

複数の機能モジュールを組合せる ロボットの構築手法

Robot Construction by Combination of Several Functional Modules

足立匠・アダチ タクミ・Adachi Takumi／長坂保典・ナガサカ ヤスノリ・Nagasaka Yasunori

SUBJECT

ロボットの新しい構築手法

ABSTRACT

本論文では、分散型制御を用いたロボットの一種である、モジュール型ロボットを基礎とし、モジュールの不均一化によるロボットの拡張性の向上を提案した。コアモードモジュールとエンドモードモジュールを組み合わせることでロボットを構成し、異なる構成でも共通の命令で動作することを確認した。また、意図的にモジュールにエラーを発生させ、モジュールが使用不可能な状況に陥った場合でも、動作可能なモジュールを判断しエラーに対応できることを確認した。

複数の機能モジュールを組合せるロボットの構築手法

足立 匠^{*} , 長坂 保典^{**}

Robot Construction by Combination of Several Functional Modules

Takumi ADACHI^{*} and Yasunori NAGASAKA^{**}

(Received October 3, 2012)

We present a robot construction concept by introducing a combination of non-uniform functional robot modules. It has two merits: higher availability and reliability coming from a modular system, and higher adaptability and wider range of available work coming from problem-specific module design. It also realizes distributed control. We have made several types of modules. A locomotion module gives a locomotive function to the robot. An arm module works like arms or legs of animals. It can mount other modules on the top of itself. An "other" module includes any other functional modules, like sensors, actuators, and special devices. There are two operation modes of modules, one is "core mode" and the other is "end mode". Only one of the robot's modules operates in core mode. It sends commands to other modules and controls other modules like an animal brain. Other end mode modules are controlled by the core mode module. We built two different prototype robots as combination of modules for experiments. In experiments we investigated whether those robots can reconfigure their functions dynamically against troubles (sudden disappearance and addition of modules). As a result, we confirmed that each robot successfully reconfigured their functions and continued the given work.

Keywords : module robots, non-uniform modules, distributed control

1. はじめに

一般にロボットを設計する際、その制御方法により、集中型制御と分散型制御に分類される。集中型制御は一つの高速なプロセッサでロボット全体の制御をしており、その特徴は各制御対象を直接制御するため、複雑な動作や特殊な機材を扱えることにある。しかし、全体を制御しているメインシステムに障害が発生した場合、全

機能が停止する可能性がある。一方、分散型制御方法は、ロボットの制御を複数の構成要素のプロセッサに分散して行う方式であり、複数のシステムが分かれているため、一つのシステムが機能停止した場合でも全体に及ぼす影響は小さい。しかし、ハードウェア・ソフトウェア共に接続部の規格や通信方式など、全てのシステムがその規則に準じていなければならないという欠点を持つ。また、複数のシステムを管理するためのシステムが必要となる。

^{*}電気電子工学専攻(Electrical and Electronic Engineering)

^{**}電子情報工学科(Department of Electronics and Information Engineering)

分散型制御を利用したロボットは多く存在する。一般に、モジュール型ロボットと呼ばれるロボットも分散型制御を採用したロボットであり、代表的なロボットとして産業技術総合研究所のM-TRAN[1, 2]やLund氏が開発したATRON[3]が挙げられる。これらのロボットは均一なモジュールで構成されており、個々のモジュールにアクチュエータが搭載されている。そして、このモジュールを共通化された接続部で連結し様々な形状・動作を可能にしている。また、制御部も個々のモジュールに搭載されているため、一部のモジュールにトラブルが発生しても、システム全体への影響は少ない。その場合、正常なモジュールと交換することにより修復することが可能である。しかし、均一なモジュールだけで構成されたモジュール型ロボットは、モジュールの機能が限定されるため、可能な作業は集中型制御と比較すると少なく、動作のパターンも限られてしまう。それを補うように、モジュールに様々な機能を搭載した場合、モジュール自身の巨大化やシステムが複雑化してしまう。

