

再構成可能な
モジュール型ヒューマノイドロボットの
システムアーキテクチャに関する研究

平成 18 年度

平 哲也

論文要旨

これからのヒューマノイドロボットには、オフィス、工場、医療などの変化に富んだ環境下で、多種多様な情報を実時間で処理・認識し、高い自律性を備えた動作を柔軟に計画・実行することが望まれている。このようなロボットが搭載する情報処理システムは、多種多様な外界・内界センサからの情報収集、多自由度の制御、状況に応じた多様な動作計画等、異なった性質を持つ多数のタスクを、実時間制約を満たしながら同時並列に処理しなければならない。さらに、オフィスや家庭での利用が想定されることより、ヒューマノイドロボットは、ユーザの様々な使用用途や環境下でも適応して動作できる能力を備える必要がある。しかし、従来のヒューマノイドロボットの制御システムでは、情報処理能力、拡張性において困難な壁に直面しており、これらの問題点を解決できるシステムアーキテクチャの提案が待たれている。

本研究では、これらの問題点を解決するために、モジュール化技術の概念とロボット動作の並列性に着目し、再構成可能なモジュール型ヒューマノイドロボットのシステムアーキテクチャを提案する。本システムアーキテクチャに基づいたヒューマノイドロボットと従来のヒューマノイドロボットとの違いは、腕ロボット、移動ロボット、頭ロボットなどの複数の機能別ロボットからシステムを構成する点である。本アーキテクチャに基づく各機能別ロボットは、専用の機械部、電子部、ソフトウェア部を備え、単体の自律ロボットとしても、ヒューマノイドロボットの一部としても動作可能である。これら機能別ロボットは、提案する制御ソフトウェアフレームワークに従って動作し、実時間通信規格である Responsive Link を利用したリアルタイムネットワーク上で他の機能別ロボットと接続することにより、従来のヒューマノイドロボットと比較して、処理の並列度が高い分散制御を実現する。

このような機能別ロボットの組み合わせでシステムを構成するモジュール型ヒューマノイドロボットは、ユーザに対して、自動車やパーソナルコンピュータのように使用用途や環境に応じてシステムの構成や機能を変更できる再構成機構を提供する。さらには、モジュール型ヒューマノイドロボットは、新たに開発した機能・アルゴリズム・デバイスを備えた機能別ロボットをシステムへ容易に組み込むことができ、従来のヒューマノイドロボットと比較して拡張性と柔軟性の高いシステムを実現する。

本論文では、研究背景について概説したあと、提案するアーキテクチャの設計とアーキテクチャに基づいた再構成可能なモジュール型ヒューマノイドロボットのプロトタイプの開発について述べ、システムアーキテクチャの有効性を処理性能、拡張性・柔軟性の観点から評価・検討を行う。その結果として、複数の機能別ロボットからヒューマノイドロボットを構成する本システムアーキテクチャは、従来のヒューマノイドロボットと比較して、処理性能、拡張性・柔軟性の面で有効であることを示した。

Abstract

This paper describes the design of the system architecture for a reconfigurable modular humanoid robot.

Humanoid robots can potentially assist human activities in human daily environments such as offices, homes, and hospitals. It is desirable for users to reconfigure their own robots to suit their needs and preferences as they can customize their own personal computers and cars. In order to develop a practical humanoid robot which performs skillful tasks in a real world together with human beings, expandable and flexible robot system architecture is required.

Our solution to these needs of users and researchers is a reconfigurable modular humanoid robot that is adaptable to new environments and various tasks. We focus on the parallelism of humanoid robot functions and the concept of modular systems. The system architecture for reconfigurable modular humanoid robots differs from conventional humanoid robot architecture by pursuing modularity of mechanical, electronic, and software parts based on functional robots such as wheeled robots, arm robots, and so on. The major advantage of our architecture is that users can reconfigure humanoid robots to different types of robots by combining several functional robots. The functional robots operate by themselves without being the parts of humanoid robots. For example, the functional robot operates as an arm part of the humanoid robot when integrated into the humanoid robot, while it operates as an arm robot when separated from the humanoid robot.

