

## 群ロボットによる異種作業割り付け型協調搬送

非会員 宮田 なつき (東京大学)  
 非会員 太田 順 (東京大学)  
 非会員 新井 民夫 (東京大学)  
 正員 浅間 一 (理化学研究所)

## Cooperative Transport by Multiple Mobile Robots Associated with Task-assignment

Natsuki Miyata, Non-member, Jun Ota, Non-member, Tamio Arai, Non-member (The University of Tokyo), Hajime Asama, Member (RIKEN)

This paper deals with a task-assignment architecture for cooperative transport by multiple mobile robots in unknown environment. The architecture should satisfy three features: to deal with the variety of tasks in time and space; to deal with a large amount of tasks; and, to decide behavior in real-time. The authors propose the following approach: (1) We consider the unit of task (task instance) as the form that should be done in short time by one robot. Based on environmental information, task instances are dynamically generated using task templates; (2) Priority of task instances are evaluated dynamically based on the number of robots and the configuration in the work space. In addition, it is avoided to generate too many task instances by suppressing movement; (3) Main part of the architecture consists of two real-time planner: priority-based task-assignment planner solved by using a linear programming method and motion planners based on short-time estimation. Effectiveness of the proposed architecture is verified by a cooperative transport simulation in unknown environment.

キーワード：実時間，タスク割り当て，移動ロボット，協調搬送，マルチロボット

## 1. 序 論

昨今，大型機械の組み立てや，建設現場における資材移動，極限環境である原子力プラントにおけるメンテナンス作業など，多種多様な大型物体を扱う作業の自動化の要求は大きい。これに対し，小型移動ロボットを複数台投入することで，様々な形状・重量の物体に応じて柔軟に作業システムの組み換えが可能であることから，こうした作業への応用が期待されている。そのため数多くの研究がなされてきたが，そのほとんどは，いかに対象物を目標軌道に追従させるかという制御問題に焦点を当てたものであった<sup>(1)(2)(3)</sup>。しかし，作業の実現が求められるのは，作業員や障害物が搬送経路上に存在するなど，ロボットのためだけの作業領域が確保されているわけではなく，事前にすべての環境情報を把握することが不可能な環境である。移動ロボットによる協調作業をこのような“開かれた”環境で行わせる場合，物体をハンドリングするだけでなく，効率の良い外界センシング，作業を妨害する障害物の除去など，作業領域の観測も含め，相互関係の深い様々な作業（これを本論文ではタスクと呼ぶ）を，状況に応じて実時間でロボットに

割り当て，その動作を計画する必要がある（図1）。

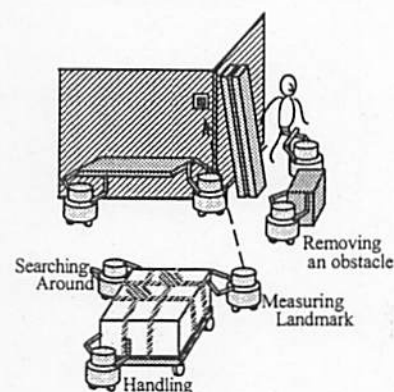


図1 開かれた環境における協調搬送作業

Fig. 1. Cooperative Transport in an Open Environment

タスク割り当て手法を提案するにあたって考慮すべき課題は3つある。まず，ロボットは作業を始める段階では環境についての情報を十分に取得しておらず，タスクはロボット群の行動した結果生成する。このような作業における“タスクの性質”という観点から，タスク割り当て器には以下

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が、それらの全てが常に必要というわけではない。走行精度が悪化すると誤差補正のためにランドマーク等の計測、搬送経路を占領する障害物を発見するとその除去など、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、それまでまったく情報を得ていない領域の監視とすでに通過した領域の監視など、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ことから、このような状況を反映した形で、その時々に必要なタスクを素早く抽出する必要がある。

