

# 機能分散モジュールからなる モジュール型ヒューマノイドロボットの構築

平 哲也      山崎 信行

慶應義塾大学

**Abstract**— 本論文では、機能分散モジュールの組み合わせによりモジュール型ヒューマノイドロボットを構築する。ロボットは、アーム制御モジュール、車輪制御モジュール、視覚処理モジュールなどのような機能分散モジュールによって構成し、各モジュールはハードウェア、通信インタフェース、ソフトウェア等から構成する。機能分散モジュールを用いたロボットシステムの構築は、ヒューマノイドロボットのリアルタイム性の保証、機能間の円滑な情報共有のための体内リアルタイムネットワーク機能、ロボットの機能の追加・変更の容易性、拡張性の提供を目的として機能別並列分散制御アーキテクチャに基づいて行う。提案するアーキテクチャの評価、ヒューマノイドロボットの動作計画、要素技術の研究を行うために、6つの機能分散モジュール、16自由度からなる上半身ヒューマノイドロボットを構築する。

**Key Words:** ロボットアーキテクチャ、リアルタイムシステム、並列分散制御、機能分散モジュール

## Development of Modular Humanoid Robot System Composed of Functionally Distributed Modules

Tetsuya TAIRA      Nobuyuki YAMASAKI

Keio University

**Abstract**— This paper explains the development of a modular humanoid robot system composed of several functionally distributed modules. The robot system consists of the functionally distributed modules, for example, arm module, wheel module, and vision module. These modules compose of software, hardware, and network interface. The development is based on the functionally parallel/distributed control architecture that achieves real-time performance, real-time network for data sharing between modules, and extensibility. We implement a prototype robot composed of six functionally distributed modules and 16-DOF for evaluation and the research on the motion planning and functions.

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

### 1. 序章

人間の活動する環境下で自律的に行動するヒューマノイドロボットを実現するためには、視覚、触覚、音声対話、アーム制御、二足歩行、ナビゲーション、モーションプランニングなどのこれまで個別に研究されてきた要素技術を統合するソフトウェア、ハードウェアの両面を考慮したシステムアーキテクチャが必要となる。ヒューマノイドロボットに必要なソフトウェア

はモータ制御や画像処理のようなI/Oを用いるハードリアルタイムタスクから、長期的な行動計画やインタラクションといったソフトリアルタイムタスクまで、数多くの並列プロセスの集合として構成される。また、ヒューマノイドロボットに必要なハードウェアは、対象とするアプリケーション・動作を実現するために必要なI/OとそのI/Oを適切に制御するためのCPUやメモリから構成される。

近年、リアルタイムOSとPCのハードウェアが目

覚ましく発展し、これらを単一のシステムとして実現可能になってきている。しかしながら、PCによる制御は、サイズ、消費電力、配線、冗長さの不足等の様々な原因から、ヒューマノイドロボットには不向きである。非PCの並列分散システムとしてヒューマノイドロボットを構築することがこの問題を解決する一つの方向性であると考えられる。そこで我々は、ハードウェア、通信インターフェース、ソフトウェア等から構成する機能分散モジュールを複数組み合わせることでロボットシステムを構築するモジュール型ヒューマノイドロボットのための機能別並列分散制御アーキテクチャを提案している [1]。

本論文では、右アームモジュール、左アームモジュール、車輪モジュール、視覚システム、コミュニケーションモジュール、メインモジュールの6つの機能分散モジュールからなるモジュール型ヒューマノイドロボットのシステム構成法とその有効性について述べる。

以下では、まず2章で研究の背景について述べ、3章では提案するモジュールロボットアーキテクチャについて説明する。4章で機能分散モジュールからなるモジュール型ヒューマノイドロボットの構築について述べ、5章で本論文のまとめを行う。