本論文では分散型制御でありながら均一なモジュールではなく、不均一なモジュールで構成されたモジュールロボットシステムを提案する。均一なモジュールでは、先に述べた問題により、特殊な機能をモジュールに搭載することはできないが、不均一にすることによりそれぞれ独自の機能を搭載することが可能となる。機能ごとにモジュールを製作することにより、状況に応じて、必要なモジュールだけでロボットを構成することが可能となり、現場の環境や課題に対し、優れた適応が図れると考える。また、モジュール型を採用することにより、トラブル発生によるシステム全体への影響を最小限に抑えたと共に、独自の機能を搭載したモジュールの組み合わせによる現場や環境への高い適応度を実現することを目的とする。実証実験として、動作中におけるモジュールの追加および、モジュールの故障によるシステム停止に対する対応を検証する。

2. モジュール

分散型制御により、特定の機能を実現するため、マイコンやアクチュエータなどで構成される1つの回路や機器の集合をモジュールとする。このモジュールを組み合わせることによりロボットを構成する。

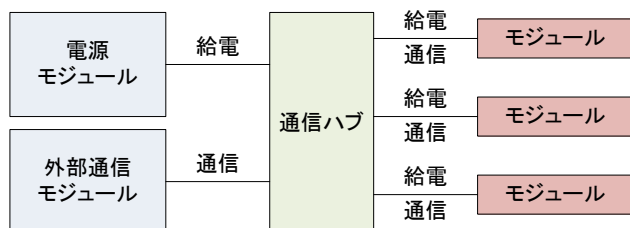


図1 モジュールの接続図

ロボット全体の構成を図1に示す。モジュールは全て通信ハブに接続する。

一つのモジュールの最大接続数は、給電される電源の容量とシステム全体を管理するプロセッサの処理能力により決定される。今回使用する電源は容量が小さく、システムを単純化するため、接続の上限を10モジュールまでとした。以下では様々なモジュールの例を示す。