### (2) ロボット群の処理能力を超える大量のタスク

(1)でとりあげたタスクの変化に伴い、タスクの総量は変化し、時に、ロボット群によってその瞬間に処理可能な量を越えてタスクが存在するため、適切な順序でタスクを選択し実行する必要がある。まず、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満たす必要がある。さらに、その協調動作を実行可能なだけの台数を確保して実行可能なタスクから、優先的にタスクを選択する必要がある。また、このように適切な選択順序をとることが可能な場合でも、タスクの増加に処理が追い付かない状態ではシステムの破綻を招くため、タスクが必要以上に増加して溢れることを可能な範囲で防止する必要がある。

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、

### (3) 実時間での動作決定

を実現する必要がある。

上記の問題に対するアプローチの一つとして、通信に基づくロボット間の交渉によりタスクを割り当てる方法がある<sup>(4)(5)(6)(7)</sup>が、決定まで時間がかかり、実時間での動作決定には向いていない。また、行動ベースロボットを用いた分散的割り当て手法が提案されてきた<sup>(8)(9)(10)(11)</sup>が、タスク間の相互関係の記述が困難であり、本課題に直接適用することはできない。割り当て問題を扱ったものとしてジョブショップスケジューリングに関する研究が数多くなされているが(例えば<sup>(12)</sup>)、頻繁なタスクの変動に対応したものはない。

そこで本論文では、上記(1)(2)(3)を満たすタスク割り当て手法を提案する。2章では提案するタスク割り当てアーキテクチャの概要を示す。4章において代表的なタスクを実装、5章において協調搬送シミュレーションを行い、提案手法の有効性を示す。6章で結論を述べる。

なお、以下の議論を進めるに当たり、

- ロボットは、ある瞬間において、ある方向のある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。

## 2. タスク割り当て

〈2.1〉提案アーキテクチャ 周辺環境に変動が起こり得る場合、(1)に示したように必要なタスクが変化するため、目標地点に至るまでのすべての動作推測に基づく最適な解を時間をかけて計画しても意味がない。したがって、その場で得られた情報に基づき高速にタスクを割り当てその動作を決定する必要がある。

ここで、目標を達成するまでの時間を短いサンプリング時間で区切り、割り当てるタスクを“1～数サンプリング時間で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)で捉えたとする。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)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。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”として高速に計算可能となる。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である。また変動要素は二つあり、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。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。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サンプリング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ことが期待出来る。

そこで、本論文では図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) 各ロボットによる実行

(Step 7) 割り当てられなかったインスタンスの消去  
という一連のプロセスを一定サンプリング時間毎に繰り返すというものである。必要なものは次の周期で環境情報をもとに再度生成される。

設計者は、環境情報に基づき次のサンプリング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の“タスクテンプレート”と、その間の各ロボットの具体的な動作を計画可能な“動作計画器”を、タスク毎に準備する必要がある。前者の内容については〈3.2〉節で説明する。後者について、例えば物体のハンドリングタスクには、筆者らの提案してきた動作

