

機能分散制御アーキテクチャに基づいたロボットシステムの設計

Design of Robot System based on
Functionally Distributed Control Architecture

○ 平 哲也 (慶大) 山崎 信行 (慶大)

Tetsuya Taira, Keio University. Hiyoshi 3-14-1, Kouhoku-ku, Yokohama, Kanagawa
Nobuyuki Yamasaki, Keio University. Hiyoshi 3-14-1, Kouhoku-ku, Yokohama, Kanagawa

This paper explains the design of a two-armed upper-body humanoid robot's hardware system based on the Functionally Distributed Control Architecture. This architecture realizes real-time control of each function and integrates several functions into a robot system. The robot system based on this architecture is designed as a parallel/distributed computer, in which each module with an exclusive PU(Processing Unit) and I/O peripherals executes functionally distributed coarse granularity tasks in real-time, cooperating with each other. A prototype robot system 'RAITA' is developed as a testbed for humanoid robot based this architecture and a platform for the research on the motion planning and functions. 'RAITA' consists of Main module, Vision module, Communication module, Wheel module, Right-arm module, and Left-arm module. It is about 982 [mm] in height and has two 1-DOF wheels, two 6-DOF arms, and a 2-DOF head.

Key Words: Robot System Architecture, Distributed Real-Time Systems, Parallel/Distributed Control

1 序章

近年ロボットの開発が盛んに行われており、マニピュレータ、画像処理システム、音声処理システム、人工知能等様々な要素技術が研究・実現している。将来、人間の活動する環境下でロボットが自律的に行動するためには、これらの要素技術をシステムのリアルタイム性 [1] と処理の並列性を損わないように一つのシステムに統合して表現しなければならない。また、ヒューマノイドロボットのような大規模なロボットの開発において、様々なタスクの実行や機能の追加・変更に対応するには、拡張性・柔軟性を備えたシステムアーキテクチャが必要である。そこで我々は、これらの要求を満たすシステム構築を実現するために機能分散制御アーキテクチャを提案してきた [2]。

本研究の目的は、提案する機能分散制御アーキテクチャの有効性を自由度と機能が多く、リアルタイム性と処理の並列性が必要なヒューマノイドロボットを用いて検証を行うことと、ヒューマノイドロボットの動作計画や要素技術の研究を行うために、機能分散制御アーキテクチャに基づいた上半身型ヒューマノイドロボット 'RAITA' のハードウェアの設計・実装を行うことである。

以下では、まず 2 章で背景と関連研究について述べる。続く 3 章では提案する機能分散制御アーキテクチャについて説明する。4 章で機能分散制御アーキテクチャに基づいたロボットシステムの設計方針と段階の実装状況を述べ、5 章でまとめを行う。

2 背景と関連研究

高機能で様々なタスクを行えるロボットシステムを実現するためには、ロボット工学の別々の分野で研究・開発されてきた様々な要素技術を、1 つのシステムとして統合する必要がある。また、単に機能を統合するのではなく、ロボットの動作に支障をきたさないようにするためには、各機能の処理にリアルタイム性を保証する必要がある。この章では既存のロボットシステムの構成法とリアルタイム性に着目し、システムアーキテクチャについて検討を行う。

2.1 システム構成法

高機能で様々なタスクを行えるロボットシステムの代表例であるヒューマノイドロボットシステムには ASIMO[3], HERMES[4] などの複数のプロセッサを用いる分散制御型ロボットシステムと、H7[5], ROBOVIE[6] などの 1 つの高速なプロセッサを用いる集中制御型ロボットシステムの大きくわけて 2 種類の構成法がある。いずれの構成法も腹部に計算機を配置し、各部位へ配線する構造である。しかしながら、ロボットシステムの構造はセンサやアクチュエータなどの I/O がシステム内に分散して存在しているため、配線が大変であり、開発、故障時の対応には相当な時間が必要になるという問題点がある。また、モータ制御は、1 つのプロセッサ上で処理を行う集中制御型が主流であるため、各機能、要素技術を分離して研究・実装を行うのが困難であると言える。

