

教育用ロボット制御システムの開発

兵頭 和人¹・山本 圭治郎¹・鈴木 良蔵²

- 1 機械工学科
2 大学院工学研究科機械工学専攻

Development of Educational System for Robot Control

Kazuhito HYODO , Keijiro YAMAMOTO , Ryozo SUZUKI

Abstract

This paper describes experimental and practical systems for education on robotics and mechatronics. Multi-agent architecture was employed, so that user can easily construct control systems. Mobile agents are convenient programs that can be dispatched from one computer and transported to a remote computer for execution. When the agent moves, it brings along its program code as well as its state(data). The agents can also communicate and interact with the others.

In this system, control system is constructed by combination of a set of agents. Mobile agent can be allowed to change the control system of the server without to program on the server machine. This flexibility can be extremely useful in designing the control system.

Key Words : Multi agent system, Education on robotics, Java language, Network

1. はじめに

ロボット工学及びメカトロニクスは、様々な領域の知識が素養として要求される分野である。そのため複雑なレベルでの教育を効果的に行う決定的な方法が無く、現行では座学もしくはシミュレーションによる模擬実験にとどまっている。しかし、実際の装置による体験を通じての学習は非常に効果がある。

「あまり細かい物を見せられると、物事の認識が困難になる」¹⁾。すなわち、授業のある段階で着目しているレベルにのみ学生が注力できる様な柔軟なシステムが必要である。本論文で提案するシステムは、自律した機能モジュール（エージェント）の結合によって制御システムの構成を行う、マルチ・エージェント型教育システムである。各モジュールが自律したモジュール（単独でも実行可能）であるので、その組み合わせ方によって、様々なレベルでのシステムの抽象化が可能であり、授業で着目しているレベルでの実習を容易に実現できる。すなわち、ロボットを完成されたブラックボックスとして見るだけでなく、必要に応じてそれを開くことができる知識と技術を身につけることを可能とする。

逆に、本システムを併用することにより授業の進め方も多様にでき、従来からのボトムアップ的な授業のみならず、トップダウン的な授業の形態が可能となる。学部の低学年においては、導入教育に重点を置き、システムの細部はブラックボックス化しておき、学年が上がるにつれて、ブラックボックスを開きながらシステム内部の詳細な授業を行い、再度、要素技術からシステムを構成して行く事により、システム全体を解析する能力を身に

つけるといった、効果的なカリキュラムを支援することが可能となる。

2. ロボット制御の階層

ロボット制御において必要となる機能は図1に示すような、3層構造に分類できる²⁾。

- ・組織化階層：行動決定、学習等の知的処理及びユーザインターフェースを提供する。
- ・調整階層：組織階層と実行階層の間のインターフェース。軌道生成、動力学計算等を行う。
- ・実行階層：位置・速度や力・トルク等の物理量を扱う。制御動作を行う。

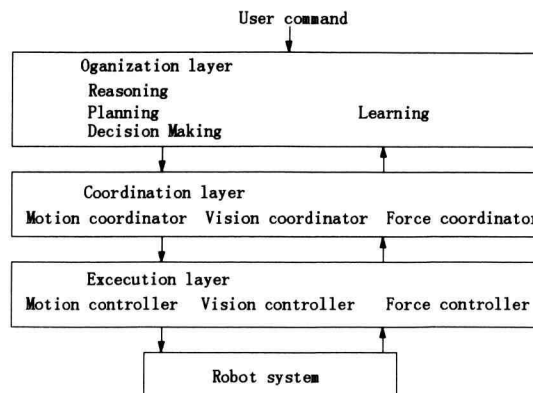


図1 ロボット制御の階層

3. クライアント/サーバ型システム

授業の段階にも依るが、ロボットまたはメカトロニクス実習装置を動作させるためには、何らかのプログラミングをする必要がある。その際、実際に実験または実習を行っている時間よりも、プログラムを作成している時間の方が遥かに長い。従って、設備等の制約のため、ユーザ数人で一つの実験装置(実験用ホスト)を共用することが必要となる、この方式で実習を円滑に進めるためにはネットワーク上にシステムを構築し、数台の端末ホストから実験用ホストにアクセスするシステムとし、マルチユーザ化することが必要である。