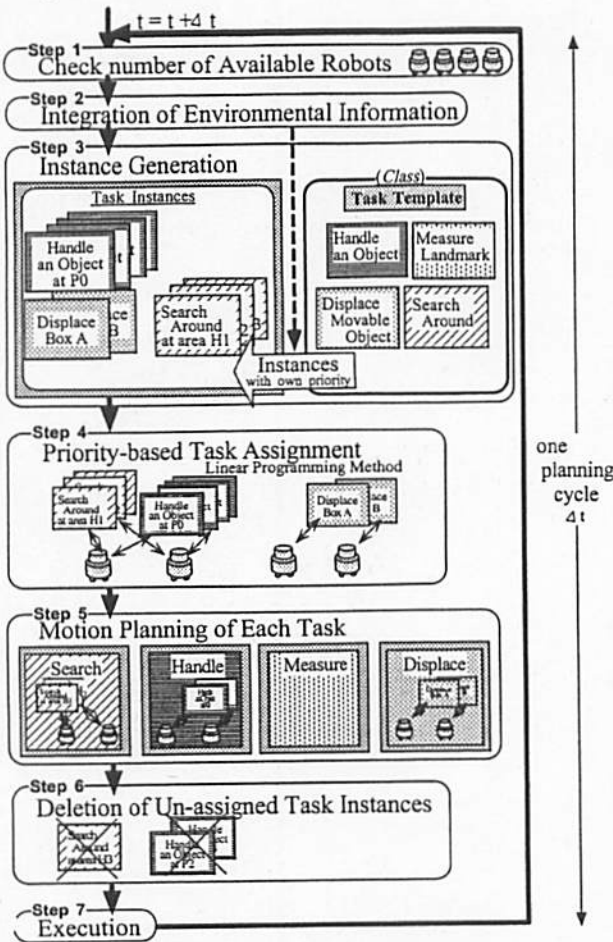


図2 計画プロセスの概要  
Fig. 2. Profile of the Planning Process

計画器<sup>(13)</sup>が適用できる。ただし、タスク毎の動作計画器の間には、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。

〈2.2〉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。

まず“(1) タスクの動的な変化”に対し、一定サンプリング時間毎に、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を表現可能である。

次に、“(2) ロボット群の処理能力を越える大量のタスク”に対し、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を評価して割り当て、未割り当てのもの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である。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。

最後に、“(3) 実時間での行動決定”に対し、必要な作業を短時間に1台のロボットが行う作業というタスクインス

タンスとして生成したため優先度というパラメータを用いていわゆる“割当問題”として解け、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。

### 3. テンプレートと割り当てアルゴリズム

〈3.1〉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様々なロボットの行動をタスクとして扱う。例えば、(t1) 搬送物体のハンドリング、(t2) 既知ランドマークの計測による自己位置の同定、(t3) 周辺の監視、(t4) 可動障害物の除去、などをタスクとみなす。ロボットはこれらのタスクを“移動”あるいは“観測”の能力を運用することで達成するが、どの能力を必要とするかによってタスクは

- group 1 “移動”のみ (t1),
- group 2 “観測”のみ (t2, t3),
- group 3 “移動”および“観測”両方 (t4)

の3グループに分類することが出来る。この時、ロボット1台は group1 および group2 のタスクを同時に実行可能である。

〈3.2〉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サンプリング時間にどのような動作(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のものである。記述内容を図3にまとめる。インスタンス生成の際には、各サンプリング周期で異なる“割り当て可能なロボット”の台数  $n_{freerob}$  が参照される。テンプレートの各項目の具体的な設定方法については4章のタスクの実装の部分で例を示して説明する。

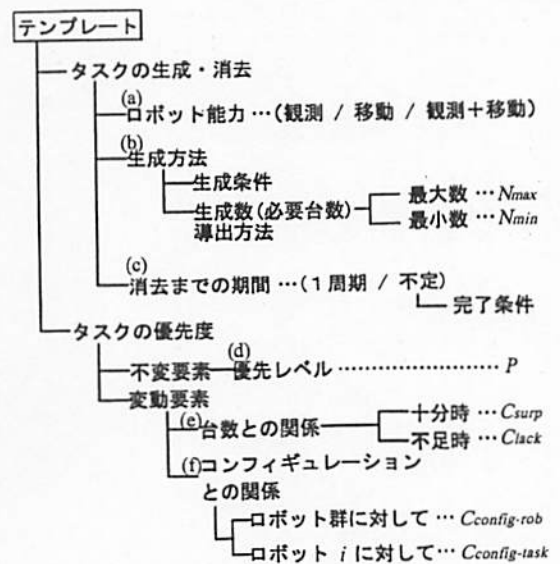


図3 タスクテンプレート  
Fig. 3. Task Template

〈3・3〉優先度を用いたタスク割り当てアルゴリズム  
 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し実時間で解くことが可能である。一方、〈3・1〉節で示したように、タスクによって必要なロボットの能力が異なり、同時に2種類のタスクを割り当てる場合もある。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。両能力を要するタスクの場合は、同じロボットが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。これらの変数のとる値とその意味は以下の通りである。

