作業システム”, 1999年度精密工学会春季大会学術講演論文集, pp.45 (1999).
- 宮田 なつき, 太田 順, 相山 康道, 新井 民夫: “移動ロボット協調作業のための作業割り当て手法”, 日本機械学会ロボティクス・メカトロニクス講演会'99講演論文集, (by CD-ROM) 1P1-30-039 (1999).
- 宮田 なつき, 太田 順, 新井 民夫: “機能分化型群ロボットによる協調搬送作業”, 第17回 日本ロボット学会学術講演会予稿集, pp.381-382 (1999).
- 太田 順, 吉田 英一, 倉林 大輔, 佐々木 順, 宮田 なつき, 新井 民夫, 大隅 久: “複数の移動ロボットの協調による搬送作業の実現 (第3報: 持ち替え動作を含んだ障害物回避行動の実現)”, 第12回日本ロボット学会学術講演会予稿集, pp.395-396 (1994).
- 太田 順, 宮田 なつき, 佐々木 順, 吉田 英一, 倉林 大輔, 山下 淳, 新井 民夫: “複数の移動ロボットの協調による搬送作業の実現 (第5報: センシングエリアを考慮した群れ作り搬送の実現)”, 第13回日本ロボット学会学術講演会予稿集, pp.863-864 (1995).
- 相山 康道, 西林 ひなた, 原 光博, 宮田 なつき, 太田 順, 新井 民夫: “2台の脚ロボッ

トによる協調搬送のための自律的制御系の構築”, 第15回日本ロボット学会学術講演会予稿集, pp.471-472 (1997).

- 佐々木 順, 山下 淳, 宮田 なつき, 太田 順, 相山 康道, 新井 民夫: “移動ロボットの協調による物体操作 (第2報: 移動ロボット間の力学的拘束を利用した相対位置センシング)”, 第15回日本ロボット学会学術講演会予稿集, pp.461-462 (1997).



