

自律移動ロボット群を用いた協調行動の実験支援システム

宇津宮 孝一 副島 匡暢 肥川 宏臣 児玉 利忠 吉田 和幸
大分大学工学部知能情報システム工学科

マルチロボットシステムの研究を促進するために、そのテストベッドとして機能する実験支援システムを開発した。このシステムは、Gung-Ho（「協力しよう」の意味）と呼ばれ、自律移動ロボット群の開発とこれらの協調行動の基本概念の試行や評価を支援する。各移動ロボットは対応するワークステーションにより遠隔制御され、システムはUNIX環境が提供する豊富な機能やツールを利用することができる。本論文では、協調行動の実験を支援するGung-Hoシステムの設計と実現およびその有効性などについて述べる。

1 はじめに

協調行動をする複数ロボットからなるシステム（CMARS（Cooperative Multi-Agent Robotic System）と略）は、人工知能やロボティクスなどの分野において、さまざまな概念を試行するための有用な環境を提供する[1]。CMARSにおいては、各ロボットは自律動作をしながらも、協調して共通の目標を達成しなければならない。ロボットがどのようにして協調行動をするかということは、非常に重要で、興味をそそる研究課題である。

これまでのCMARSの研究は、特定の応用分野に焦点を絞ったものか、あるいは机上でのシミュレーションによる概念的なものが多い[2]。その原因は、適当なハードウェアの製作が容易でない、あるいはリアルタイムに近い状況での実験が困難であることなどに起因する。事実、完全なマルチロボットシステムは非常に複雑で、その開発には多大な労力と時間を要してきた。

そこでわれわれは、Gung-Ho（「協力しよう」の意味）と呼ばれるCMARSのための実験支援システムを開発した[3]。このシステムは、自律移動ロボット群の開発とこれらによる協調機構の解明のための実験を支援するものである。Gung-Hoシステムで

は、各移動ロボットは、バックエンドコンピュータであるUNIXワークステーションにより制御されるので、UNIX¹が提供する並行処理機能、ネットワーク機能、Xウィンドウシステム²や共有メモリなどの機能や開発ツールを利用することができる。これらの機能やツールは、CMARSの開発には極めて有用である。

本論文は、まずGung-Hoシステムの設計と実現について述べ、次にこれを用いた予備実験と協調行動問題の考察を行うことにより、このシステムがCMARSによる協調機構の実験支援システムとして有効であることを示す。

2 実験支援システムの設計

ここでは、実験支援システムの中で実現しなければならないCMARSの形態や制御方式について考察する。

2.1 システムの形態

マルチロボットシステムの形態は、主として集中型と分散型の2つに分類される。集中型システムは、ロボット数が少なく、その環境があらかじめよく知られている場合には、効率的な制御方法を提供できる。しかしながら、環境の複雑さが増せば、この方式の実現

A Testbed for Studying Cooperative Behavior Using Autonomous Mobile Robots

by Kouichi UTSUMIYA, Masanobu SOEJIMA, Hiroomi HIKAWA, Toshitada KODAMA and Kazuyuki YOSHIDA (Department of Computer Science and Intelligent Systems, Oita University)

¹ UNIXオペレーティングシステムはAT&Tが開発し、ライセンスしている。

² Xウィンドウシステムはマサチューセッツ工科大学の登録商標である。

は非常に困難である。一方、分散型システムは、中心となる統率者をもたず、各ロボットは独立して動作する。この方式は、各ロボットが各自の判断で行動する（自律動作）ので、全体の統率や制御は容易でないが、柔軟性は高い。

CMARSは分散型システムに属する。分散システムに要求される最も重要な機能は、ロボット間の通信である。各ロボットは自分の行動を決めるために、センサ情報のようなさまざまな情報をお互いの中で交換しなければならない。

2.2 ロボットの制御方式

従来の自律移動ロボットの制御方式は、知覚、モデル化、計画などの機能単位から構成されていた。このような方式の移動ロボットシステムは、正確な世界モデルと経路計画を基本とする [4]。しかしながら、世界知識と経路計画は集中的な管理と処理を必要とするので、分散型制御方式による実現は容易ではない。