2.2 リアルタイム性

システムの機能を変更したり、システムに機能を追加することにより、制御や処理に必要な計算量が増えてしまい、リアルタイム性を保証する制御が困難になってしまう問題がある。この問題点を改善するために様々なアーキテクチャが検討されてきた。分散制御型ロボットシステムである ROBITA[7], 山彦 [8] は、機能毎に固有のプロセッサを割り当て、それらの機能を情報共有バス等で結合することで、機能のリアルタイム制御を実現している。しかしながら、バス結合ではプロセッサ数が増えると、バスの調停が困難になりオーバーヘッドが非常に大きくなるため、システム全体としてのリアルタイム性を実現するのは困難であり、拡張性の点で問題があると考えられる。

一方、集中制御型ロボットシステムは、RT-Linux を利用し、比較的時間に余裕のあるタスクはユーザ空間、実時間タスクはカーネル空間と分離して処理することにより機能のリアルタイム制御を実現している。しかし計算機への負荷が大きく、すべての処理をボディ上で行うことができないためネットワーク資源を用いて処理を行ったり、機能を並列処理できないという欠点がある。

2.3 システムアーキテクチャの検討

システム構成や開発性の面から考えると、2.1の考察より、計算機資源を集めて配置したり、モータ制御の処理を集中制御するのではなく、複数の計算機をシステム内に分散させて配置し、機能の処理とモータ制御の両方を局所的に行い、それらの計算機をネットワーク化することでロボットを構築する並列分散制御型システムが適していると考えられる。例えば、マニピュレータ部は、モータ制御とセンサ処理に必要な計算機を腕部に持ち、局所的な制御ループを形成するマニピュレータモジュールとして実装され、ネットワークを介して他のモジュールと協調してタスクを行う。このように分割することで機能を一つのシステムとして開発でき、高機能な機能モジュールが実現可能であると考えられる。また、省配線化、電子部の簡略化、故障時の対応が容易であるなどの利点もある。

リアルタイム性や計算機資源の面から考えると、2.2の考察より、ヒューマノイドロボットのような高機能のシステムを一つのプロセッサでリアルタイム制御するのは、情報処理の負荷が著しく大きいため実現が困難である。そこで、各機能に固有のプロセッサを割り当て、並列処理を行うことで、機能毎のリアルタイム性、システム全体のスループット向上を実現する必要がある。

以上より、ロボットシステムを局所的にアクチュエータやセンサなどのI/Oを制御するモジュールの集まりとして設計することで、機能の追加や拡張に対して柔軟に対処することができる。開発時間、計算能力の制約を受けず、リアルタイム制御や動作の並列性を実現できる統合システムを構築するには、それらのモジュール毎におおの必要リアルタイム性を満たすことができるPU (Processing Unit) を割り当て、これらのモジュールをリアルタイム性を保証できるネットワークで結合するシステムアーキテクチャが有効であると考えられる。