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

タスクインスタンスがロボット台数を越えて大量に存在する際に、タスクテンプレートで定義したタスクの優先度のうち、不変要素である優先レベルが高いものは優先的に実行しておく必要がある。また、優先レベルが低い場合でも、台数の過不足に関わらず実行に意味のあるタスクはなるべく多くのロボットに割り当て消化しておく必要がある。そこで、優先レベルの低い方のタスクから、実行可能になるまでそのタスクに関する制約条件を緩和し、式1で表される評価関数  $PI$ の最適化を行う、というアルゴリズムを提案する。ここでは分枝限定法を用いて解く<sup>(14)</sup>。

$$PI = \sum_i^{n_{instance}} \sum_j^{n_{freerob}} w_{ij} X_{ij} \dots \dots \dots (1)$$

$n_{instance}$ : 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。

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

$$C_{num} = \begin{cases} C_{surp} & n_{freerob} > N_{min} \text{ の場合} \\ C_{lack} & \text{上記以外の場合} \end{cases}$$

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)

$N_{min}$ : タスクインスタンスの最小生成数 (図3の(b))  
 これにより、優先レベルの高いタスクは拘束条件の効果によって確実に割り当てられる。また低いタスクでも、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  $C_{config}$ の値が大きなものについては、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。したがって、ロボット台数を越えた大量

のタスクが存在する場合にも、適切な順序で割り当てる事が可能となる。

#### 4. タスクの実装

本章では、協調搬送シミュレーションのために今回実装した“周辺監視”、“可動障害物除去”、“物体ハンドリング”の3種類のタスクについて説明する。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表1にまとめる。

3つのタスク間の実行順序関係を表す“優先レベル”(表1の(d))は、次のように決定した。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”により前に発見されている必要がある。したがって優先レベルは、表1に示す順序となる。

以下、各タスクの説明を行い、特に〈4・2〉節の“障害物除去”タスクの説明において、優先レベル以外の各項の設定方法について詳しく説明する。

表1 実装タスクのテンプレート内容  
 Table 1. Task Templates for Implemented Tasks

	周辺監視	障害物除去	ハンドリング
(a) 能力	観測	観測+移動	移動
生成条件	情報が一定期間以上古い	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) $N_{min}$ $N_{max}$	下配+ $N_{freerob}$ 上記を満たすセル集合を成す部分集合の数	障害物の個数 障害物の個数	$N_{freerob}$ 2
(c) 期間 完了条件	1周期 —	不定 物体が搬送経路から出て数に接触した時	1周期 —
(d) $P$	1	2	3
(e) $C_{surp}$ $C_{lack}$	1.0 8.0	1.0 8.0	4.0 1.0
(f) $C_{config}$	$\alpha \times$ (ロボットとタスクを生成する領域重心との距離) + 領域の平均タイムー値	$\alpha \times$ (ロボットとタスクを生成する障害物との距離)	$\alpha \times$ (ロボットとタスクを生成する把持位置との距離)

ただし  $\alpha = \begin{cases} 1: \text{間に他の障害物なし} \\ 10: \text{間に他の障害物あり} \end{cases}$

〈4・1〉周辺監視タスク 環境が変動する場合、ある領域に関する情報は新しいほど信頼でき望ましいと考えられる。タスクインスタンスを生成するためには、各サンプリング周期において監視すべき領域を求め、これを1台のロボットで監視可能な部分領域に分割する必要がある。そこでまず、作業環境を小さい矩形セルに区切り、前回監視されてからの時間をセル毎に計上し情報の古さを表現する。これを用い搬送対象物周辺に限定して一定以上情報の古いセルの集合(これを監視必要領域と呼ぶ)を求める。該当するセル集合を1台のロボットが1周期で監視可能と推定される部分集合に分割し、その部分集合毎にインスタンスを生成する(図4)。