Brooksは、機能単位ではなく動作単位とでもいうべきものから構成される包括アーキテクチャ (Subsumption Architecture) の概念を提案した [5]。この動作単位はエージェント (Agent) と呼ばれる。エージェントが実行する個々の作業は、行動 (Behavior) とも呼ばれている [6]。このようなモジュール化は非常に柔軟性が高いソフトウェア構成法である。エージェントの全体のつながりを再構成することなく、必要に応じて行動の挿入や削除が容易にできる。包括アーキテクチャでは、エージェントは独立したプロセスとして動作し、関連情報の交換を大域的に行う。それゆえ、エージェント間の並列処理機能と通信機能は、システムの基本機能として提供されなければならない。

ソフトウェアアーキテクチャは頑健でなければならない。さらに、新たな機能を導入できるように柔軟でなければならないし、デバッグするのに容易でなければならない。その結果、ロボット群全体と個々のロボットの制御方式として、分散制御方式を選択した。

2.3 システム開発ツール

ロボットシステムを開発するためのツールとして、いくつかのロボット言語も提案されているが、高水準

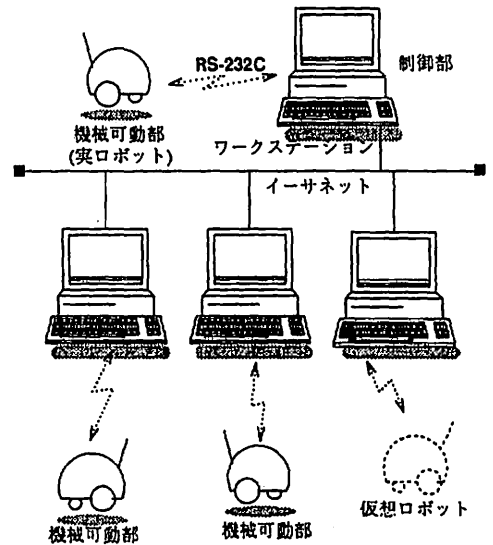


図 1: Gung-Hoシステムの構成

言語の使用はロボットシステムのプログラミングを容易にするものと思われる。しかしながら、CMARSのテストや評価を容易にするための理解しやすいテストベッドには、これらの言語ばかりでなく、さまざまなライブラリ、ロボットのシミュレータやモニタなどのソフトウェア開発ツールやデバッグツールが備えられていなければならない。シミュレータは、導入する新しいアルゴリズムや方式を事前に試行するのに有効である。モニタは、ロボットの位置などの種々の情報やロボットの状態などを追跡する。シミュレータとモニタを組み合わせれば、マルチロボットシステム開発の初期段階において、テストとデバッグの強力なツールを提供することができる [2]。

3 Gung-Hoシステム

Gung-Hoシステムの構成を図 1に示す。各機械可動部 (移動ロボット本体) はバックエンドコンピュータ上に実現された制御部によって遠隔制御される。ロボットの機械可動部と制御部を分離することにより、ロボットの制御方式とハードウェアを独立に開発することが容易になる。われわれはCMARSを開発する際に、柔軟性と作りやすさの点からこのような