2.1. 移動モジュール

移動モジュールは、ロボット全体を指定する方向へ移動させるモジュールである。製作した4輪型移動モジュールの外観を図2に示す。

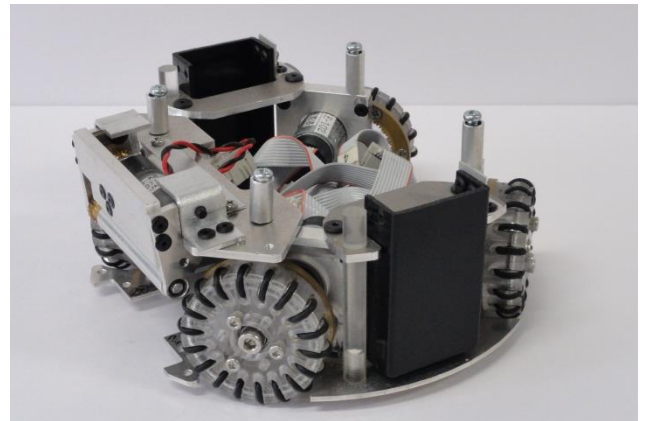


図2 4輪型移動モジュール

4輪型移動モジュールは、4個のオムニホイールを使用することにより全方向移動が可能なモジュールである。

2.2. アームモジュール

アームモジュールは、主にその先端に取り付けた別のモジュールを指定された目的地まで移動させるモジュールである。今回製作したアームモジュールを図3に示す。

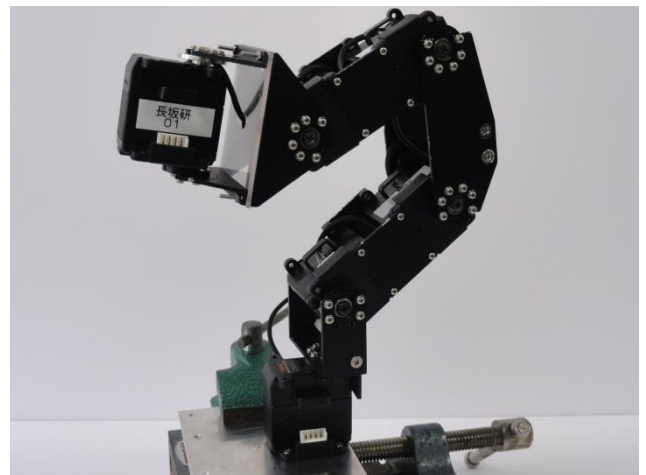


図3 アームモジュール

先端に他のデバイスを接続するためのコネクタが取り付けられており、通信ハブへ接続する事ができる。アームモジュールは図4に示すシリアルサーボモータを各関節に使用している。



図4 アームモジュールの関節に使用したシリアルサーボモータ

このモータは通信規格 RS485 でシリアル通信することが可能であり、モータの回転角度やパラメータの読み書きをする事が可能である。

2.3. その他モジュール

その他モジュールとは、移動モジュールやアームモジュールに属さない特殊な機能を持つモジュールを指す。今回製作したその他モジュールは、モノを掴むことを目的としたハンドモジュールである。図5にその外観を示す。



図5 その他モジュール（ロボットハンド）

このハンドモジュールは2つのシリアルサーボモータで構成されており、モータに取り付けられたツメを開閉させることによりモノを掴むことが可能である。

2.4. 通信モジュール

通信モジュールとは、ロボットが外部と通信をするためのモジュールである。図6にその外観を示す。

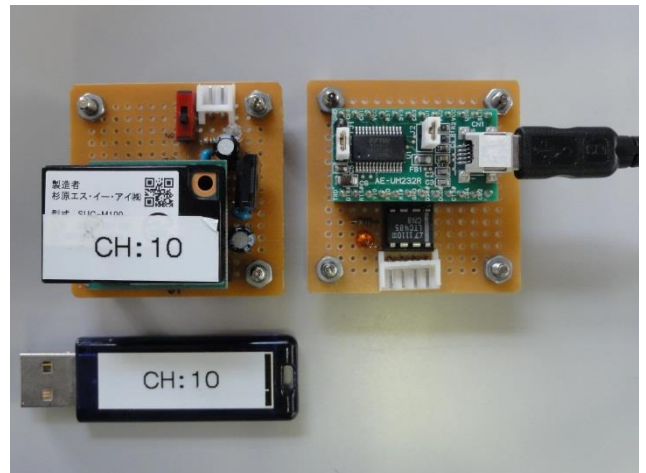


図6 通信モジュール(左:無線式, 右:有線式)

通信モジュールは図6の左側に示す無線式と、右側に示す有線式の2種類がある。この2つはどちらもパソコンとUSBで接続することが可能であり、直接通信ハブを流れるコマンドをモニタすることが可能である。

2.5. 電源モジュールと通信ハブ

電源モジュールは、ロボット全体にモジュールの動作電圧 12V を作り出し供給するモジュールであり、通信ハブはモジュール同士を電氣的に接続し、全モジュールへの給電や、モジュール間の通信の中継機能を持つ。図7にその外観を示す。