〈4・2〉障害物除去タスク 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。タスク消去

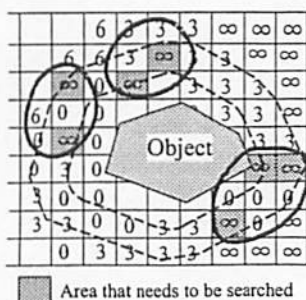


図4 “周辺監視”タスクインスタンスの生成

Fig. 4. How to Generate Instances in the “Search Around” Task

とみなし生成する。

物体ハンドリングの動作計画器では、原則として対象物は定速で動作させることにするが、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) 前サンプリング周期の“周辺監視タスク”において、監視必要と見なされたにも関わらず監視が完了しなかった領域（セル）が一つでも残っている場合。

## 5. タスク割り当て手法評価のための協調搬送シミュレーション

〈5.1〉シミュレーション内容 提案したタスク割り当て手法の有効性を検証するため、未知障害物存在下における協調搬送シミュレーションを、構成台数の異なる2つのロボット群について行った。ロボット群Aは4台の、ロボット群Bは2台のロボットから成る。その他のシミュレーション条件および環境は両群で共通であり、これを表2に示す。ただしロボットは、センシングエリア内に含まれる障害物の形状により、それが可動障害物かどうかを判断する。

提案アーキテクチャは、計画を行う主体により、

- ロボット群の中の1台が他のロボットの取得した環境情報を収集・統合し、これを元に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、互いに取得した環境情報を交換・統合し、これを元に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

という形態をとり得るが、ここでは前者の形態で実装を行った。

表2 シミュレーション条件

Table 2. Simulation Conditions

the shape of the object	a 1.2 × 0.6-[m] rectangle
the shape of the movable object	a 0.6(w)×0.4(d)-[m] box
the shape of the robots	circle with a 0.2-[m] radius
sensing area of the robots	a sector with a 1.5-[m] radius and a 20-degree interior angle
maximum speed (handling)	0.1 [m/s]
maximum speed (running alone)	0.2 [m/s]
division of work space for sensing	a 0.2×0.2-[m] rectangle
sampling time	3.0 [s]

〈5.2〉シミュレーション結果（時間経過） シミュレーション結果を図6および図7に示す。各図における扇形は、各ロボットのセンシングエリアを表す。また、濃灰色の矩形の集合は、周辺監視タスクの生成等に用いる監視必要領域を表す（詳細は〈4.1〉節を参照）。図6(a)および図7(a)は各ロボット群の初期状態であり、右方の小さな丸が対象物の目標位置を表している。

4台から成るロボット群Aは、搬送作業を開始し、しばらく

までの期間について、一旦ある物体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“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。タスクの優先度の変動要素のうち、ロボット台数との関係に基づく優先度に関しては、障害物個数に対しロボット台数が不足していたとしても、1つでも多くの障害物を割り当てる必要があることから、 $C_{surp} < C_{lack}$ となる。具体的な値は、この設計指針をもとに試行錯誤的に決定した（他のタスクのこのパラメータについても、同様に試行錯誤的に決定した）。また、コンフィギュレーションとの関係に基づく優先度  $C_{config}$  については、ある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ことが可能であるため好ましく、また、ロボット群により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ことからこのような算出式になっている。

なおタスク割り当て手法に焦点をおくため、動作計画は単純な戦略をとり、最も近い壁に接触するまで、搬送対象物の軌道に垂直に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、壁に接触する最終目的地が重ならないよう経路を調整する（図5）。

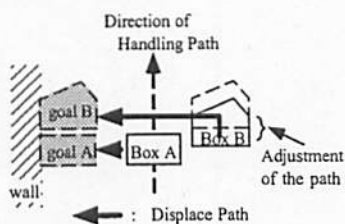


図5 “障害物除去”タスクにおける除去経路

Fig. 5. Displacing Path in the “Displace Movable Objects” Task

〈4.3〉物体ハンドリングタスク ロボットは手先ではモーメントは発生させず平面内2次元の十分な力を生成し、台車に載せられた搬送物体を、押すまたは引くことでハンドリング動作を実現することを想定する。タスクの生成・消去に関して、対象物上の各ロボットの把持位置における、1サンプリング時間内のハンドリング動作をインスタンス