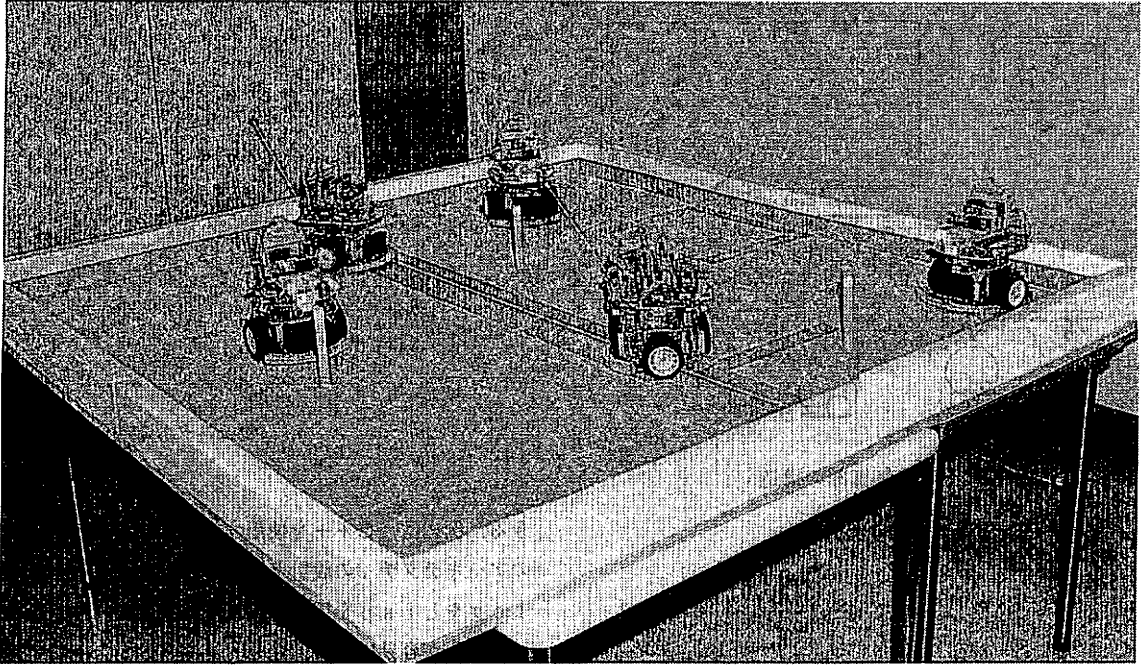


図 2: 移動ロボット群と実験環境

構成を選択した。研究の最終目標はCMARSによる協調行動の研究、つまり、協調機構の解明や協調アーキテクチャの開発と評価に重点を置いているので、当面の間、ロボットシステムの機械的な部分ではできるかぎり最小限度に止めておきたいからである。CMARSに要求される最も重要な部分は、ロボット間のコミュニケーションである。この機能は、バックエンドコンピュータとして利用するワークステーション同士がイーサネットを介してメッセージをやりとりすることによって実現する。

上述したさまざまな要求を達成するために、システムの制御部として機能するバックエンドコンピュータは、UNIXワークステーション群を用いる。UNIX環境では、並行プロセス機能、強力な通信機能などのさまざまなソフトウェア開発技術が利用でき、これらはCMARSの開発に非常に適している。

3.1 機械可動部

各ロボットの機械可動部は、直径が20cm、高さが25cm程の大きさである(図2)。これらのロボット群は、構造が定型化されたオフィス空間の中を

動き回るように設計されている。機械可動部には、オンボードプロセッサ(現在はMC6809)が搭載されており、2台のステッピングモータで2つの車輪を駆動する。センサ系は、2つの触覚センサと4方向超音波センサから構成される。

超音波センサは、物体とロボット間の距離を前後左右の4方向について計測することができる。オンボードプロセッサはモータやセンサなどロボットの入出力資源を操作・管理する。プロセッサは後述する制御部からのコマンドを実行し、センサ情報を制御部に戻す。

コマンドとセンサ情報は、ロボットの機械可動部と制御部の間で無線通信を介して交換される。プロセッサは制御部が存在するバックエンドコンピュータからコマンドを受け取ると、そのコマンドに基づいてモータの制御、ロボットと物体間の距離の計測などを行う。センサ情報は、制御部がその情報を要求したときに転送される。

Gung-Hoシステムの機械可動部は現在できるかぎり最小化を図っているが、最終的にはいくつかのマニピュレータも付加したいと考えている。

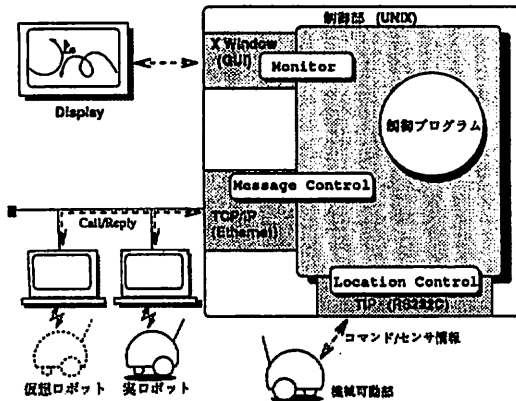


図 3: 制御部の構造

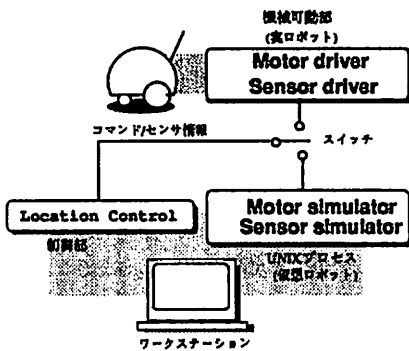


図 4: 実ロボットと仮想ロボットの切替え

3.2 制御部

各ロボットを実質的に制御するプログラムは、対応するUNIXワークステーションにより実行される。機械可動部へのコマンドおよび機械可動部からのセンサ情報は、図3に示される位置制御モジュール (Location Control) により処理される。ロボットの機械可動部を制御する各ワークステーション間の通信はTCP/IPプロトコルを用いて行われる。図3に示されるメッセージ制御モジュール (Message Control) は、メッセージを当該ワークステーションにイーサネットを介して送信することにより通信する。