## 2. 研究の背景

### 2.1 関連研究

実世界で行動するヒューマノイドロボットのソフトウェアシステムは、二足歩行、アーム制御に必要なアクチュエータ・センサの実時間制御ループを処理しながら、視覚処理やモーションプランニングなど高次の知能処理を行わなければならない。そのためには1msec程度の様々なサーボループ内で運動学や動力学を解きながら、メモリやCPUを大量に消費する高次の処理を行う必要がある。これらの要求を満たすために、現在のロボットシステムでは、1つのプロセッサが全てを制御するという事は稀であり、多かれ少なかれ複数のプロセッサによって分散制御を行なっている。

HRP-2[2]は、Pentium3 1.26GHzを2つ用いて歩行制御とその他の制御を分離したPCベースの負荷分散型システムである。システムは商用のPCとI/Oボードのみから構成されており、ソフトウェアの開発性が高いと言える。しかし、集中システムのために、機能拡張が物理的に複雑でコストがかかる、配線やネットワークのロバスト性にかかる問題点がある。また、PC間通信にEthernetを用いているために、通信遅延が予測できない点も問題として挙げられる。

一方、HERMES[3]はDSP(TMS 320C40)を複数個胴体に配置し、視覚処理、音声処理を行う。運動制御は、CANバスで情報共有を行いながら、アクチュエータ毎に持つコントローラで局所的に行う。バス通信は、接続された全てのモジュールに非同期にデータを送信でき、柔軟な情報共有を行うことができる。しかしながら、モジュール数の増加に従ってバスの調停が困難になるため、システム全体のリアルタイム性の低下、拡張の制限などの問題点が挙げられる。

ASIMO[4]は画像処理、音声認識、運動制御毎に計算機を割り当て胴体に配置した機能分散型システムであり、体内ネットワークを用いてアクチュエータの分散処

理を行っている。ASIMOはHRP-2とHERMESのハイブリッドのようなシステム構成であるが、すべてのアクチュエータの行動計画を1つの計算機で集中的に処理するため、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り、拡張性の点で問題があると考えられる。

以上より、集中型システム、配線、冗長さの不足、リアルタイムネットワーク機能の欠如などのデメリットが既存のロボットシステムが採用しているアーキテクチャや機能統合の構成法の限界として問題となっている。実時間制御機能を持つ並列分散システムがこの問題を解決する一つの方向性であると考えられる。

### 2.2 ヒューマノイドロボット用アーキテクチャへの要件

ロボット用並列分散システムアーキテクチャに用いるハードウェア、ソフトウェア、体内ネットワークとして、以下の要件が挙げられる。

- コンテキストスイッチのオーバーヘッドの小さいハードリアルタイムOS機能
- マルチタスク、マルチユーザー、プロセス間通信、ネットワークなどの汎用OS機能
- エディター、コンパイラ等のソフトウェア開発環境
- 分散ノードコントローラ間の実時間通信機構
- 分散ノードが担う機能のリアルタイム性を保証できるCPUとメモリ

これらの要求を考慮しつつ、分散制御を行うためには、I/Oの種類/数量、空間的な配置問題、演算量、通信量、リアルタイム性を指標としてシステム全体をどのようにして分割するのかを決定する必要がある。

本研究では、既存のシステムが問題点とするリアルタイム性、拡張性、空間的な配置問題の解決を重視したシステム分割を行うために、ソフトウェア開発環境よりも、実時間OSのオーバーヘッドの低さ、実時間通信機構、組込みシステムの採用を優先し、レスポンスプロセッサ、RT-Frontier、レスポンスリンクを選択することにした。

### 2.3 レスポンスプロセッサとRT-Frontierの概要

PCベースのシステムで問題となる計算機のサイズ、消費電力に関する問題を解決するために組込み用途のプロセッサを用いる。そこで、各分散ノードコントローラには、SPARC:100MHzをコアとし、Memory:16MByte、多数の制御I/Oやコンピュータ周辺I/Oを搭載したシステムオンチップであるレスポンスプロセッサ[5]を用いる。レスポンスプロセッサはA/Dコンバータ(8ch)やD/Aコンバータ(2ch)、PWMジェネレータ(9ch)、パルスカウンタ(9ch)、PIO(16ch)を有しているため、ロボット制御用には十分である。