く周辺を監視しながら物体のハンドリングを続けた後、障害物を検知し、48秒に4台のうち2台のロボットが対象物を離れ、“障害物除去”へと向かう(図6(b))。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(図6(c))。除去作業を終えたロボットは再び“ハンドリング”作業に割り当てられ対象物のハンドリングにそれぞれ戻り(図6(d)(e))、最終的には4台のロボットのハンドリングによりゴールに到達する(図6(f))。

2台から成るロボット群Bは、搬送作業を開始した後、同様に障害物を検知するが、Aの場合と異なり54秒に2台ともが対象物を離れ“障害物除去”へと向かう(図7(b)(c))。1台のロボットがハンドリングに戻っただけでは実際に動くことが出来ないためその場に待機し(図7(d))、もう1台が戻ってきた時点でハンドリング動作を再開し(図7(e))、184秒でゴールに到達している(図7(f))。

図8にロボット群Aの作業においてタスクインスタンスが生成された様子を示す。

シミュレーション結果より、以下のことが言える。

- ロボット群Aは139秒、ロボット群Bは184秒と、目的達成までの動作時間は異なるが、両群ともに目標を達成する動作が計画されている。
- 計算時間は、SPARCstation20を用いて計算を行ったところ、ロボット群A(4台)の場合、各周期平均で約0.11秒、ロボット群B(2台)の場合約0.07秒であった。
- 図8より、作業の進行に伴い異なる質・量のタスクインスタンスが生成されており、必要な作業が変化の様子が分かる。
- ロボット群Bは必要な作業に対し台数が少ないため、図7(b)のように、一旦あるタスク(ハンドリングおよび監視)を諦め他のタスク(除去作業)を先に行うなど、タスクの選択順序にロボット台数との関係項が大きく影響する様子が顕著に表れた。
- ロボット群Aでは、図8(3)(4)のように、台数としてはハンドリング可能であるにも関わらず、前2台のロボットが先に周辺監視を行ってから障害物除去を割り当てられるなど、台数との関係よりもコンフィギュレーションとの関係に基づく実行可能かどうかの判断が強く影響した結果が見られた。
- 両群ともに、監視しつつハンドリングのために移動しては残留領域の監視のために停止、完了後に再度移動、という動作が頻繁に見られた(例えば図8(1)から(2))。このように移動を抑制することで、見落としにより監視必要領域が増加し続けることはなかった。

以上より、提案アーキテクチャにより、タスクの種類や質が変動し、時にロボット群の1サンプリング時間内に実行可能な量を超えてタスクが存在する未知環境においても、時間経過の中で、割り当て可能な台数に応じた適切な順序でタスクが選択され全体としての目標を達成する動作を計画可能であることが確認された。