We evaluated the efficiency of our proposed architecture through the experiences using reconfigurable modular humanoid robot prototype R1. Experimental results show that the proposed architecture achieves expandable and flexible reconfiguration for users and researchers by changing the robot configuration to different types of robots for many purposes. We believe that our reconfigurable modular humanoid robot will enable user-specific humanoid robots more easily than ever before.

目次

第1章	序論	1
第2章	背景	5
2.1	ヒューマノイドロボットへの期待	5
2.2	ヒューマノイドロボット実現への技術的課題	9
2.2.1	ヒューマノイドロボットの情報処理システムの特徴	9
2.2.2	ヒューマノイドロボットに必要な情報処理能力	13
2.2.3	ロボット技術のシステム統合化	15
2.3	本章のまとめ	17
第3章	関連研究	19
3.1	情報処理システムの分類	19
3.2	集中制御システム	19
3.2.1	Robovie	20
3.2.2	H7	23
3.2.3	HRP-2	24
3.3	分散制御システム	26
3.3.1	ASIMO	27
3.3.2	HERMES	28
3.4	本章のまとめ	30
第4章	本研究の目的と動機	35
第5章	提案・設計方針	37
5.1	並列分散処理化	37
5.1.1	ヒューマノイドロボットの情報処理能力向上	37
5.1.2	行動計画部の並列・分散処理化	38
5.2	システム再構成機構	39
5.2.1	システム再構成機構とは	39

5.2.2	Plug-and-Play	40
5.2.3	モジュールロボット	41
5.2.4	モジュールロボットに関するまとめ	44
5.3	設計方針	45
5.4	前提条件	48
5.5	本章のまとめ	48
第6章	システムアーキテクチャの設計	51
6.1	制御アルゴリズムの適用検討	51
6.1.1	機能別ロボットの提案	51
6.1.2	脳ロボットの提案	52
6.1.3	再構成可能なモジュール型ヒューマノイドロボットの制御方式	53
6.1.4	階層化制御ソフトウェアプラットフォーム	55
6.1.5	モジュール間通信機構 (体内 LAN) の設計	58
6.1.6	レスポンスリンクを利用したモジュール間通信機構 RL-Net	75
6.2	プロトタイプロボット P1 を利用した予備実験	88
6.2.1	プロトタイプロボット P1 のシステム構成	88
6.2.2	プロトタイプロボット P1 を利用した予備実験	91
6.2.3	プロトタイプロボット P1 を利用した予備実験の評価・考察	94
6.3	システム再構成機構	99
6.3.1	モジュール接続ユニットの設計	99
6.3.2	再構成可能とする制御ソフトウェアの設計	101
6.4	本章のまとめ	105
第7章	プロトタイプロボットの実装と評価実験	107
7.1	再構成可能なモジュール型ヒューマノイドロボット R1	107
7.1.1	プロトタイプロボット R1 の全体概要	107
7.1.2	プロトタイプロボット R1 のシステム構成	108
7.1.3	プロトタイプロボット R1 の制御ソフトウェア	116
7.1.4	モジュール接続ユニット	117
7.2	プロトタイプロボット R1 を利用した実験による評価・考察	118
7.2.1	プロトタイプロボット R1 を利用した実験	118
7.2.2	行動計画部の並列・分散処理化に関する評価	125
7.2.3	拡張性・柔軟性に関する評価	127
7.3	本章のまとめ	132

目 次

2.1	我が国人口の自然増加数の推移	6
2.2	我が国の労働力人口の推移と高齢化	7
2.3	団塊世代の産業別就業人数 1	7
2.4	団塊世代の産業別就業人数 2	8
2.5	年齢別労働力人口の将来推移	8
2.6	ロボットに関連する要素技術およびシステム統合化技術の例	9
2.7	ヒューマノイドロボットの制御フロー	10
3.1	集中制御システムと分散制御システムの比較	20
3.2	集中制御システム	21
3.3	Robovie の電気・電子ハードウェア構成	22
3.4	Robovie のソフトウェア構成	23
3.5	H7 の電気・電子ハードウェア構成	24
3.6	H7 のソフトウェア構成	25
3.7	HRP の電気・電子ハードウェア構成	26
3.8	HRP のソフトウェア構成	27
3.9	分散制御システム	28
3.10	ASIMO の電気・電子ハードウェア構成	29
3.11	ASIMO のソフトウェア構成	30
3.12	HERMES の電気・電子ハードウェア構成	30
3.13	HERMES のソフトウェア構成	31
5.1	M-TRAN のシステム構成	42
5.2	OPEN-R のシステム構成	44
5.3	要素研究の研究体系	45
5.4	再構成可能なモジュール型ヒューマノイドロボット	46
5.5	再構成可能なモジュール型ヒューマノイドロボットの構成図	46
5.6	モジュール型ヒューマノイドロボットの再構成機構	49