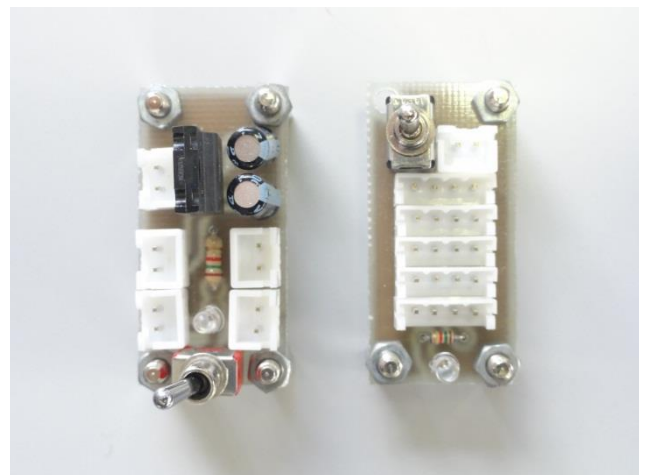


図7 左:電源モジュール, 右:通信ハブ

2.6. モジュールの制御基板

移動モジュール、アームモジュール、その他モジュールは図8に示す、ベースユニットと制御対象を制御する回路で構成される。

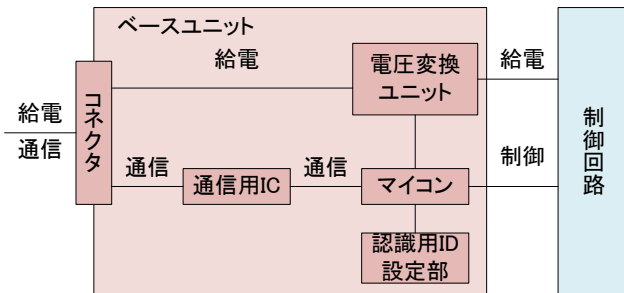


図8 制御基板の構成

ベースユニットは、他の基板と通信するための通信用インターフェース IC と演算部となるマイコン、個々のモジュールを識別するための認識用 ID 設定部と通信用 IC とマイコンの電源電圧を作り出す電圧変換ユニットで構成されている。制御回路は制御する対象ごとに異なるため、モジュールごとに設計されている。

3. モジュールの動作

モジュールにはコアモードとエンドモードの2つのモードが存在する。コアモードとはモジュール全体を管理するモードであり、ロボット全体のモジュールの情報が登録されているデータベースを持つ。このデータベースを元に与えられた命令を実行可能なモジュールへ転送する。また、エンドモードは管理を受けるモードであり、コアモードモジュールから転送されてくるコマンドのみ実行する。

3.1. コアモードとエンドモード

常に通信ハブを流れるコマンドを監視しており、コアモードのモジュールがネットワーク上に存在するか確認をしている。もし、コアモードモジュールからの信号が発信されない。コマンドを送信しても反応がない場合は、コアモードモジュールのメインシステムが停止したと判断し、正常に動作することのできるモジュールで通信し、新たにコアモードモジュールを選択する。

3.2. 移動モジュールの動作

移動モジュールは、図9に示す進行方向、進行速度、

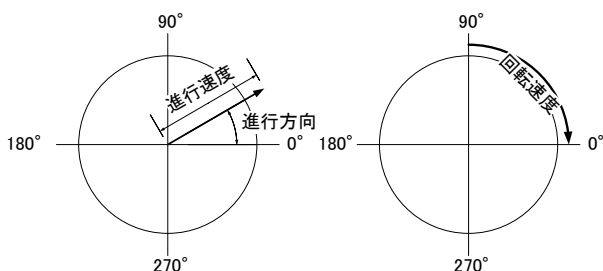


図9 移動モジュールのコマンドパラメータ

回転速度の3つのパラメータで動作を決定する。

移動方向はロボット正面を 90° としたときの進行角度を指定する。また、進行速度は移動モジュールの出力を[%]で表し入力値は-100~100である。回転速度も出力を[%]で表し、同様に入力値は-100~100である。

3.3. アームモジュールの動作

アームモジュールは、モジュールの取り付け位置を原点として図10に示す座標軸の設定で、アームの先端座標(X, Y, Z)、およびモータの回転速度を指定する。

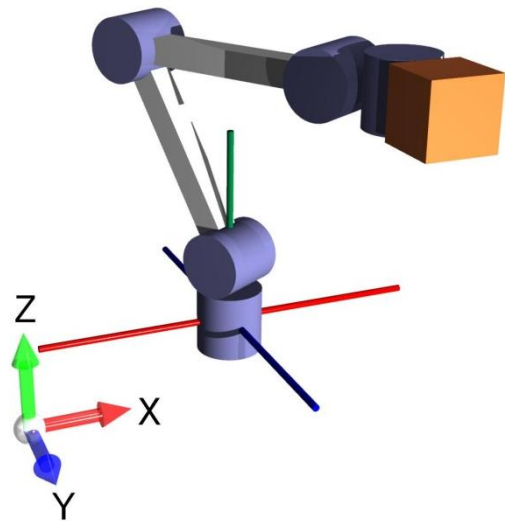


図10 アームの座標軸の設定

3.4. その他モジュールの動作

その他モジュールは、定められた書式やパラメータは存在せず、それぞれのモジュールごとに決められたコマンドが存在する。なお、今回製作したハンドモジュールはアームの回転角度、回転速度を指定する。