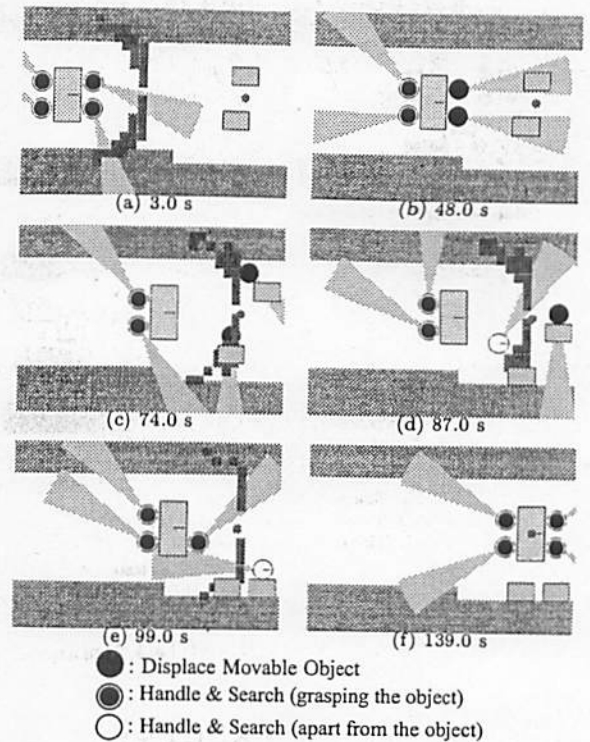


図6 シミュレーション結果-ロボット4台の場合  
Fig. 6. Simulation Results-4 Robots

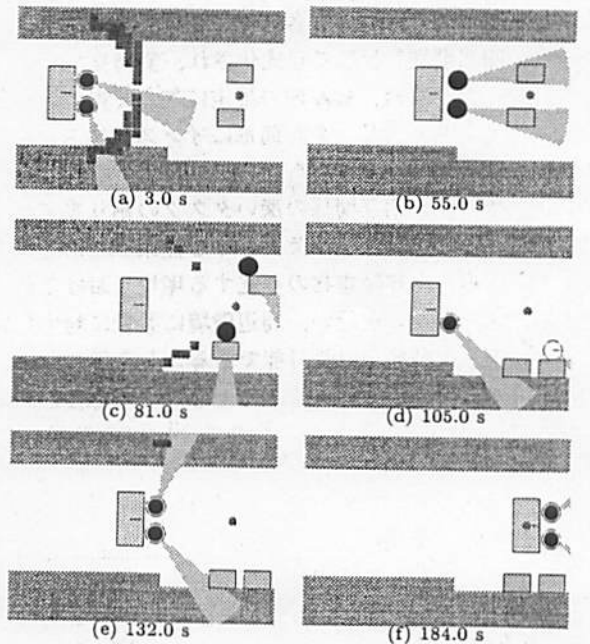


図7 シミュレーション結果-ロボット2台の場合  
Fig. 7. Simulation Results-2 Robots

## 6. 結 論

本論文では、複数台の移動ロボットが周囲の環境に応じて必要な種々のタスクを動的に割り付け協調作業を行うためのタスク割り当て手法を提案した。

- 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、一定サンプリング時間毎に環境情報

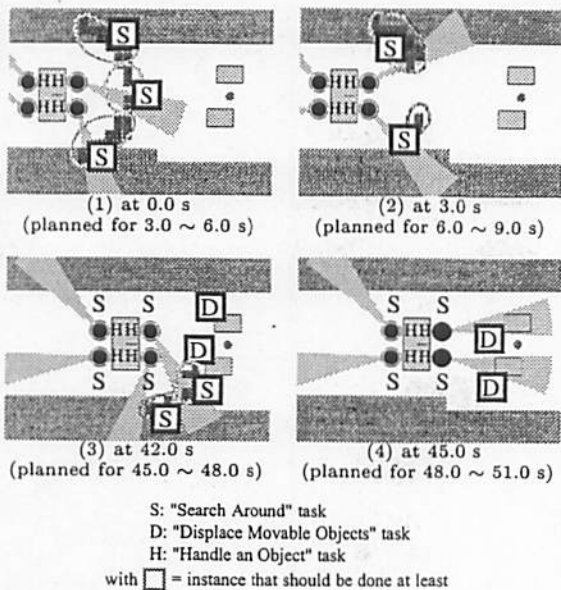


図8 動的に生成されたタスクインスタンス  
 Fig. 8. Dynamically Generated Task Instances

に基づきタスクインスタンスとして生成

- ある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

の2つを行うことにより、各瞬間におけるタスク割り当て問題が“割当問題”として定式化され、実時間での計画が可能となった。また、短期間の推定に基づく動作計画器と組合せ、一定サンプリング時間毎に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、動的な変化に対応し、実行順序など相互関係の深いタスクの割り当てが時系列の中で解決可能なアーキテクチャを提案した。その有効性検証のため、未知障害物の存在する環境における協調搬送シミュレーションを行い、周辺環境に柔軟に対応し搬送目的を達成する動作が計画可能であることを示した。