3 機能分散制御アーキテクチャ

我々は、2章で議論した要求を実現するために、機能分散制御アーキテクチャを提案・設計してきた。Figure.1にアーキテクチャの概要図を示す。

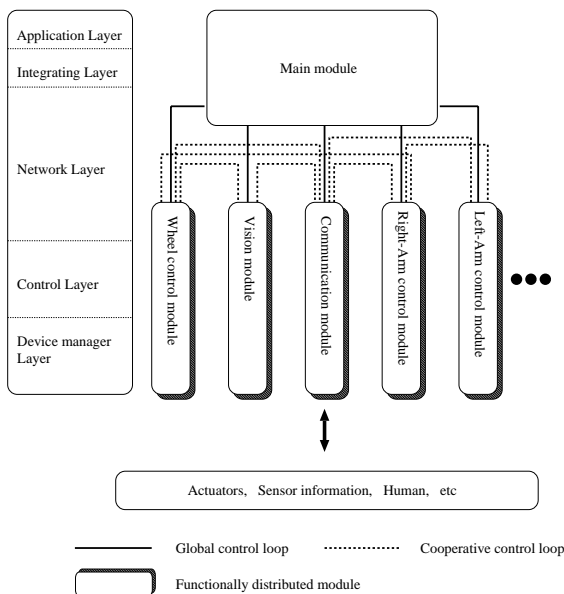


Figure 1: Architecture Overview

機能分散制御アーキテクチャに基づいたロボットシステムは、1つのメインモジュールと複数の機能分散モジュール群から構成

する並列分散コンピュータとして実装される。これらのモジュールは各機能毎にリアルタイム性を保証する処理能力のある固有のPUを持つ。メインモジュールは機能分散モジュール間の調停、システム全体としてのプランニング、学習を行うモジュールとして用いる。ただし、メインモジュールは、ロボットシステム全体を調停しているが、モジュール個々のタスク処理には全く関与しない。機能分散モジュールは一つの機能を実現するモジュールである。これらの機能毎に局所的に制御を行っている複数のノードコントローラがネットワークを介して互いに通信を行いながらシステム全体を制御し、機能の統合を行う。モジュール間の処理・協調は以下の3つの制御機構により行われる。

- Local Control :
機能分散モジュール内に閉じた1つの機能をリアルタイム制御する局所的な制御機構。割り当ててある固有のプロセッサ上で制御を実現する。
- Global Control :
メインモジュールのプランニングによるロボットシステム全体を制御する大局的な制御機構。各機能からのイベントに対してメインモジュール内のGlobal Plannerが機能分散モジュールの環境認識の結果から作成するマップとシステムの状態を表すデータベースを参照することで、各機能分散モジュールへタスクを依頼することで実現する。
- Cooperative Control :
複数のモジュールがメインモジュールのプランニングなしに自律的に協調してタスクを実行する制御機構。反射的行動やGlobal Controlによる行動と並列な動作実行を実現する。メインモジュールの負荷軽減や耐故障性の役割もする。

機能分散制御アーキテクチャの階層モデルの詳細を Table.1 に示す。

Table 1: 機能分散制御アーキテクチャの階層モデル

名称	概要
Application Layer	ロボット全体のアプリケーションを定義するユーザ空間レイヤ
Integrating Layer	Global Control 機構によりシステム全体のプランニングやマネジメントを行うレイヤ
Network Layer	Global Control 機構, Cooperative Control 機構を実現するためにシステム内のネットワーク構築をサポートするレイヤ
Control Layer	Local Control 機構により画像処理システムや音声処理システムなどの機能・要素技術を実現するレイヤ
Device Manager Layer	機能分散モジュール内のアクチュエータやセンサなどのI/Oデバイスを直接制御するレイヤ

階層モデルを考えることで機能別だけでなく、各層を個別に設計・実装することができ、ある層だけ入れ替えることが可能となり、機能の部品化が容易となる。階層モデルに従った設計をすることにより、異なるロボットおよび、ロボット部品間での相互運用性が向上でき、新しい機能の追加や変更を容易に実現することができる。

マルチプロセッサ構成のシステムにおいて問題となるモジュール間通信は、リアルタイム性・拡張性を重視したネットワーク通信を用い、ソフトウェア上で通信プロトコルを統一することにより、機能やインタフェースに透過なネットワークを実現す