分散協調方式を含むいろいろな制御方式は、UNIX上のプログラミング言語を用いて実現することがで

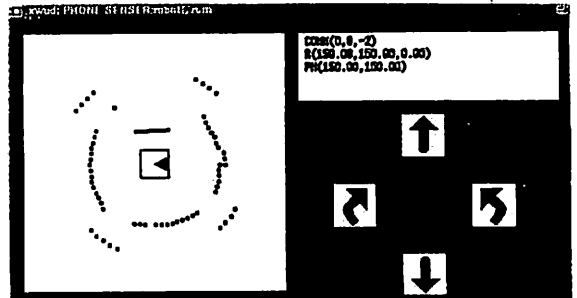


図 5: センサ情報のモニタリング

きる。例えば、C++を使用すればオブジェクト指向のアーキテクチャの設計と開発が容易になり、PerlやLispを用いればAI指向のアーキテクチャの導入が容易になる。Gung-Hoシステムにおいては一連の機能はプログラミング言語とUNIXの間のインターフェースをとるために用意される。UNIX環境においては、プログラミングツールばかりでなく、CMARSの開発に要求されるさまざまな機能が、部品としてあらかじめ装備されている。したがって、ロボット間通信のようなハードウェア指向の機能もUNIXソフトウェアを用いて容易に実現することができる。

制御部においては、多重プロセス制御、プロセス間通信および並行処理などの有用な機能が、UNIXワークステーション上で提供される。制御方式を構成する各モジュールはこれらの機能を利用して、ロボットのさまざまな制御方式を開発したり、試行したりすることができる。

3.3 シミュレータ/モニタ

試行環境において、シミュレータは仮想ロボットを生成する。仮想ロボットは実際はしないが、実ロボットと同様に同一のコマンドを受け取ることにより、実ロボットの機能を代替するものである(図4)。例えば、ロボットを取り巻く環境の地図を用いて、模擬のセンサ情報を返す(図5)。このシミュレータを利用すれば、開発中のアルゴリズムの改善やデバッグを実現

境でテストする前に即座に実施できる。Gung-Hoシステムでは、これらの仮想ロボットと実ロボットが共存し、相互作用をすることができる。この機能はまた、実ロボットの数が十分でない状況でも、多数のロボットからなるCMARSを容易にテストすることができる。

上述した制御用のプロセス群とモニタプロセスは同時に動作する。モニタは、位置制御モジュールやメッセージ制御モジュールから、各ロボットに関するすべての情報を獲得する。モニタプロセスは、Xウィンドウシステムを用いてワークステーションの画面上に、ロボットの位置、センサ情報、制御部の状態などロボットのさまざまな情報を表示する。モニタ自身は、仮想ロボットが実ロボットと同じ情報を生成するので、両者の区別をすることはできない。それゆえ、両方のロボットの状況を監視することができる。このツールは、ロボットの振舞い、例えば、実現している制御方式の効率などの記録や評価をするのに大変強力である。シミュレータやモニタの機能は、新しい制御方式やアルゴリズムを開発する際に大きな助けになる [2]。

4 行動原理に基づくマルチロボット制御方式の試作と検討

ロボットの開発環境としてGung-Hoシステムの有効性を確かめるために、簡単なロボット制御アルゴリズムのプログラムを書いてみた。行動原理に基づくマルチロボット (Behavior-based Multi-Robots) の例として、互いに交信しながら、目的地に行くように制御される移動ロボット群を考える。目的地はあらかじめ制御プログラムによりロボットに与えられているものとする。包括アーキテクチャに対しては、行動を起こしている各エージェントは、UNIXのプロセスとして実現され、並列に実行される。プロセス間通信は、UNIXの機能である共有メモリ機構により処理される。