6.1	機能別ロボット	52
6.2	機能別ロボットから構成するヒューマノイドロボット	53
6.3	Local Control	54
6.4	Global Control	55
6.5	5階層のソフトウェアアーキテクチャ	57
6.6	単体ロボットのソフトウェア構成	58
6.7	協調制御のためのソフトウェア構成	59
6.8	RL-Net の設計・評価環境	64
6.9	IEEE1394 の実験環境	65
6.10	レスポンスブリンクネットワークの実験結果 (イベントリンク)	67
6.11	レスポンスブリンクネットワークの実験結果 (データリンク)	67
6.12	IEEE1394 ネットワークの実験結果	68
6.13	レスポンスプロセッサ	72
6.14	Responsive Multithreaded Processor	73
6.15	リアルタイム OS:RT-Frontier	74
6.16	RM スケジューリングの例	74
6.17	モジュール間通信機構:RL-Net	75
6.18	イベントパケットとデータパケット	77
6.19	RL-Net のパケットヘッダ	78
6.20	RL-Net のパケットトレイラ	78
6.21	RL-Net が提供する通信用 API	81
6.22	ルーティングテーブル	81
6.23	ルーティングテーブル例	83
6.24	RL のホップ数と通信遅延に関する評価	85
6.25	RL-NetSim	87
6.26	プロトタイプロボット P1	89
6.27	プロトタイプロボット P1 の情報処理システム	90
6.28	プロトタイプロボット P1 の制御ソフトウェア構成図	91
6.29	プロトタイプロボット P1 の情報処理システムの構成図	91
6.30	アプリケーション 1:障害物回避行動	92
6.31	アプリケーション 2:自律移動	93
6.32	アプリケーション 3:人間とのインタラクション	94
6.33	アプリケーション 3 の情報処理システムの制御フロー	95
6.34	集中制御化したプロトタイプロボット P1 の情報処理システム	95

6.35	従来の制御手法での制御フロー	96
6.36	プロトタイプロボット P1 の情報処理システムの構成図 2	98
6.37	モジュール接続ユニットの設計概念図	101
6.38	モジュール型ヒューマノイドロボットの制御ソフトウェアの再構成	103
6.39	モジュール型ヒューマノイドロボットの制御ソフトウェアの切り替え	105
7.1	再構成可能なモジュール型ヒューマノイドロボットのプロトタイプ R1	108
7.2	再構成可能なモジュール型ヒューマノイドロボット	109
7.3	脳ロボット	109
7.4	車輪移動ロボット	110
7.5	機能別ロボット単体の行動例：車輪移動ロボット	111
7.6	歩行移動ロボット	112
7.7	機能別ロボット単体の行動例：歩行移動ロボット	113
7.8	右腕ロボット	113
7.9	機能別ロボット単体の行動例：右腕ロボット	114
7.10	頭ロボット	115
7.11	機能別ロボット単体の行動例：頭ロボット B	116
7.12	プロトタイプロボット R1 のソフトウェア構成	117
7.13	モジュール接続ユニット	118
7.14	モジュール接続ユニットを用いた機能別ロボットの再構成	119
7.15	モジュール型ヒューマノイドロボットの情報処理システム	120
7.16	握手動作	121
7.17	モジュール型ヒューマノイドロボットの情報処理システム	122
7.18	物の受け渡し行動	123
7.19	モジュール型ヒューマノイドロボットの情報処理システム	124
7.20	自律行動	125
7.21	プロトタイプロボット R1 が搭載する情報処理システムの制御フロー	126
7.22	集中制御化したプロトタイプロボット R1 の情報処理システム	126
7.23	従来の制御手法での制御フロー	127
7.24	再構成可能なモジュール型ヒューマノイドロボットの利点:頭ロボット	130
7.25	再構成可能なモジュール型ヒューマノイドロボットの利点:移動ロボット	131
7.26	情報処理システムの構成図	132