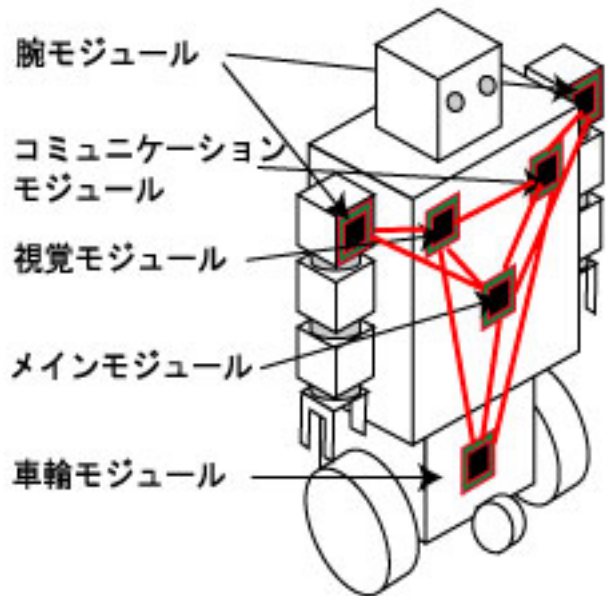
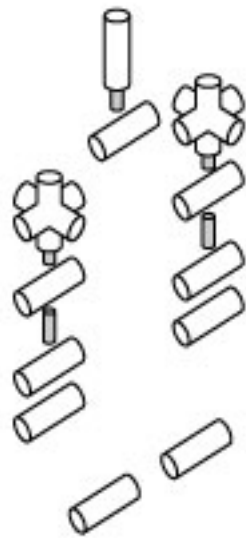


Figure 2: Robot System "RAITA"

る。情報共有・メッセージ通知には、通常処理と反射処理を統一に取り扱うために、割り込みベースの通信を行う。

本アーキテクチャは車輪型ロボット P1, 歩行ロボット P2 の実装・実験により有効性が検証されてきた [9]。

4 上半身型ヒューマノイド:RAITA

機能分散制御アーキテクチャに基づいて、Figure.2:left に示す上半身型ヒューマノイド'RAITA'のハードウェアを設計・開発した。本ロボットは、提案する機能分散制御アーキテクチャの有効性を自由度と機能が多く、リアルタイム性と処理の並列性が必要なヒューマノイドロボット上で検証を行うことと、ヒューマノイドロボットの動作計画や要素技術の研究を行うことを目的として設計・実装した。

ロボットのボディは、CAD ツール AMADA AP100 を用いて設計し、NC ターレットを用いて加工した。寸法は、各方向の最大長さで、高さ 982[mm]、幅 715[mm]、奥行き 500[mm]である。自由度配置は、車輪部 2 自由度、腕部各 6 自由度、頭部 2 自由度の計 16 自由である Figure.2:center に自由度の構成を示す。

4.1 システム構成

上半身型ヒューマノイド RAITA のシステム構成を Figure.2:right に示す。ロボットシステムはメインモジュール、車輪モジュール、腕モジュール 2 つ、視覚モジュール、コミュニケーションモジュールの計 6 つの機能分散モジュールから構成される。計算機資源は Figure.2:right に示すように体内に分散して配置する。各機能の処理や制御はモジュール内に閉じており、各モジュールとも自律的に処理を行うことができる。機能毎の開発や実験、補修を容易にするために、モジュール間の接続部分はモジュールの分離と組み立てを容易に行える設計にしてある。

これらのモジュールを用いて並列分散制御を実現するためには、各モジュールの計算機間の通信にリアルタイム通信が必要不可欠である。従って、通信インタフェースには、並列分散制御用のリアルタイム通信規格である Responsive Link[10]を採用する。

各モジュールの概要は以下である。

メインモジュール

メインモジュールは機能分散モジュール間の調停、システム全体としてのプランニング、学習を行うモジュールである。モジュールコントローラには SPARC:100MHz をコアとし、Memory:16MByte、各種コンピュータ用周辺、各種制御用周辺など多数の機能を集積した組込用途のシステムオンチップである Responsive Processor[11]を使用する。OS は Responsive Processor 上で動作する実時間 OS である RT-Frontier[12]を使用する。Global Control を行うために、視覚モジュールとは PCI バスで、他のモジュールとは Responsive Link を用いて接続する。