また、分散ノードコントローラを制御するためのリアルタイムOSには、レスポンスプロセッサ上にマイクロカーネル型の分散実時間オペレーティングシステムであるRT-Frontier[6]を用いる。RT-Frontierは1msecを単位時間としたスケジューリングを行うことができ、タスクの実行時間は100 $\mu$ secを単位として管理することが可能である。

ノード間の円滑なデータ共有, リアルタイム通信を実現するための通信インタフェースには, レスポンシブリンクを用いる。ロボットに用いられている I/O 及び通信インタフェースとして, ISA, PCI, Ethernet, USB[8], CAN(Controller Area Network) バス [9], I2C バス [10] などの標準規格がある。ISA や PCI は, バックプレーンボードが必要であり, 配置の制約がある。CAN バス・I2C バスは, 高速性を満たさない。Ethernet では, 通信の際の衝突があった場合に再送するプロトコルがあるために, 最悪通信時間を予測することができずリアルタイム性がないと言える。また, USB は, トポロジーがループを許さないツリー構造のみであるため耐故障性, 拡張性の点で問題があると言える。レスポンシブリンクでは, レイテンシとスループットのトレードオフを考慮してソフトリアルタイム用のデータ(データリンク)とハードリアルタイム用のイベント(イベントリンク)が分離されているという特徴があり, レイテンシの保証, バンド幅の保証を行う。また, パケットの衝突による予測性の低下を避けるために結合形態は Point-to-Point となっている。レスポンシブプロセッサは 4 対のレスポンシブリンクを備えており, トポロジーフリーで柔軟なネットワークを構築でき, 計算機資源や機能モジュールの拡張に物理的制約を与えない高い拡張性が実現可能である。

これらの機能を持つレスポンシブプロセッサ, RT-Frontier, レスポンシブリンクをベースとして分散制御システムを構築することで, 計算機資源や機能の拡張に物理的制約をほとんど与えない高い拡張性, 実時間 OS のオーバーヘッドの低さ, 実時間通信機構を実現することができる。

### 3. 機能モジュールから構成するシステムアーキテクチャ

#### 3.1 機能分散モジュールの設計及び設計方針

我々が提案しているヒューマノイドロボット用の機能別並列分散システムアーキテクチャの主な特徴を Table 1 に, 構成図を Fig.1 に示す。

Table 1 提案するシステムアーキテクチャのコンセプト

システムの分割粒度	機能モジュールに分割
コントローラの配置	機能毎にモジュール化
ソフトウェア構成	階層構造の機能別並列分散制御アーキテクチャ
コントローラ間通信機構	レスポンシブリンクを用いたリアルタイムネットワーク

アーム制御や視覚処理などのリアルタイム性を考慮すると, Fig.1 の Arm Module や, Vision Module が示すように, ロボットの一機能に必要なデバイス, リアルタイム性を保証する計算機資源, リアルタイム OS を合わせて 1 つのモジュールとして設計し, システム全体をモジュールの集合体として構築する手法が適していると考えられる。

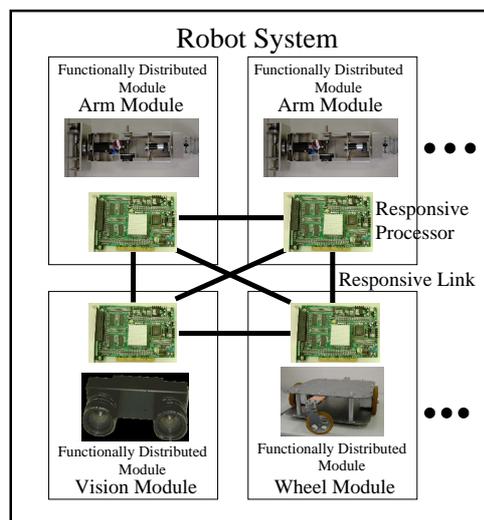


Fig.1 アーキテクチャ構成図

モジュール化の分割粒度であるが, 機能の拡張や変更に伴うシステム全体に与えるコストや開発性を考慮すると, 既存の研究を大きく改良することもなくシステムに組み込むことができる, アーム制御部, 視覚処理部, 歩行制御部といった機能別の分割が適していると考えられる。機能毎にシステムを構築できるため, 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。