表 目 次

2.1	リアルタイム性	11
3.1	集中制御システムと分散制御システムの比較	32
5.1	Plug-and-Play に必要な条件	40
5.2	再構成可能なロボットの比較	42
5.3	再構成可能なモジュール型ヒューマノイドロボットの位置付け	47
5.4	再構成可能なモジュール型ヒューマノイドロボットの位置付け	49
6.1	通信インタフェースの比較	62
6.2	レスポンスリンクと IEEE1394 の比較	63
6.3	レスポンスリンクと IEEE1394 の比較	65
6.4	通信プロトコルのソフトウェアオーバヘッド	66
6.5	標準偏差による比較	68
6.6	peak-to-peak による比較	69
6.7	消費電力に関する評価	70
6.8	レスポンスリンクと IEEE1394 の比較結果	70
6.9	ルーティングテーブルの属性ビット	82
6.10	ルーティングアルゴリズム	82
6.11	通信パケットの優先度	85
6.12	プロトタイプロボット P1 の制御	97
6.13	プロトタイプロボット P1 の制御 (集中制御時)	97
6.14	ネットワークトポロジに関する評価	99
7.1	プロトタイプロボット R1 の制御	128
7.2	プロトタイプロボット R1 の制御 (集中制御時)	129
7.3	頭ロボットの比較	129
7.4	ネットワークトポロジに関する評価	133

第1章

序論

20世紀は自動車とパーソナルコンピュータの時代であったと言われている。1台の自動車やパーソナルコンピュータには、社会に存在するありとあらゆる要素技術が結集されているからである。これらの製品は、現代社会において人間の日常生活に深く浸透し、多くの人にとって欠かせない存在である。そして21世紀はロボットの時代であると言われている。工場などの生産現場で汎用機械として産業用ロボットが登場したのは1960年代のことだが、その後の技術革新によって、ロボットの高速化、高精度化は驚くほど高いレベルに達している。また、ロボットの作業内容も、簡単な部品の運搬から、溶接、塗装、組立などに至るまで、多岐に渡って実現されるようになってきている。日本は1980年をロボット元年と称し、その頃から産業用ロボットの生産および保有台数で世界一を誇っている。さらに近年では、犬、蛇などの機能や形態を模倣した動物型ロボットや、ヒューマノイドロボットと呼ばれる人間の機能や形態を模倣したロボットが続々と発表されている。このような背景から、ロボットへの期待は産業用にとどまらず医療、福祉、介護、教育、娯楽、災害救助など様々な分野へ広がっている。

昨今の新聞やテレビ、閉幕した2005年日本国際博覧会【愛・地球博】などを見ると、様々な最先端のロボットがこの期待に応えるために開発されていることがわかる。ロボットが、人間同様に歩くのはもちろんのこと、走ったり、倒れた状態から起き上がったり、また、人間と会話したり、握手したりと、部分的に見ればかなり高度なことが、いとも簡単にできているように見える。費用さえ考えなければ、今すぐにでも我が家でロボットが活用できるのではないかと、錯覚しそうになることもある。

しかし実際には、ロボットが家庭内で活躍する時代はまだまだ先のことでありと考えられる。ロボット技術は、電気・電子技術、センサ・アクチュエータ技術、システム・制御技術、計算機・情報技術、機械・製造技術など、電気・電子ハードウェアとソフトウェアの両面を集積して総合的に形成されるものである。そのロボット技術は、動作環境や作業内容の違いによって、技術的な難易度はかなり違い、産業用ロボットの技術と街や家庭などの生活環境において想定されるロボットの技術の間には大きな差がある。