教育用システム開発の第1段階として、クライアント/サーバ型の制御システム(図2)を構築した。

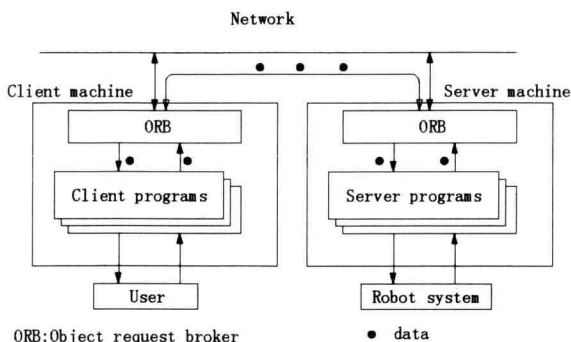


図2 クライアント/サーバ型制御システム

クライアント/サーバ型システムにおいても、システムの細部をユーザ(学生)から隠蔽でき、システム要素に対する知識を前提としない、実習・実験システムを構築することは可能であったが、以下のような問題点があった。

- a) サーバ側のカスタマイズが容易でない。
- b) ネットワークのトラフィックの問題。
- c) システムの拡張性が低い。

サーバ側が供給するコンポーネントの役割が開発時に決まってしまうので、ブラックボックス化のレベル変更が容易でない。実行階層のモジュールのように即応性が求められる部分に多様性を持たせ様々な実験実習を可能にするためには、サーバ側のシステムをユーザがカスタマイズする必要が生ずる。この作業中は他の実習者は長時間実験用ホストが空くのを待たされることになり、ネットワークを使用している効果が失われてしまう。

クライアント/サーバ型システムでは、その構造上、サーバ側からデータを受け取って見なければサーバ側で何が起きているのか知り得ない。そのため、その都度データの要求を実験用ホストに送り、実験データを受け取るという動作を繰り返すのでトラフィック作り出してしまふ。このネットワークのトラフィックを解消するには、必要なデータ/メッセージのみがネットワーク上で伝達されるシステム構造が必要である。

さらに、よりよい教育システムを構築して行く上では、実際にシステムを使用した学生からのフィードバックが大変貴重な情報である。このような情報に素早く対応

するには、システムの拡張・改造が短時間に行えるシステムである必要がある。それには、サービスを提供する側、受ける側と明確な性格付けがなされていてシステム構造の柔軟性に乏しいクライアント/サーバシステムでは対応しにくい。

それらの問題を解決するためには、端末用ホストで作成したモジュールを実験用ホストは移動させ、そこで実行させるモバイル・コードの枠組みの導入、各モジュールへの自律性の付加、各モジュールが情報の送信者でも受信者でもあり得るような枠組みの導入が必要である。

4. マルチ・エージェント型システム

前述の問題点を解決するため、本論文で提案する様な、自律したコンポーネント(エージェント)の集合としてシステムを定義する、エージェント志向型のシステムとして構築した(図3)。エージェント志向型システムとすることにより、ブラックボックス化のレベル設定が容易であり、授業で着目しているレベルのみをユーザに開示し、それ以外のエージェントの集合はブラックボックスとして取り扱っても効果的な実験・実習が可能であり、様々なレベルでの実験・実習に対して同一のシステムで対応でき、一貫性を持った実験・実習を行うことができる。マルチ・エージェントシステムの特徴としては以下の物が挙げられる。

- a) モジュールの自律性
- b) モジュールの移動性
- c) システムの柔軟性
- d) 負荷分散能力、並列処理能力

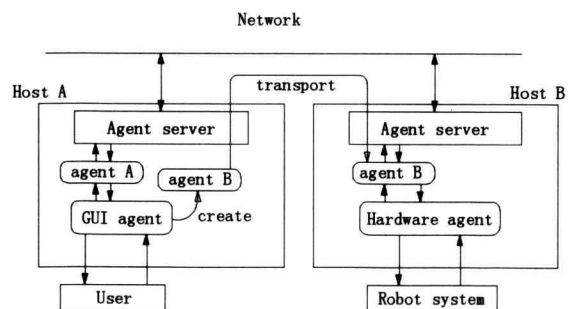


図3 マルチ・エージェント型システム

本システムでは、ユーザによるシステムのカスタマイズは、ブラックボックスとして入/出力に対する情報が与えられたエージェント群と独自に設計したエージェントを組み合わせることで実験環境の構成を達成できるので、多くの知識が前提として必要となることはない。

また、エージェントはネットワークを通じて実験装置が接続されたホストへ移動可能であるので、カスタマイズ作業はユーザの使用している端末ホスト内のみで行うことができる。つまり、即応性が要求されるエージェントは移動した後に実行を始めるようにしておくことで、実験装置が接続されたホストには一切手を加えることなく、ユーザの実験環境を構築できる。そのため、実際に