以上より, ロボットを機能別に分割・モジュール化し, これを機能分散モジュールとして定義する。モジュールに含まれる構成要素は次の通りである。

- 機能分散モジュールのハードウェア
  - センサ・アクチュエータなどの各種デバイス類
  - I/O 制御およびシステム管理のための PU (レスポンシブプロセッサ)
  - モジュール間のデバイス制御やデータ交換のための通信インタフェース (レスポンシブリンク)
  - 負荷分散や I/O 処理などの必要に応じたサブコントローラ
- 機能分散モジュールのソフトウェア
  - PU 上で動作するリアルタイム OS (RT-Frontier)
  - モジュールの I/O 制御のためのデバイスドライバ
  - 各モジュールで提供する機能を実現するアプリケーションプログラム
  - リアルタイムネットワーク通信機構

#### 3.1.1 システム全体のソフトウェア構成

機能分散モジュールを組み合わせるヒューマノイドロボットのシステム制御のリアルタイム性の保証, システム全体を用いたモーションプランニング, 機能の追加や変更の容易性を提供するために, システム全体を Fig.2 に示す階層構造の機能別並列分散制御アーキテクチャに従って設計する [1]。

ロボットシステムは, 提案するアーキテクチャに基づ

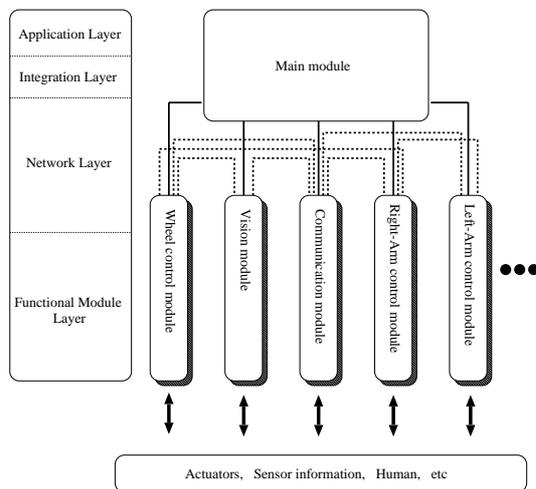


Fig.2 提案するソフトウェアアーキテクチャ

いて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に分類される。機能分散モジュールは、各モジュール独自のスケジューリングポリシーに従ってアクチュエータ制御・センサ処理、画像処理などを行うタスク処理機構部と、後述する通信処理機構部から成る。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を行うモジュールとして用いる。メインモジュールは以下の2つのパートから成る。

- システム管理パート  
周期的に機能モジュールと通信を行い、システムの安定性をハードリアルタイムに保証する。
- 高次な制御タスク管理パート  
長期的な行動計画、対話、インタラクション等のソフトリアルタイムタスクを非周期的に処理

各階層の概要は以下である。

- Application Layer  
ロボット全体のアプリケーションやタスクを定義するレイヤ
- Integration Layer  
機能分散モジュールの提供する機能を統合し、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  
モジュール間の協調制御を実現するネットワークをサポートするレイヤ。メインモジュールと機能分散モジュール間を接続するグローバルネットワークと、機能分散モジュール間を接続するローカルネットワークを形成する。
- Functional Module Layer  
機能分散モジュールが画像処理システムやアームなどの機能を提供するレイヤ