3.5. モジュールの故障と再接続

コアモードモジュールは、先に述べたデータベースを元に、登録されているモジュールへ定期的に返信要求コマンドを送信している。このコマンドに対し、エンドモードモジュールから返信がない場合は、データベースから該当するモジュールのデータを消去する。

また、データベースに登録されていないモジュールへコマンドを送信し、一定時間以内に返信がある場合は、新しいモジュールを発見したとみなし該当するモジュールの情報をデータベースへ登録する。

3.6. モジュール間の通信プロトコル

モジュール同士で情報のやり取りをする際、あらかじめ決められたコマンドを使用して通信を行う。コマンドには基本文法が存在し、先頭から

Source, Destination, Function, Parameter の順となっている。Source には送信者が、Destination には受信者が当てはまり、この部分には表 1 に示す記号を使用する。

表 1 記号一覧

記号	内容
A	全てのモジュールが対象
E	エンドモードで動作しているモジュールが対象
C	コアモードで動作しているモジュールが対象
H	外部の通信端末が対象
数字	同じ ID のモジュールが対象

Function の部分には、何をするのかを表 2 に従ってコマンドを指定する。

表 2 コマンド一覧

コマンド	内容
get	情報を取得する
set	情報を設定する
check	動作可能か確認をする
run	動作の実行を命令する

Parameter にはその都度必要なコマンドを指定する。以下に、簡単な例を示す。

H, C, set, parameter, ***

この場合、“外部の通信端末(H)からコアモードで動作しているモジュール(C)へ、パラメータ(parameter)を***に設定(set)する。”という内容となる。

4. 補助アプリケーション

補助アプリケーションは、ロボットの外部で動作するアプリケーションであり、図 11 に示すようなパーソナルコンピュータ上で動作するプログラムを今回製作した。このアプリケーションには、モジュール間を流れるコマンドを表示するコマンド表示部、ロボットの動作状況を表示するイベント表示部がある。この 2 つにより、ロボットの現在の状況をモニタすることが可能である。また、コアモードモジュールから命令要求された場合のみ、このアプリケーションからコマンドを送信することが可能である。よって、ロボットへ順次送信するコマンドの入力部である、送信コマンド入力欄がある。

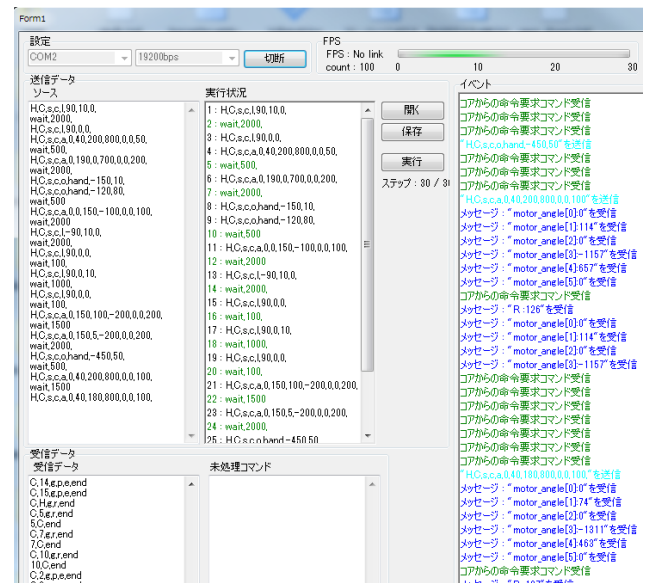


図 11 補助アプリケーションの画面表示

5. 動作実験

今回の実験はロボットに課題を与えるため、補助アプリケーションからコマンドを送信した。また、各モジュールの動作の監視も同アプリケーションで行った。以下に示す実験は、意図的にエラーを発生させ、その時のロボットの対応を検証すると同時に、異なるモジュールの構成下において動作するかどうかについても検証を行った。