車輪モジュール

車輪モジュールはロボットをナビゲーションする機能分散モジュールである。自由度の配置は、各車輪 1 自由度の計 2 自由度であり、maxon 社の RE シリーズ 10W モータを採用している。モジュールコントローラは Responsive Processor であり、OS は RT-Frontier を使用する。モジュール内の超音波センサ、赤外線センサ、タクトイルセンサ、ジャイロセンサからの情報を基にナビゲーション、障害物回避を自律的に実現する。他のモジュールからの命令に応じたナビゲーションを行うために、メインモジュール、視覚モジュール、コミュニケーションモジュールと接続する。

腕モジュール

腕モジュールはロボットのマニピュレータ部分を担う機能分散モジュールである。自由度の配置は、胴体正面での作業空間を確保できるよう設計してあり、両腕とも肩関節 3 自由度、肘関節 1 自由度、手首関節 2 自由度の計 6 自由度ずつである。モータ出力は 2Kg 程度の重い物でも持ち上げられるように、肩関節、肘関節の自由度には maxon 社の RE シリーズ 70W モータを採用し、手先位置の精度向上のためにハーモニックドライブを出力軸が間接軸になるように設計し、ノンバックラッシュ性を高めている。モジュールコントローラとして片腕毎に Responsive Processor を持ち、OS は RT-Frontier を使用する。腕モジュールは、他モジュールからの命令に応じてモジュール内のセンサ情報を基にモーションを行う。メインモジュール、視覚モジュール、コミュニケーションモジュールと接続する。

視覚モジュール

視覚モジュールは視覚情報に関する処理を行う機能モジュールであり、現在開発中である。自由度の配置は視野を拡大し足元を見ることが出来るように設計しており、2自由度である。maxon社のREシリーズ10Wモータを採用している。この自由度を視覚モジュールが制御を行うことで、対象物のトラッキングや障害物までの距離測定などを自律的に行なうことができる。画像処理は計算コストが大きいため、モジュールコントローラにはCPU:Pentium4 2.8GHz, Memory:1.0GHzを搭載するMicroATX規格のマザーボードを使用し、OSはLinux(Kernel-2.6.1)を使用する。カメラはIEEE1394により制御用プロセッサと接続しており、得られる画像を高速転送する。PCIバスにResponsive Processorを配置し、NIC(Network Interface Card)として使用することで、他モジュールとの実時間通信を実現する。メインモジュールとはPCIバスで、腕モジュール、車輪モジュールとはResponsive Linkを用いて接続する。3次元距離測定、顔認識等を行なう予定である。

コミュニケーションモジュール

コミュニケーションモジュールは音声認識、音声合成を行う機能モジュールである。音声認識・合成も比較的計算コストが大きいため、モジュールコントローラにはCPU:Pentium3 750MHz, Memory:768MHzを搭載するMicroATX規格のマザーボードを使用し、OSはLinux(Kernel-2.4.20)を使用する。無線LANを介してネットワーク資源と接続でき、ネットワーク上の情報を参照することができる。PCIバスにResponsive Processorを配置し、NICとして使用し、メインモジュールとはPCIバスで、腕モジュール、車輪モジュールとはResponsive Linkを用いて接続する。本モジュールは現在マイク入力を基にした音源定位を実装中である。

4.2 ソフトウェアシステム

現在はシステムのハードウェア部分が完成した段階であり、各機能のソフトウェアは実装段階である。機能分散制御アーキテクチャに基づいてロボットシステムの設計を行っているので、機能別、階層別の設計・開発や、今までに作成したプロトタイプロボットのソフトウェアや既存のシステム・アルゴリズムをそのまま使用することが容易であるため、開発時間の大幅な削減を見込むことができる。

5 まとめ

リアルタイム制御、処理の並列性を実現できる機能分散制御アーキテクチャに基づいて、上半身型ヒューマノイドロボット'RAITA'のハードウェアを設計・開発した。本ロボットの計算機システムは、各機能毎にリアルタイム性を保証する計算機を持ち、リアルタイム性を保証する通信インタフェースで機能をネットワーク化した大規模な並列分散コンピュータであり、処理の並列性の実現やヒューマノイドロボットの動作計画や要素技術の研究を行うために十分なシステムを構築できたと考えられる。また、アーキテクチャに基づいて設計することで、複数の機能の同時開発、システム構築の容易性等を実現した。