他のユーザが待たされる時間は、あるユーザが実験を行っている間だけとなる。

ネットワークのトラフィックの問題に関しては、一旦エージェントを送り込んでしまえば、端末用ホストと実験用ホストと間の接続は確保されている必要が無く、仮に逐次実験データを表示するなどシステムを構築した際も、本システムではデータにあらかじめ定めておいた変更があった時のみ、データを端末ホストへ転送するようなエージェントを実験用ホストに置いておくことにより、データ転送のためにネットワークを使用する頻度はクライアント/サーバシステムに比べて格段に低く抑えることができ、ネットワークのトラフィックの問題が解消される。

拡張性に関しても、追加の機能を実現するエージェントを作成することのみで対応できるので、ユーザからの要求に素早く答えることができる。

5. 本システムの記述言語

本マルチ・エージェント型教育システムでは、エージェントの記述言語としてJavaを採用した。

スクリプト型言語 (Basic, Lisp, Tcl/Tk 等) では、記述の簡便さ、ユーザとの会話性は優れているが、インタプリタによる実行時までそのプログラムコード内のエラーが見つからないため、安全なエージェント作成が困難であり、少なくとも移動を行うエージェントには適さない。

一方、バイナリ・オブジェクト型言語 (C++, C, Java 等) ではソースコードからバイナリ・オブジェクトへのコンパイル過程において文法上のエラーが検出されるので、スクリプト型言語に比べると、安全なエージェントが作成しやすい。しかし、C++及びCではポインタ (変数のアドレス参照) の明示的な使用を許可しているため、他のホストから移動してきたエージェントが安全に振る舞う保証が無い。これに対しJavaでは、明示的なポインタ使用を許可していないので、移動してきたエージェントの実行時の安全性を確保しやすい。

そのため、後述する一部のエージェントを除いてJavaで記述した。C言語によって実装された部分を持つエージェントはシステム管理者が責任を持って開発・運用する。また、エージェント開発環境としてIBMの東京研究所が開発中であるAglets WorkBench⁹⁾を使用している (現在はバージョンはアルファ版リリース5a)。

6. 開発システムの概要

開発システムは、図4に示すように端末用ホスト、実験用ホスト、2自由度ロボットアーム (2台)、4自由度アーム、6自由度アームから構成される。実験用ホストにはADコンバータ64ch、24軸分のモータ制御ボードが搭載されており実験用途に応じて、システムの構築が可能である。同時に24軸の制御を行っても、モータ制御のサンプリングタイムは0.4msである。4自由度アーム及び6軸アーム (三菱電機製) はそれぞれ専

用のコントローラを持ち実験用ホストによって制御されている。これらは、応用的課題を行う時に使用する。2自由度ロボットアームは、基礎実験用として当研究室で開発したアームであり、制御系の設計、軌道生成法等、基本的且つ理論的展開に重みを置いた課題を行うときに使用する。