1. 序論

生活環境での活躍が最も期待されているヒューマノイドロボットは、工場などで働くタスク指向の産業用ロボットと異なり、人間と協調して動作する必要がある。多くの大学や企業の研究者は、コミュニケーションロボット、アミューズメントロボットなどのロボット開発から、2足歩行、マニピュレーション、センシング技術、環境モデリング、動作計画、ロボット-ヒューマンインタフェース、AIなどのヒューマノイドロボットを構成する要素技術分野で、ヒューマノイドロボットの実用化に向けて努めている。そのようなヒューマノイドロボット実現の要となる情報処理システムは、多種多様な外界・内界センサからの情報収集、多自由度の制御、状況に応じた多様な動作計画等、異なった性質を持つ多数のタスクを、実時間制約を満たしながら同時並列に処理しなければならない。また、ヒューマノイドロボットは、様々な要素技術の実証実験をする際の基盤となる。そのため、ヒューマノイドロボットを構成する機械構造、電気・電子ハードウェア、ソフトウェアは、様々な研究者が研究・開発に参入できるように、また既存のアルゴリズム等を簡単に組み込めるように、拡張性や柔軟性を考慮する必要がある。さらに、医療、福祉、介護、教育、娯楽、災害救助など様々な利用が想定されることより、ヒューマノイドロボットは、ユーザの様々な使用用途や環境下でも適応して動作できる能力を備える必要がある。そのために、機械構造、電気・電子ハードウェア、ソフトウェア間のインタフェースを統一し、ロボット技術のモジュール化・部品化を進めることが解決の1つの方向性であると考えられる。ロボット技術のモジュール化・部品化は、部品毎の設計・実装、部品間の入れ替えを可能し、多数の研究機関や企業のロボット事業参入にも繋がる。

初期のヒューマノイドロボットは、リモートブレイン方式 [Inaba 97] と呼ばれる、体外に情報処理システムを持つシステムアーキテクチャが主流であったが、マイクロプロセッサの発達によって、1996年に完全自律型のヒューマノイドロボット P-2 [Hirai 97] が実現された。P-2の滑らかな歩行を実現したのは、胴体に搭載された4個の Micro SparcII プロセッサ (60MIPS) を VME バスを介して結合した情報処理システムであった [HONDA]。P-2の登場以来、様々なヒューマノイドロボットが大学・企業・国家プロジェクトから考案されてきた。代表的なヒューマノイドロボットとして、経済産業省が1998年から5ヵ年計画で実施した「人間協調・共存型ロボットシステムの研究開発」(HRP: Humanoid Robotics Project)の一環で川田工業株式会社、独立行政法人 産業技術総合研究所、株式会社安川電機、清水建設株式会社が開発した HRP-2 が挙げられる。HRP-2の情報処理システムは、2個の1.2GHzの Pentium-III プロセッサを100Mbpsの Ethernet で結合した構成である。HRP-2は、その情報処理システムの能力を駆使することで、整地歩行、転倒制御、転倒回復技術の実現、屋外共同作業への応用などの実証実験で成果を残した。しかし、ヒューマノイドロボットが生活環境で活躍するためには、さらなる運動技術や、環境認識技術、コミュニケーション技術など、様々な要素技術を統合化する必要がある。Pentium系のプロセッサ

1. 序論

の処理能力は，ムーアの法則に従うと，6年で20倍に向上することが期待される．2006年現在，3GHz級のPentium 4は1個で100W程度の電力を消費する．バッテリーで駆動するロボットの活動時間を長時間確保するためには，プロセッサにこれ以上の電力を割く余裕はない．また，プロセッサが発生する熱処理も深刻な問題であり，現在の技術では，大型のヒートシンクとファンの搭載が必須である．ファンの使用はノイズ源となり，音声認識にも影響を与えてしまう恐れがある．このように多くの問題を抱えている情報処理システムは，現在ヒューマノイドロボットの発展において重要な課題である．

一方，HRP-2を初めとする多くのヒューマノイドロボットは，A/D，D/A，カウンタなどセンサやアクチュエータの制御用I/Oを，胴体に配置するPCIカード上へ実装している．I/Oのチャンネル数は，関節数と同数以上必要である．その数が多いため，コネクタやケーブルの配置が問題となりやすい．制御用I/Oのケーブルは，胴体部にあるPCI上の制御用I/Oピンから，屈曲する狭い関節部分をメカや電気・電子ハードウェアと接しながら通り抜けて，手先・足先にあるセンサやアクチュエータまで配線される．このようなシステム構成は，拡張性に乏しく，優れた要素技術をシステムへ容易に組み込むことができない．また，このようなシステムでは，自動車やパーソナルコンピュータのように用途や環境に応じてロボットの機能や形態等をオリジナルにカスタマイズできる再構成機構をユーザへ提供することができない．そのために，従来のヒューマノイドロボットは，ユーザの様々な使用用途や環境下へ柔軟に適応することが困難であると考えられる．