### 3.1.2 モジュール間のリアルタイム通信機構

モジュール間のリアルタイムネットワーク機能は、ネットワーク層が制御する。ロボット内ネットワークとしてメインモジュールと機能分散モジュール間を接続するグローバルネットワークと、機能分散モジュール間を接続するローカルネットワークを形成する。グロー

バルネットワークは、メインモジュールがシステム全体の状況を判断して、安定したシステム制御や長期的な行動計画を行うためのネットワークである。ローカルネットワークは、collision-freeな腕の制御や緊急停止などを実現するために、複数の機能分散モジュールが必要に応じて形成する協調制御ネットワークである。メインモジュール以外のモジュールが自律的に判断し協調ネットワークを持つので、メインモジュールの負荷分散、緊急時や制御に対するロバスト性、システム全体のスループットの向上が行える。グローバルネットワークとローカルネットワークを用いた制御例としてアーム制御を挙げる。グローバルネットワークにより1sec周期でアームモジュールの行動を計画する。その計画に対して、腕モジュールと視覚モジュールのローカルネットワークにより100msec周期のビジュアルフィードバックループを形成し、腕の目標位置を決定する。その位置に対してアーム制御同士のローカルネットワークにより10msec周期でcollision-freeな軌道を決定する。計画された軌道に対して腕モジュールのコントローラがセンサ値を元に1msec周期で位置制御・速度制御を行う。以上のように、グローバルネットワーク、ローカルネットワークを用いることで、既存の集中システムに比べ、システム全体のロバスト性や情報処理能力の向上が可能である。

モジュール間メッセージには、制御用途に応じて同期メッセージであるEvent Message、非同期メッセージであるState Message、Signalの通信メッセージを用いる。主な用途は以下の通りである。

- センサ・アクチュエータや画像処理などの時間駆動系タスクグループは資源予約に基づき実時間通信のQoSを保証するReal-Time Channel Manager[11]を利用し、イベントリンクを用いたState Messageにより他のモジュールと周期的に通信しながら協調して処理を行う。緊急時にはイベントリンクを用いたSignalを使用することで対応する。
- 音声認識や音源定位などの事象駆動系タスクグループはイベントリンクを用いたEvent Messageにより他のモジュールと通信しながら処理を行なう。

各機能モジュールを新たにシステムへ組み込む場合には、(1) 機能モジュールが提供する機能、APIを他のモジュールに公開する、(2) 追加するモジュールに、他のモジュールへのアクセス、他のモジュールからのアクセスを可能にするためにリアルタイムネットワーク機構の追加を行う。これらの2点を定義することにより、モジュール間でデータ入出力の依存関係や動作を妨げない情報共有、比較的低コストでソフトウェアの変更、モジュールの組み合わせが可能となると考えている。

## 4. モジュール型ヒューマノイドロボットの設計と実装

### 4.1 モジュール型ヒューマノイドロボット

機能分散モジュールから構築するモジュール型ヒューマノイドロボットのシステム構成をFig.3に示す。ロボットシステムはメインモジュール、車輪制御モジュール、腕モジュール(2個)、視覚モジュール、コミュニケーションモジュールの5つの機能モジュールの計6個の

モジュールから構成される。各モジュールの詳細は後述する。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 1000[mm]、幅 715[mm]、奥行き 500[mm] である。各機能モジュールの制御用計算機資源は Fig.3 に示すように各機能分散モジュール毎に配置する。各機能の処理や制御はモジュール内に閉じていおり、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。

## 4.2 各機能分散モジュール

### 4.2.1 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を行うモジュールである。モジュールコントローラにはレスポンスプロセッサ、OS には RT-Frontier を使用する。システム的全ジュールとレスポンスリンクを用いて接続する。

### 4.2.2 車輪モジュール

車輪モジュールはモジュール内の超音波センサ、赤外線センサ、接触センサからの情報を基にナビゲーション、障害物回避、緊急停止を自律的に実現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 18W モータを採用している。モジュールコントローラはレスポンスプロセッサであり、OS は RT-Frontier を使用する。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レスポンスリンクを用いて接続する。ソフトウェアは、メインモジュール、視覚モジュールからの指令値を元に、エンコーダ、ジャイロセンサの値を参照して 1msec の周期で位置制御・速度制御を行い、10msec の周期でカルマンフィルタを用いて自己位置推定を行う。