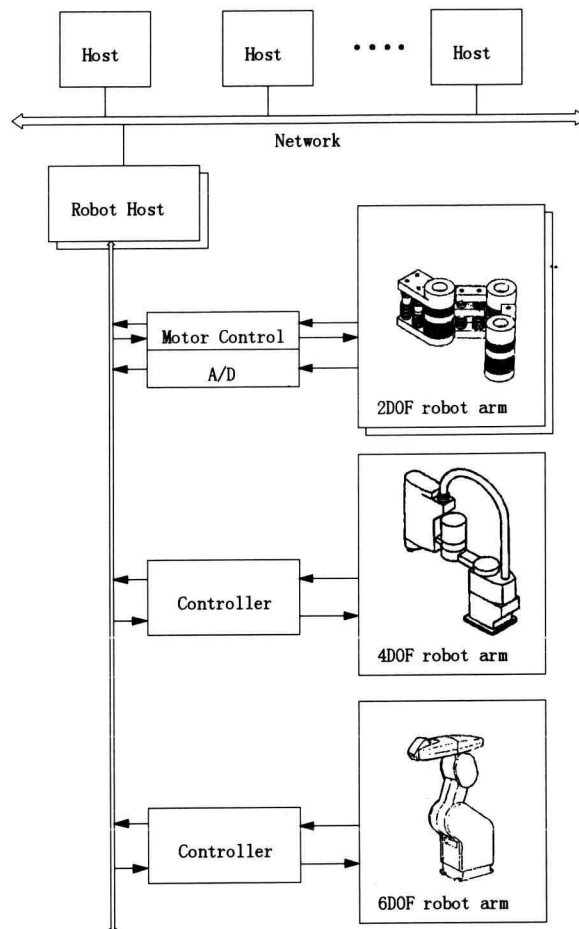


図4 開発システム

7. 本システムにおけるエージェント

7.1 移動しないエージェント

7.1.1 ハードウェア・エージェント

実行レベルのエージェントを構成するためには、ハードウェアへのアクセスが必要であり、Javaでは実現できない部分であるので、その一部をC言語で実装した。よって、各種ボード (モータ制御ボード等) をコントロールするエージェントは移動できないようにしてある。

7.1.2 ユーザインタフェース・エージェント

エージェント作成上の制約ではないが、ユーザとの対話を処理するためには、実験システムを起動しているユーザの端末ホスト上で動作する必要があるため、移動能力を持たない。

7.1.3 3次元表示エージェント

ユーザへロボットの状況を3次元グラフィックスで表示するエージェントも、実行時の速度の面から、その一部をC言語で実装した。よってこのエージェントも移動できない。

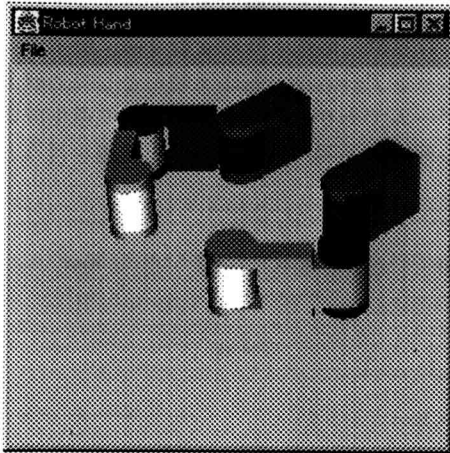


図5 3次元表示エージェントの例
(2自由度アーム2本を使用した状況の表示)

しかし、ハードウェア・エージェントは実験用ホストでのみ、後の2つのタイプのエージェントは実験を行っているユーザの端末ホスト上でのみ、存在意義がある物なので、本来移動能力を持っている必要のない物である。

7.2 移動可能なエージェント

7.2.1 組織化階層エージェント

ユーザが作成したアルゴリズムに基づき調整階層エージェント(群)に指示を出す。場合によっては、指示に対する結果を受け取り計画の修正を行ったり、学習の処理を行ったりする。必ずしも、作成されたホスト上で実行する必要はなく、ネットワーク上の空いている端末ホストに移動して実行することによって計算負荷の分散を図ることも可能である。

7.2.2 調整階層エージェント

組織化階層エージェント(群)から与えられる目標を達成するような軌道計画などを、ユーザが設計したアルゴリズムに基づいて行い、実行階層エージェント(群)へ各物理量の目標値を送る。これら目標値に対する達成度等のデータを上位のエージェント(群)へ送ることも行う。組織化レベルエージェントと同様に、負荷に応じて空いている端末ホスト上で実行する事も可能である。

7.2.3 実行階層エージェント(図6)

サーボ系を実現するためのエージェントはユーザの端末ホスト上で設計/製作され、実験用ホストへ移動し、ハードウェア・エージェントからロボットの各種データ

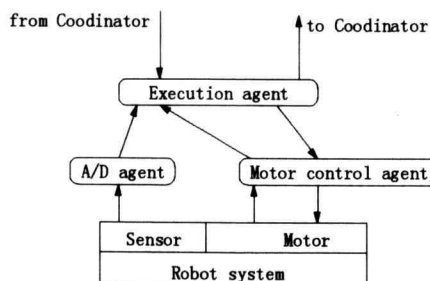


図6 実行階層エージェント

を、調整レベルのエージェントから目標値を取得し、何らかのサーボ演算を行い、ハードウェア・エージェントへ出力要求を行い、その担当動作を遂行する。このレベルのエージェントを作成する際、ユーザはハードウェア・エージェントのもつデータへのアクセス法(例えば第1モータの現在位置の取得等)、のみを知っていれば、サーボ系を自由に設計できる。

8. 本システムにおけるエージェントの作成

本システムにおける、エージェントのプログラミングは、比較的簡単に行うことができるが、各エージェントはJavaのスレッドとして実装してあるのでデッドロックの回避を考慮したプログラミングが必要であり、また各階層毎に、決まり切った定型処理が含まれているので、それらをデザインパターン⁴⁾として、それらの設計上の定石を再利用可能なクラスとして、抽象化することにより、より一層のプログラミングの簡略化を図る。