5.1. コアモードモジュールの故障と対応

通常、必ず 1 つのコアモードモジュールが存在し、ロボット全体のモジュールを管理する。しかし、何らかの原因によりコアモードモジュールが故障し、通信不能になる可能性がある。その際、動作可能なモジュールの中から新たにコアモードモジュールを選び、再び動作する。この内容を検証するため以下の実験を行った。図 12 に機器の配置を示す。

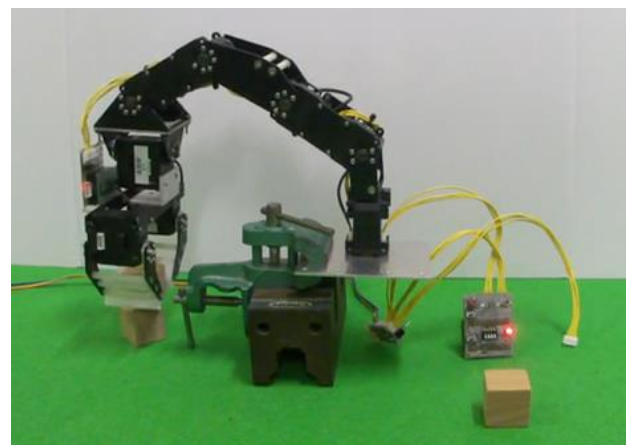


図 12 実験風景 1

使用したロボットの構成は、アームモジュールの先端にハンドモジュールが接続されており、コアモジュールとして移動モジュールの制御基板のみ、通信ハブに接続されている。また、このロボットには図 12 に示すように手前に 3 つの積み木があり、これを左に積み上げる課題を与えた。この条件で、ロボットの動作中にコアモードモジュールとして動作している移動モジュールを通信ハブから取り除くことにより故障した状況を作り出す。その後、コアモードモジュールを自動的に決定し、作業が再開できるか検証した。

・結果

はじめ、すべてのモジュールを接続した状態で一番左の積み木を運搬する。次に真ん中の積み木を取ろうとした瞬間にコアモジュールである移動モジュールの制御基板を通信ハブから取り除く。その後ロボット全体の動作が停止するがハンドモジュールがコアモードとして動作を開始し、作業を再開した。

5.2. 新しいモジュールの発見

コアモードモジュールは、与えられた命令を処理すると同時に、常に通信ハブに新しいモジュールが接続されていないか確認をしている。そこで、動作途中のロボットに新しくモジュールを接続し、自動的にコアモードモジュールが持つデータベースを更新し、与えられる命令を割り振ることができるのか検証した。図 13 に機器の配置を示す。

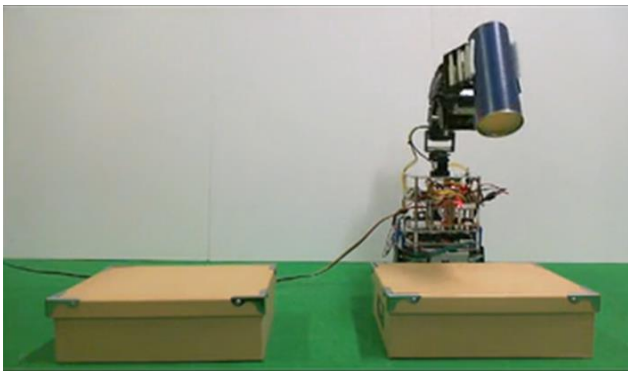


図 13 実験風景 2

今回のロボットの構成は移動モジュールの上にアームモジュールを搭載し、その先端にハンドモジュールを接続した。今回、アームモジュール、ハンドモジュールを動作途中で検出するため移動モジュールだけ電源が ON になっており、アームモジュール、ハンドモジュールは意図的に電源が OFF になっている。また、このロボットには箱の上に置かれた筒をもう一方の箱へ移動させる課題を与えた。