例として、2種類のロボットを使用する。その1つは実ロボットで、他の1つはシミュレータによって生成された仮想ロボットである。ロボット間の位置データがやりとりされる。そして、仮想ロボットは実ロボットの障害物となる。実ロボットは、仮想ロボットとの

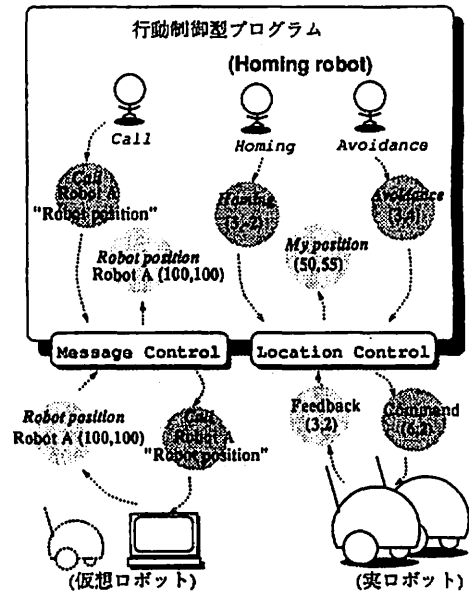


図 6: エージェントモデルによる制御方式の実現

衝突を回避しながら目的地に向かって動こうとする。実世界では、仮想ロボットは実際には存在しないので、実ロボットが活性化するのみである。

4.1 エージェント

上述したタスクを遂行するために、エージェントモデルを導入し、下記のエージェント群を用意した [7],[8]。

- HOMINGエージェントは、所定の位置へロボットを向かわせるコマンドを生成する。
- OBJECT-AVOIDANCEエージェントは、センサ情報と他のロボット位置情報を調べる。障害物が検出されると、衝突を回避するために方向を変えるコマンドを生成する。
- INTERRUPTエージェントは、目的地に行こうとする他のロボットの進路を邪魔して目的地に到達しないようにする。
- CALLエージェントは他のロボットの情報を更新する。

図 6 に示されるように、エージェントは制御部の中でさまざまな情報を交換している。エージェントに

よって生成されるコマンド群は、1つのコマンドにまとめられてロボットに供給される。センサ情報は、センサ情報要求コマンドを各ロボットに定期的に送信することによって収集される。

4.2 予備実験

まず、前節で述べたエージェント群を開発し、シミュレータやモニタを用いてテストする。次に、これらのエージェントを実ロボットで実現し、システムを実環境で実行する。この予備実験は、これまでに述べてきたすべての機能を実演するために、2台の実ロボットと1台の仮想ロボットにより遂行する。実ロボットは目的地へ進む(Homing)ように動き回り、仮想ロボットはそれを邪魔する(Interrupt)ように動く。

HOMING, CALL, OBJECT-AVOIDANCE エージェントによって制御される実ロボットは、INTERRUPT, CALL, OBJECT-AVOIDANCE エージェントによって制御される仮想ロボットとの衝突を回避するために、方向を変えながら与えられた目的地に向かって進んでいく。図7はこれらのロボットの軌跡を描いたモニタ画面を例示している。図に示されるように、エージェントは正しく機能している。実ロボットは仮想ロボットとの衝突を回避しながら、画面の3つの目的地の1つに到達する様子が示されている。

この実験はエージェントモデルと包括アーキテクチャの原理と特長をうまく表し、この方法の有用性を実証するものとなっている[9]。仮想ロボットを用いることにより、ロボット数が実在する実ロボットの数に制約を受けずに、マルチロボットシステムのための試行環境を提供することができる。もちろん実ロボットも使用するので、設計者はロボットと実世界の相互作用を容易に試行することができる。

4.3 検討

上記の例のような制御方式の試作を通じて、Gung-Hoシステムが分散ロボットアプリケーションを開発するのに十分対応できることがわかった。このシステムを利用することにより、上述した例プログラムの開発は1人でわずか半月で済んだ。この例が示すように、われわれのシステムは効率的なロボットプログ