最後に本研究の一部は(財)河上記念財団の補助を受けた。(平成11年12月17日受付, 同12年02月29日再受付)

## 文 献

- (1) 小菅一弘, 他: “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, 87/95 (1998).
- (2) 橋本雅文, 他: “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, 886/893 (1995).
- (3) D. J. Stilwell, et al. "Toward the Development of a Material Transport System using Swarms of Ant-like Robots", *Proc. IEEE Int. Conf. Robotics and Automation*, 766/771 (1993).
- (4) 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, 79/98 (1993).
- (5) P. Caloud et al.: "Indoor Automation with many mobile robots," *Proc. 1990 IEEE Int. Workshop on Intel. Robots and Systems*, 67/72 (1990).
- (6) H. Asama et al.: "Development of a Task-assignment System Using Communication for Multiple Autonomous Robots," *J. of Robotics and Mechatronics*, vol.4, No.2 122/127 (1992).

- (7) T. Vidal et al.: "Incremental Mission Allocation to a Large Team of Robots," *Proc. 1996 IEEE Int. Conf. on Robotics and Automation*, 1620/1625 (1996).
- (8) L. E. Parker: "ALLIANCE: An Architecture for Fault Tolerant Multirobot Cooperation", *IEEE Trans. Robotics and Automation*, Vol.14, No.2, 220/240 (1997).
- (9) T. Shibata et al.: "Spontaneous Behavior of Robots for Cooperation -Emotionally Intelligent Robot System-", *Proc. 1996 IEEE Int. Conf. on Robotics and Automation*, 2426/2431 (1996).
- (10) D. Jung, et al.: "An architecture for distributed cooperative planning in a behavior-based multi-robot system", *Robotics and Autonomous Systems* 26, 149/174 (1999).
- (11) Y. Kuniyoshi et al.: "Vision-Based Behaviors for Multi-Robot Cooperation", *Proc. 1994 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 925/932 (1994).
- (12) 関川隆夫, 他: “生産マネジメントの手法”, 朝倉書店 (1996).
- (13) 宮田なつき, 他: “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, 1021/1028 (1999).
- (14) 大山達雄: “最適化モデル分析”, 日科技連 (1993).

宮田 なつき (非会員) 1972年5月18日生。1995年3月東京大学工学部精密機械工学科卒業。1997年3月東京大学大学院工学系研究科精密機械工学専攻修士課程修了。同年, 同大学院博士課程に進学, 現在に至る。複数台の移動ロボットの協調動作の研究に従事。日本ロボット学会, 精密工学会の学生会員。



太田 順 (非会員) 1965年2月19日生。1989年3月東京大学大学院工学系研究科精密機械工学専攻修士課程修了。同年新日本製鉄(株)入社。1991年東京大学工学部助手。同講師を経て1996年東京大学工学系研究科助教授。1996~1997年Stanford大学客員研究員。複数台の移動ロボットの計画, ロボットのための環境設計の研究に従事。IEEE, 日本ロボット学会などの会員。博士(工学)。



新井 民夫 (非会員) 1947年8月4日生。1970年東京大学工学部精密機械工学科卒, 1977年同博士課程修了, 工学博士。1987年東京大学工学部精密機械工学科教授。1979年英国エディンバラ大学人工知能学科研究員。産業用ロボット言語の標準化活動を推進。自動組立, 移動ロボットなどの研究に従事。IEEE, CIRP, 精密工学会, 日本ロボット学会などの会員。



浅間 一 (正員) 1959年1月18日生。1984年東京大学大学院工学系研究科修士課程修了。1986年理化学研究所化学工学研究室研究員補。生化学システム研究室研究員を経て, 現在工学基盤研究部技術開発促進室長。工学博士(東京大学)。自律分散型ロボットシステム, 複数移動ロボットの協調の研究等に従事。1995年日本機械学会ロボメック賞等受賞。IEEE, 日本ロボット学会などの会員。