### 4.2.3 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 モータを採用する。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レスポンスプロセッサを持ち、OS は RT-Frontier を使用する。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レスポンスリンクを用いて接続する。ソフトウェアは、メインモジュールやもう一方の腕モジュールと 10msec の周期で通信を行い、与えられる指令値と運動力学の結果を基にエンコーダの値を参照にして 1msec 毎に位置制御・速度制御を行い、collision free な把持や握手等の高度なモーションを行う。

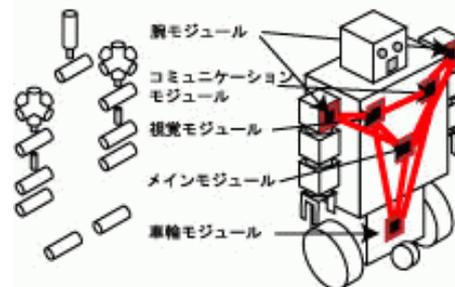


Fig.3 プロトタイプロボット

### 4.2.4 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あり、2 自由度である。画像処理は計算コストが大きいので、モジュールコントローラにはレスポンスプロセッサを使用し、サブコントローラとして CPU:Pentium4 2.8GHz、Memory:1.0GByte を搭載する MicroATX 規格のマザーボードを使用し、OS は Linux を使用している。カメラは CANON VC-C3 を利用している。レスポンスプロセッサとサブコントローラは PCI を用いて、メインモジュール、腕モジュール、車輪モジュールとはレスポンスリンクを用いて接続する。ソフトウェアは、対象物のトラッキングや肌色検出、顔認識等を行う。

### 4.2.5 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、音声出力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ので、モジュールコントローラにはレスポンスプロセッサを使用し、サブコントローラとして CPU:Pentium3 750MHz、Memory:768MByte を搭載する MicroATX 規格のマザーボードを使用する。OS には Linux を使用する。レスポンスプロセッサとサブコントローラは PCI を用いて、メインモジュール、腕モジュール、車輪モジュールとレスポンスリンクを用いて接続する。ソフトウェアは、HMM を元に音声認識や音声出力を行っている。

## 4.3 プロトタイプロボットの基礎評価

プロトタイプロボットを用いて人間の方へ向き、握手を促す行動を示すという基礎的な実験を行った。メインモジュールからの要求で車輪制御モジュールが人間のいる方向へ向き (Fig.4 左)、人間を確認したら腕モジュールとコミュニケーションモジュールにタスクを要求し握手を行う (Fig.4 右) アプリケーションである。

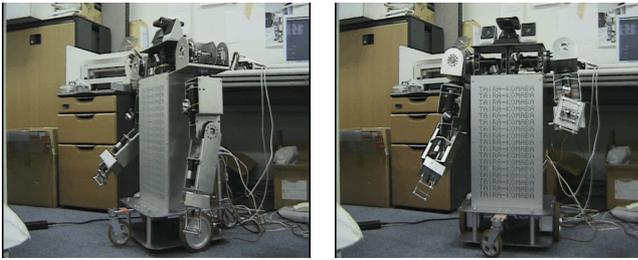


Fig.4 (左) 車輪制御モジュールによる制御 (右):メインモジュール, 腕モジュール, コミュニケーションモジュールによる協調制御

アプリケーション実行時の制御ループの周期, リアルタイムネットワークの通信遅延は以下の Table 2, 3 通りである.