以上述べたように，HRP-2を初めとする従来のヒューマノイドロボットは，(1) ヒューマノイドが搭載する情報処理システムの情報処理能力不足，(2) 物理的・エネルギー的な拡張性・柔軟性のあるシステム統合化技術の欠如という2つの問題点を抱えている．人間支援のためのヒューマノイドロボットを実現するためには，これらの問題点を解決できるシステムアーキテクチャの提案が待たれている．

そこで本研究では，これらの問題点を解決するために，モジュール化技術の概念とロボット動作の並列性に着目し，再構成可能なモジュール型ヒューマノイドロボットのシステムアーキテクチャを提案する．本システムアーキテクチャに基づいたヒューマノイドロボットと従来のヒューマノイドロボットとの違いは，腕ロボット，移動ロボット，頭ロボットなどの複数の機能別ロボットからシステムを構成する点である．本アーキテクチャに基づく各機能別ロボットは，専用の機械部，電子部，ソフトウェア部を備え，単体の自律ロボットとしても，ヒューマノイドロボットの一部としても動作可能である．これら機能別ロボットは，提案する制御ソフトウェアフレームワークに従って動作し，実時間通信規格であるResponsive Linkを利用したリアルタイムネットワーク上で他の機能別ロボットと接続することで，従来のヒューマノイドロボットと比較して，処理の並列度が高い分散制御を実現する．その結果，システム全体の処理能力を向上できる．

1. 序論

このような機能別ロボットの組み合わせでシステムを構成するモジュール型ヒューマノイドロボットは、ユーザに対して自動車やパーソナルコンピュータのように使用用途や環境に応じてシステムの構成や機能を変更できる再構成機構を提供する。さらに、モジュール型ヒューマノイドロボットは、新たに開発した機能・アルゴリズム・デバイスを備えた機能別ロボットをシステムへ容易に組み込むことができ、従来のヒューマノイドロボットと比較して拡張性と柔軟性の高いシステムを実現する。プロセッサ-制御用I/O間の距離を短く抑えることもでき、システム全体の信頼性、メンテナンス性も向上できる。

以上、本論文では、従来のヒューマノイドロボットが抱えている問題点(1)、(2)を解決できる再構成可能なモジュール型ヒューマノイドロボットのシステムアーキテクチャの設計とその有効性について述べる。

本論文の構成は以下の通りである。2章では、研究の背景について概説する。3章では、関連研究について述べ、従来研究の問題点について述べる。4章では、本研究を行うに至った動機と目的について述べる。5章では、従来の問題点を解決できるシステムアーキテクチャの設計方針を述べる。6章では、再構成可能なモジュール型ヒューマノイドロボットのシステムアーキテクチャの設計について説明し、本研究の研究範囲を明確にする。7章では、システムアーキテクチャに基づいたプロトタイプロボットの開発とプロトタイプロボットを利用した本システムアーキテクチャの有効性検証実験を示す。そして8章で本研究をまとめる。

第2章

背景

本章では、ヒューマノイドロボットへの期待とヒューマノイドロボット実現への技術的要件について明らかにする。

2.1 ヒューマノイドロボットへの期待

ロボットは今、新しい時代の幕開けを迎えている。我が国は、技術面でも市場規模の面においても、いわば「ロボット大国」として位置付けられている。ただし、90年代以降、産業用ロボットの市場規模は緩やかな成長にとどまり、用途が特定の産業に限定されている。また、サービスロボットについても、実用例は少数にとどまっている。