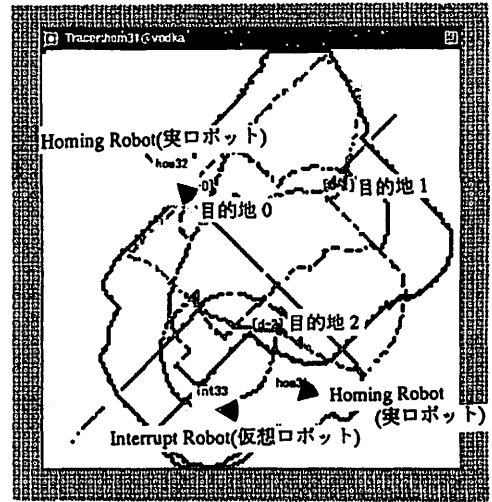


図7: 移動ロボット群の軌跡

ラミング環境とそのテストベッドを提供することができる。

しかしながらGung-Hoシステムは、ロボット制御アルゴリズムを前もってテストするための簡単な実験環境を提供するために設計されたので、リアルタイム操作機能が現時点では十分ではない。現時点では、ロボットの機械可動部が単純で動作が遅いので、リアルタイム操作の欠如が問題を起こすという状況にはまだなっていない。今後は、リアルタイム操作を取り扱うアルゴリズムの開発が、実世界のアプリケーションには不可欠になるものと思われる。

リアルタイム操作機能と同様、さらに複雑な状況で実施する実験に対応するためには、ロボットの機械可動部のハードウェアの改良が必要である。機械可動部をより高速にするために、オンボードプロセッサの強化を考えている。さらに、センサの種類と数を増やすことも計画している。現在の版は非常に単純なハードウェアが使用されているにもかかわらず、これまでに述べた予備実験で、CMARSの実験支援システムとして有用であることを示すことができた。

5 協調行動の考察とその試行

本来のCMARSを開発するためには、まず協調行動を考察し、次に適切なモデル問題を選択し、その問題をGung-Hoシステムで試行することにより、協調機構を解明することが必要である。

5.1 協調行動の分類

協調という用語は「協同調和」の略であり、いろいろな定義が考えられるが[10],[11], 広辞苑(第四版)にはその意味が次のように説明されている。

- (1) 利害の対立した双方がおだやかに相互間の問題を解決しようとする事。例) 労使協調。
- (2) 性格や意見の異なった者同士が互いに譲り合って調和をはかる事。例) 協調性。
- (3) 生体を構成する諸部分が相互に調整を保った活動をすること。

これらの意味を整理すると、協調行動は、複数の主体が共通の目標を達成するために、次の3つの場合に必要となるものと考えられる。

- (a) 競合(利害の衝突)の解決 妥協や遠慮によって解決する。例) 遠慮ロボット [12]。
- (b) 個性の補い合い 我を殺し、双方の長所だけを出し合う。
- (c) 一致協力 同期や調整を保って活動する。例) 協調融資、御輿かつぎなどの動作。

協調機構を解明するためには、上記の3つの分類を含む協調行動の試行ができる問題を選択し、考察する必要がある。

5.2 協調行動の試行

現段階のGung-Hoシステムの性能や能力では、実行できるタスクは限定される。したがってここでは、協調行動を試行するモデル問題として自律移動ロボット群による「棒倒し問題」を選んだ。これは、前節の

- (a) に対しては、どの棒をどのロボットが倒すかということで競合が起こる事象をどのように解決すればよいかを考える。

(b) に対しては、あるロボットが事故を起こして動けなくなった場合に、他のロボットが自らのタスクをどの時点で中断して相手を助ければよいかを考える。

(c) に対しては、大きな棒を複数のロボットで倒す際の一致協力の仕方について考察する。

これらの協調動作を、棒を倒すという大域的なタスクを与え、協調を促進あるいは後退させるような下記のいくつかの要素を変化させ、大域的タスクの終了時間、各ロボットが行うタスク数の偏り、タスク終了時間、走行距離などを通じて解析・評価するものである。

試行方法は、図2で示すような実験環境において、3台のロボットで15本すべての棒を倒すという大域的タスクを与える。処理は、実験の最中に不測の事故や任意の競合が起こることも想定している。実験は各ロボットとタスクについて、以下の要素を組み合わせて行う。

- 知識 ロボットはすべての棒の位置情報をもつ/部分的な棒の位置情報をもつ。
- 場 意図しなくても他ロボットの位置情報が入手できる/意図しないと他ロボットの位置情報が入手できない。
- 推論 他ロボットの棒の選択が予測できる/予測できない。
- 群構造 他のロボットの棒の選択に干渉できる/できない。
- タスクの性質 棒はロボット1台で倒れる/2台でしか倒れない。