Table 2 制御ループ

	msec
メインモジュールのシステム管理部	1.0
車輪, 腕モジュールの位置・速度制御	1.0
車輪モジュールの自己位置推定	10.0
メインモジュールと両腕モジュールの協調制御	10.0
メインモジュールと車輪モジュールの協調制御	10.0

Table 3 ソフトウェアオーバーヘッドとハードウェアオーバーヘッドを含めた通信遅延時間

通信速度	40 MBaud
データサイズ	16 Byte
通信遅延	123 $\mu$ sec

機能毎のハードウェア・ソフトウェアのモジュール化, 体内リアルタイムネットワーク通信機能により, 機能毎のリアルタイム性の保証だけでなく, 機能間のリアルタイム性保証も行うことができる. 機能間のリアルタイム性保証により, 様々なモジュール間制御ループを形成することができ, 情報処理能力の向上, システムのロバスト性の向上が可能であると言える.

## 5. まとめ

本論文では, 機能分散モジュールの組み合わせによるモジュール型ヒューマノイドロボットの構築について述べた. 機能分散モジュールを用いたロボットシステムは, 提案している機能並列分散制御アーキテクチャに基づいて構築された, 機能や通信のリアルタイム性の保証をする分散制御型システムである. 構築したロボットは従来のシステムに比べ, 様々なモジュール内/間の制御ループを形成することができるので, 情報処理能

力の向上, システムのロバスト性の向上が可能であると言える. さらに, 機能毎の開発性, 配線長も最小限に抑えることが可能であり, 従来の PC による制御で問題となるサイズ, 配線, 冗長度の不足等の様々な原因を改善することができた. 非 PC の並列分散システムがヒューマノイドロボットを構築する際の問題を解決する一つの方向性であることを示せた. 今後は, 現在視覚モジュール等で使用しているサブコントローラも非 PC 化することで, さらなる改善を行って行く予定である. また, 同じ機能をもった集中型構成のヒューマノイドロボットを構築し, 定量的な評価・比較によりアーキテクチャの検証を行う予定である.

## 謝辞

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.

## 参考文献

- [1] T. Taira and N. Yamasaki: "Functionally Distributed Control Architecture for Autonomous Mobile Robots", Journal of Robotics and Mechatronics, Vol. 16, No. 2, pp. 217-224, 2004.
- [2] 五十棲隆勝, 赤池一彦, 平田勝, 金子健二, 梶田秀司, 比留川博久, "ヒューマノイドロボット HRP-2 の開発", 日本ロボット学会誌, Vol. 22, No. 1, 2004.
- [3] 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.
- [4] Yoshiaki Sakagami, Ryujin Watanabe, Chiaki Aoyama, Shinichi Matsunaga, Nobuo Higaki, and Kikuo Fujimura. "The intelligent ASIMO: System Overview and Integration", In IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2478-2483, 2002.
- [5] N. Yamasaki: "Responsive Processor for Parallel/Distributed Real-Time Control", in IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1238-1244, Oct 2001.
- [6] H. Kobayashi and N. Yamasaki: "RT-Frontier: A Real-Time Operating System for Practical Imprecise Computation", in Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium, pp. 255-264, May 2004.
- [7] <http://www.itscj.ipsj.or.jp/ipsj-ts/02-06/toc.htm>.
- [8] 徳山陽人, 精廬幹人, 橋本周司, "USB インターフェース搭載型の小型モータコントローラの開発", 第 18 回日本ロボット学会学術講演会予稿集, pp.119-120, 2000.
- [9] Kaiser and Michael Mock, "Implementing the Real-Time Publisher/Subscriber Model on the CAN-Bus", in Proceedings of International Symposium on Object-oriented Real-time Distributed Computing, May 1999.
- [10] 小屋迫光太郎, 水内郁夫, 星野由紀子, 稲葉雅幸, 井上博允, "ロボットボディの組み込みプロセスモジュールの開発", 第 14 回ロボット学会学術講演会, pp.355-356, 1996.
- [11] K. Katoh, H. Kobayashi, N. Yamasaki and Y. Anzai: "Elastic Network Management for Flexible and Efficient Real-Time Communication", Proceedings of 18th International Conference on Computers and Their Applications, pp. 80-83, 2003.