しかし、近年、ロボットを巡る状況は着実に変わりつつある。製造業においては、ロボット・セルのようなさらに高度化・知能化した産業用ロボットが、生産現場に投入されつつある。社団法人日本ロボット工業会の統計によると、2004年の産業用ロボット出荷額は、5,768億円であり、うち国内出荷は2,724億円(総出荷額のうち47%)、輸出は3,045億円(同53%)となっている。さらに、産業用ロボットの稼働台数も世界で最も多い。産業用ロボットの主要ユーザ産業は、自動車産業及び電子電機産業である。このうち、自動車産業向けでは、スポット溶接、アーク溶接、塗装が出荷額の大半を占めている。また、電子電機産業向けでは、電子部品実装(プリント基板実装)、半導体実装、クリーンルーム及び組立等が出荷額の大半を占めている。特に、近年ではテレビのフラット化が急速に進展する中、液晶テレビ用のガラス基板搬送等に使われるクリーンルーム用ロボットも、市場として急拡大している。

産業用ロボット以外の市場規模は約70億円程度と見込まれ、市場としては「離陸段階」に差し掛かりつつある状況と言える。2005年日本国際博覧会【愛・地球博】では、ヒューマノイドロボットやサービスロボットの実用化に向けた実証実験が行われ、国内外から大きな注目を集めた。実際のビジネスにおいても、掃除ロボットの導入等、実用例も着実に増えている。このように我が国のロボット産業は、従来のステージから一段高いステージ

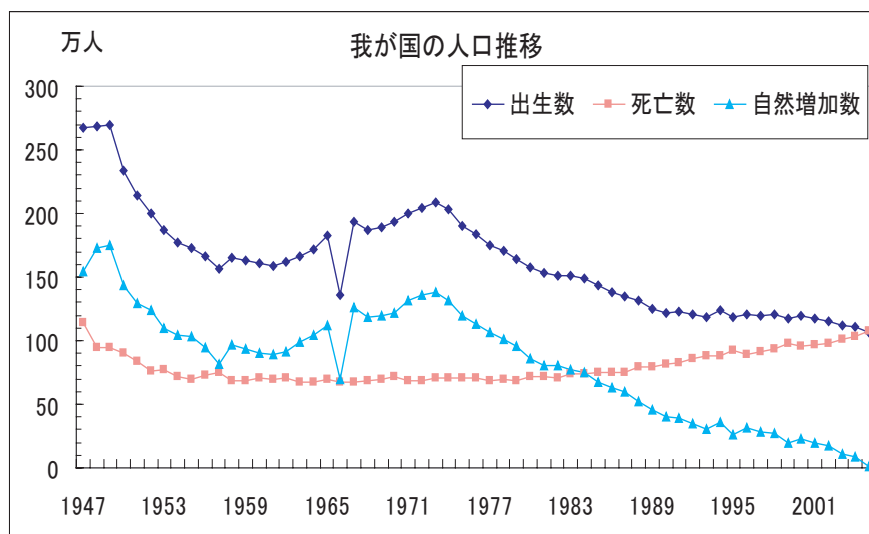


図 2.1: 我が国人口の自然増加数の推移

に移行し、「第2のロボット普及元年」の幕開けを迎えている。

一方で、現在の我が国は、少子高齢化・労働力人口の減少・国際競争の激化といった課題を抱えている。1995年から2000年にかけては生産年齢人口が、1999年には労働力人口が、それぞれ減少に転じた。また、人口動態統計(年間推計)によると、ヒューマノイドロボットが注目を浴びた2005年には、我が国総人口が約1万人の自然減となった(図2.1)。このように、我が国は人口減少時代に入つつある。

また、労働力の年代別構成(図2.2)を見ると、労働力人口の中で突出したボリュームゾーンを形成する団塊の世代(1947~1949年生まれ)は、学卒期が製造業・建設業の成長期にあたったため、男性の4割以上がこの2業種のいずれかに就職している(図2.3, 図2.4)。

具体的には、製造業で25.6%(同世代が、現在の製造業就業者の13.7%を占めている)、建設業で15.0%(同14.5%)となっている。このため、今後65歳までの雇用延長を適用したとしても、2011年以降、団塊の世代の労働市場からの一斉退出は、製造業等における労働力不足のみならず、技能の断絶による生産性や製品品質の低下が懸念されている(図2.5)。これがいわゆる2007年問題である。