協調問題については、タスクの分割をどのようにするかという点に関してすでにいくつかの研究がなされている [13],[14],[15]。ここでは、協調機構の解明を目指して、上述した要素の組み合わせにより、協調行動にどのような影響が及ぼされるのかについて、今後本格的な実験により調査していく。

6 おわりに

われわれは、CMARSの開発と研究のための支援システムであるGung-Hoシステムについて述べてきた。これはUNIXワークステーションを基盤に

しているので、UNIXが提供するさまざまな機能や開発ツールを利用することができる。したがって、エージェントモデルや包括アーキテクチャなど新しい方式のマルチロボットシステムの実現に非常に適している。また、ワークステーションがロボットの制御部として働き、機械可動部である移動ロボットを遠隔操作するという方法は、最終的なロボットの姿や仕様を開発を進めながら明確にしていく場合には、特に有効な手法である。

これまでは、分散協調ロボット群のための効率的なテストベッドを構成する基本要素を開発することに焦点を当ててきた。そのため、その効果を実際に示すために、単純なマルチロボットシステムを実現し、その有効性を確かめた。今後は、分散協調機構を解明するために、協調の概念の整理を行い、その試行ができる棒倒し問題を選択し、Gung-Hoシステム上でさまざまな実験をする計画である。

参考文献

- [1] 浅間一：マルチエージェントロボットシステム研究の動向と展望，日本ロボット学会誌，Vol.10，No.4，pp.428-432，1992.
- [2] I.A.Ferguson: Touring Machines: Autonomous Agents with Attitudes, IEEE Computer, Vol.25, No.5, pp.51-55, 1992.
- [3] M.Soejima, K.Utsumiya, H.Hikawa, T.Kodama and K.Yoshida: Gung-Ho: A Testbed for Prototyping Distributed Cooperative Intelligent Robots, Proc. of 1992 Int. Computer Symposium, pp.784-789, 1992.
- [4] R.A.Brooks: Intelligence without Representation, MIT Tech. Report, 1988 (邦訳, 柴田正良: 表象なしの知識, 現代思想, 18巻, 3号, pp.85-105, 1990.
- [5] R.A.Brooks: A Robust Layered Control System for a Mobile Robot, IEEE Journal of Robotics and Automation, Vol.RA-2, No.1, pp.14-23, 1986.
- [6] J.H.Connell: Minimalist Mobile Robotics, Academic Press, 1990.
- [7] A.Minsky: The Society of Mind, Simon & Shuster Inc., 1986 (邦訳, 安西祐一郎: 心の社会, 産業図書, 1990).
- [8] 中島秀之: エージェントモデル, コンピュータソフトウェア, Vol.9, No.5, pp.3-11, 1992.
- [9] 大里延康: マルチエージェントによるロボット制御, 電子情報通信学会論文誌, Vol.J75-D-1, No.8, pp.714-722, 1992.
- [10] T.W.Malone and K.Crowston: What is Coordination Theory and How Can it Help Design Cooperative Work Systems, CSCW'90 Proc., pp.357-370, 1990.
- [11] 石田亨: 協調とコミュニケーション, 信学技報 (人工知能と知識処理), Vol.92, No.84, p.35, 1992.
- [12] S.Premvuti and S.Yuta: Consideration on the Cooperation of Multiple Autonomous Mobile Robots, Proc. of IEEE Int. Workshop on Intelligent Robots and Systems, pp.59-63, 1990.
- [13] K.Ozaki, H.Asama, Y.Ishida, A.Matsumoto, H.Kaetsu and I.Endo: The Cooperation among Multiple Mobile Robots Using Communication System, Proc. of Int. Symposium on Distributed Autonomous Robotic Systems, pp.199-208, 1992.
- [14] 伊藤, 安西: 自律移動ロボットのための協調タスクプランニング, 電子情報通信学会論文誌, Vol.J75-D-II, No.12, pp.2038-2048, 1993.
- [15] 浅間, 尾崎, 松元, 石田, 速藤: 通信を用いた分散的管理に基づく複数の自律型ロボットの協調作業分担決定手法, 日本ロボット学会誌, Vol.10, No.7, pp.955-963, 1992.