今後はモジュール間を制御するGlobal Control, Cooperative Controlに関するソフトウェアを構築し、視覚による環境認識に基づく環境内の自律行動やインタラクションなどの行動計画、各要素技術の研究等を行っていく予定である。また、視覚モジュールや、コミュニケーションモジュールに関して、省スペース、低消費電力化、軽量化を行うために、パフォーマンスが高く、低消費電力なResponsive Multithreaded Processor[13]やカードPCなどの組み込み用プロセッサを用いることで改善したいと考えている。

謝辞

本論文の研究は、高橋財団の予算を使用させて頂きました。ロボットの開発において駒場製作所の駒場武志氏に貴重なご意見を頂きました。ここに深く感謝を表します。

References

- [1] John A. Stankovic. Misconceptions about real-time computing. *IEEE Computer*, pp. 2–10, 1988.
- [2] 平哲也, 高田和泉, 山崎信行. 自律移動ロボットのための機能分散制御アーキテクチャ. ロボティクス・メカトロニクス講演会'03講演会論文集, pp. 1–2, 2003.
- [3] Yoshiaki Sakagami, Ryujin Watanabe, Chiaki Aoyama, Shinichi Matsunaga, Nobuo Higaki, and Kikuo Fujimura. The intelligent ASIMO: System overview and integration. In *IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, pp. 2478–2483, 2002.
- [4] R.Bischoff and V.Graefe. Integrating, Vision, Touch and Natural Language in the Control of a Situation-Oriented Behavior-Based Humanoid Robot. In *IEEE International Conference on Systems, Man, and Cybernetics.*, pp. 999–1004, 1999.
- [5] Satoshi Kagami, Koich Nishiwaki, James Kuffner, Yasuo Kuniyoshi, Masayuki Inaba, and Hirochika Inoue. Online 3D Vision, Motion Planning and Bipedal Locomotion Control Coupling System of Humanoid Robot : H7. In *IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, pp. 2557–2562, 2002.
- [6] 神田崇行, 石黒浩, 小野 哲雄 今井倫太. 人間と相互作用する自律型ロボット Robovie の評価. 日本ロボット学会誌, pp. 1–9, 2002.
- [7] 於久健太郎, 清水健二, 松坂要佐, 小林哲則. マルチモーダルロボット用マルチプロセッサアーキテクチャ. 情報処理学会 情報処理学会計算機アーキテクチャ研究会, pp. 47–52, 2002.
- [8] 鈴木晃, 坪内幸司, 油田信一. 自律移動ロボット「山彦」アーキテクチャのPC/ATへのシステム実装. 第19回日本ロボット学会学術講演会, pp. 1103l–1104, 2001.
- [9] Tetsuya Taira and Nobuyuki Yamasaki. Functionally Distributed Control Architecture for Autonomous Mobile Robots. *Journal of Robotics and Mechatronics*, Vol. 16, No. 2, 2004.
- [10] Responsive Link(IPSJ-TS 0006:2003). <http://www.itsecj.ipsj.or.jp/ipsj-ts/02-06/toc.htm>.
- [11] Nobuyuki Yamasaki. Responsive processor for parallel/distributed real-time control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1238–1244, 2001.
- [12] Hidenori Kobayashi and Nobuyuki Yamasaki. An Integrated Approach for Implementing Imprecise Computations. *IEICE transactions on Information and Systems*, Vol. E86-D, No. 10, 2003.
- [13] Nobuyuki Yamasaki. Design Policy of Responsive Multithreaded Processor for Distributed Real-Time Control. *Journal of Robotics and Mechatronics*, Vol. 16, No. 2, 2004.