8.1 組織化階層エージェント

この階層のエージェントはユーザインターフェースエージェントによって生成され、調整階層エージェントとメッセージ交換を行いながら所定の機能を実現する。この階層のエージェントの動作を特徴付けるもの(つまりユーザが作成しなければならない部分)は、生成時に所定の調整階層エージェント(群)を生成すること、所定のアルゴリズムに基づいて調整階層への目標を計算すること、終了時の処理を実装すること、他のエージェントからのメッセージを処理することである。そこで、抽象クラスでは生成時にメソッドinit()を、実行開始後にメソッドexec()を、終了時にend()をコールするようにしてある(抽象クラスにおいては何もしない空のメソッド)。ユーザはそれらのメソッドをオーバーライドする事により、独自の初期化処理、実行部処理、終了時処理を実装することができる。メッセージ処理については、抽象クラスのメソッドhandle_Message()で"start", "stop"のメッセージに対し、それぞれ実行開始、動作停止を行うようにしてあるので、その他のメッセージに対する動作の規定の部分のみを実装し、handle_Message()をオーバーライドする。

従って、組織化階層のエージェントを作成する際には抽象クラスOrganizerを継承したクラスを定義し、上記4つのメソッドをオーバーライドすることのみで完成する。

8.2 調整階層エージェント

組織化階層エージェントと同様に、抽象クラスCoordinatorを継承し、init(), exec(), end(), handle_Message()をオーバーライドする事によって、所定の実行階層エージェント(群)の生成・実験用ホストへの配送、所定の軌道計算・動力学計算等の実行、終了時処理、独自のメッセージ処理などを実装した調整階層エージェントが完成する。

8.3 実行階層エージェント

実行階層エージェントを作成する際には、抽象クラス

Executor を継承し、メソッド `init()`, `exec()`, `end()`, `handle_Message()` をオーバーライドする事のみで、所定のロボットを制御するのに必要なハードウェアエージェントを実験用ホスト内で検索し、サーボ演算を行い、所定の終了処理、メッセージ処理を行うエージェントが作成できる。

8.4 3次元表示エージェント

Silicon Graphics 社が開発した OpenGL を Java のネイティブメソッドとして使用して3次元表示を行っており、通常手間のかかるレンダリングコンテキストの作成・設定などは抽象クラス `View3D` が行うので、ユーザーは初期化処理、描画処理、表示の射影変換設定用のメソッド `init()`, `draw()`, `prep()` のオーバーライド及び表示物体の形状データを用意することによって、独自の3次元表示を行うエージェントを作成できる。なお、物体の形状データファイルは物体をポリゴン表示するための、頂点データ、面の構成データの並びが書き込まれたファイルである。今後 VRML 形式のデータに対するパーサーを開発し実装する予定である。

8.5 ユーザインタフェース・エージェント

組織化階層エージェント及び3次元表示エージェントの生成/制御に関する部分、ボタンなどユーザインタフェース部品を簡便に配置するためのレイアウトマネージャ等を実装した抽象クラス `Monitor` を使用して、プログラミングを行う。今後、計測データのグラフ化等のデザインパターンを用意して行く予定である。

9. ブラックボックス化のレベル設定について

9.1 導入教育

導入教育時は、動機付けを行うことが目的であるので、システムのかんりの部分をブラックボックス化した状態で、実習・実験を行う。

調整階層以下のシステムは、あらかじめ用意しておいた物を使用して、ごく簡単な組織化階層のエージェントの作成を通じて、ロボット制御の全体像が掴めるような課題を行う。例えば、軌道生成、逆運動学、ハードウェアへのアクセス法などはブラックボックスとして行った、「積み木の積み替え動作を遂行する動作計画の作成」(平成9年度の機械工学セミナー I の山本研究室配属の学生が行った課題)では、効率的に作業を行うための動作計画を立案し、実際のロボットによって作業を行い、計画との差異等についての検討を行い、再度計画を見直すという一連の過程において、ロボット制御における問題点の認識ができた。

この実験において使用した軌道生成法は、与えられた作業空間における始点と終点のみ逆運動学によって関節空間へ変換し、その変換された関節座標ベクトルを用いて、関節空間における補間軌道を生成しているので、作業空間内では軌道は直線にならず、積み木との接触が生ずることもある。そのため学生は、中間点を指定したり、余裕を持った動作計画を立てたりと何らかの対策を立て