・結果

アームモジュール、ハンドモジュールの電源が OFF の

時はコアモードモジュールである移動モジュールへアームやハンドのコマンドを送信したが全てエラーコマンドが返信で帰ってきた。次に電源を ON にした瞬間アームモジュールとハンドモジュールの情報の取得を開始し、アームを使用して筒を移動させることができた。

6. 考察

今回、不均一なモジュールによるロボットの構築について、モジュール間で連携しての動作やトラブル発生時の対応などの検証を行なった。このシステムは、モジュールを組み合わせるだけで、ロボットを自由に構成することが可能であるため、非常に高い即応性と柔軟性を両立することができる長所がある。しかし、常時稼働し続けるようなロボットで構成の変更を必要としない場合は、一般的な集中制御のシステムと比較した場合に機構が複雑で部品数が多くなるため、メンテナンスが大変であるという短所がある。ロボットを使用する状況やロボットで解決する問題が、時間の経過に従って変化していく現場（災害現場など）で、ロボットの構成を必要に応じて変更する場合に本システムの長所が活かされると考える。

7. まとめ

本論文では、分散型制御を用いたロボットの一種である、モジュール型ロボットを基礎とし、モジュールの不均一化によるロボットの拡張性の向上を提案した。コアモードモジュールとエンドモードモジュールを組み合わせることでロボットを構成し、異なる構成でも共通の命令で動作することを確認した。また、意図的にモジュールにエラーを発生させ、モジュールが使用不可能な状況に陥った場合でも、動作可能なモジュールを判断しエラーに対応できることを確認した。

以上より、集中型制御の複雑な制御の可能性と分散型制御のシステムの安全性・保守性を兼ね合わせることが可能であることを確認できた。

このシステムは、広範な環境や課題が存在する災害現場など緊急時に即座に対応できることが期待できる。

今後の課題はロボットの操作を外部からだけではなく、モジュールに搭載されたメモリから読み出し動作可能とするなど自律行動について研究を進めていきたいと考える。また、外郭の環境の把握をするため、多くのロボットにはセンサーが取り付けられているが、今回のモジュールはその役割をするモジュールが存在しなかった。センサーモジュールは移動モジュールなど命令を動作として処理するのではなく、値を取得するモジュールとなるため、扱い方が異なる。この点についても今後考える必要がある。

謝 辞

本研究は、中部大学情報科学研究所および中部大学特別研究費(CP)の支援を得て行われた。ここに記して謝意を表す。

参 考 文 献

- 1) 神村、村田、吉田、黒河、富田、小鍛：自己組立可能なモジュール型ロボットシステムに関する研究，日本機械学会論文集(C編) Vol. 68 No. 667, pp. 886～892, 2002.
- 2) 黒河、吉田、神村、富田、村田、小鍛：変形し移動する自立モジュール型ロボット、日本ロボット学会誌 Vol. 21 No. 8, pp. 855～859, 2003.
- 3) M.W. Jorgensen, E.H. Ostergaard and H.H. Lund: "Modular ATRON: Modules for a self-reconfigurable robot," Proc. of the 2004 IEEE/RSJ International Conference on Intelligent
- 4) 川内、稲葉、福田：セル構造化ロボットシステムに関する研究(適応性、自己組織性、自己進化性のあるロボットを目指して)、日本ロボット学会誌 Vol. 12 No. 1, pp. 1116～1132, 1994
- 5) 渡部、王碩、石田、井上、河田、水野、木村：全方向移動型歩行訓練機の経路追従の制御、日本機械学会 Vol. 2008 No. 46, pp. 423～424, 2008
- 6) 出村 公成，簡単！実践！ロボットシミュレーション、森北出版株式会社