ながら実験を行っていた。また動作速度と振動等、ロボット工学の講義への導入としての教育効果は高いと思われる。このように、本システムを用いれば、単にシステム全体をブラックボックス化して、安直な実験を行うのではなく、その後の教育への導入効果が期待できるような部分的にブラックボックス化された教材の提供が可能である。

9.2 基礎段階

システムを細部から分析する過程においては、本システムを構成している様々なエージェントを機能ブロック毎に解析して行くことによって、ロボット制御の詳細を身につけて行くことができる。その際に、着目している機能ブロック以外のエージェント群はブラックボックス化しておくことにより、解析を行っている部位のみへ集中する事ができる。これらは主として、調整階層以下のエージェントの設計・製作を通して行われる。

現行のカリキュラムに適応させると、以下のような流れになる。全ての項目で必ずしも実験が必要となるわけではないが、数値によるシミュレーションだけでなく、ビジュアライズすることによって講義内容の理解が高められる。

9.2.1 順/逆運動学

同次変換行列による物体の移動・回転などの理論を3次元表示エージェントを使用して、仮想空間上にロボット及び作業環境を構築することにより、効果的に解説することができる。

9.2.2 軌道生成(調整階層)

前項で構築した部分をブラックボックスとして使用し、軌道の生成の仕方による、ロボットの動作の違いを解析する。

9.2.3 制御法(実行階層)

様々な制御系の設計を行い、制御系の構造/制御パラメータによる、軌道誤差/振動抑制/移動時間/外乱抑制等の特性への影響を解析する。

メカトロニクスの講義では、ハードウェア・エージェントの動作の仕方も学習する。

9.2.4 動力学(調整階層)

ロボットの動力学に基づくエージェントを作成し、実験を行うことによって、動力学の各要素の物理的意味の理解を深める。

本システムでは、エージェント群の組み合わせでシステムが記述できるので、各段階で設計・製作したエージェントは、そのまま他の段階でのブラックボックスを構成する一要素として使用したり、応用段階での教育に再利用することができる。

9.3 応用段階

作業計画を自動生成するアルゴリズム(導入教育時には、試行錯誤しながら人手で作成する)、知的制御(ニューロ、ファジィ、GA等)、より複雑な制御系の構築(H_∞制御、非線形制御)、複数のロボットによる協調動作など応用的な課題を、組織化階層エージェントの設計・製作を通じて行う。

所定の解析を行うのに適した調整階層以下のシステムは、基礎段階で使用したエージェントを組み合わせることにより構成する。これにより、複雑な課題に専念することができる。

応用段階では、ブラックボックスの中身に対する知識を元にシステムを学生が構築する点が、教材として与えられたものを使用する導入教育との大きな違いである。

この段階の課題では、実習者によって必要とする実験システムが多様になることが予想されるので、システム構成の変更に対する柔軟性の高い本システムを使用することによる教育効果が大きい。

10. おわりに

本論文では、マルチ・エージェント型教育用ロボット制御システムを開発した。マルチ・エージェント型システムとすることにより、様々な段階における教育を効果的に行うことを支援する、実験システムの実現が可能となった。現在、平成10年度からの機械及び電気工学実験II「サーボ機構」のためのシステムを構築中である。

今後は、エージェントによるシステムの構成を支援するためのグラフィカルな開発環境、自習時に使用するためのシミュレーション・エージェント、実験システム内を巡回しシステムの状態(CPU, ロボットの状況)を把握するための管理エージェントの開発を進める。

また、力覚、視覚センシング・システムを追加することにより、より多様な実験・実習が行えるように進展させる。

謝辞

本研究の一部は、日本私学振興財団「特色ある教育研究」(平成7年度)、「特色ある教育研究の推進」(平成8年度)の研究助成によるもので、ここに改めて謝意を表します。

尚、Javaはサンマイクロシステムズの登録商標、OpenGLはSilicon Graphics社の登録商標である。

参考文献

- 1) M. Minsky 著, 安西祐一郎訳:『心の社会』, 産業図書, 1990
- 2) Kimon P. Valavanis, George N. Saridis: "Intelligent Robotic Systems", Kluwer Academic Publishers, 1992
- 3) Danny B. Lange, Mitsuru Oshima: "Programming Mobile Agent in Java", <http://www.trlibm.co.jp/aglets>
- 4) Erich Gamm 他著, 本位田真一, 吉田和樹監訳: 『デザインパターン』, ソフトバンク, 1995