図2.3からも明らかなように、女性の6割以上は小売業・サービス業に就職している。具体的には、団塊の世代は、小売業(卸・飲食店を含む)に28.9%、サービス業に33.3%が就業している。これらの割合は、現在の両業種のそれぞれ14.2%、11.8%を占めている。従って、これらの分野についても、団塊の世代の労働市場からの退出による労働力不足が懸念される。

今後、これらの問題は、労働生産性の向上、女性の社会進出等、総合的な政治対策の中

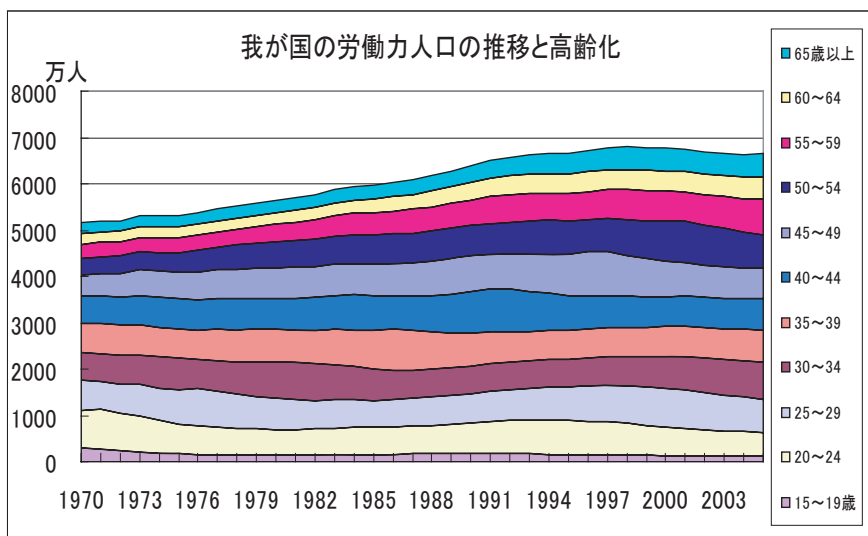


図 2.2: 我が国の労働力人口の推移と高齢化

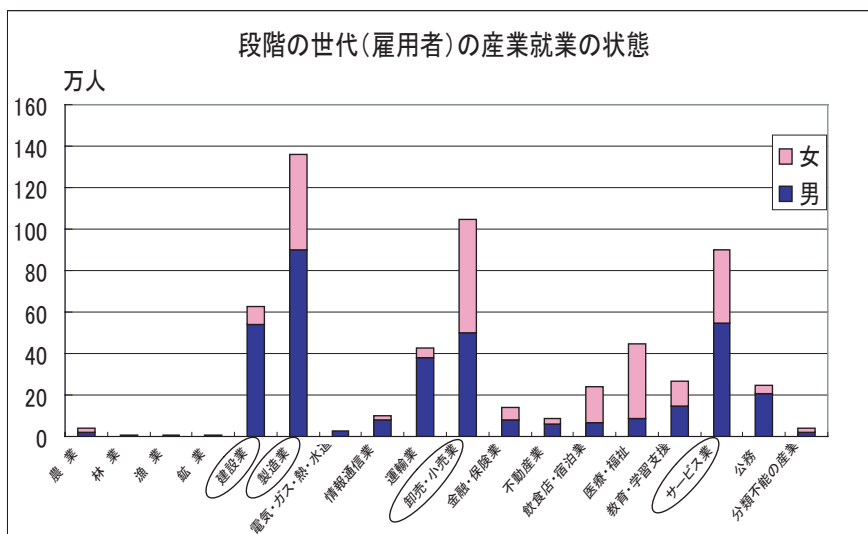


図 2.3: 団塊世代の産業別就業人数 1

で解決されるべきであるが，図 2.6 に示す我が国が得意とするロボット技術は，下記の理由でその解決の糸口を与え，日本のさらなる成長の一翼を担っていくと考えられる．

1. ヒューマノイドロボットやサービスロボットを含む新たなロボット産業が創成される上で，ロボット産業国である我が国は，競争力を発揮しやすい
2. 我が国製造業及びサービス業が，今後，少子高齢化・労働力の減少が進む中で，国際的な競争力に打ち勝つためには，製造業の生産工程又はサービス業の作業工程にお

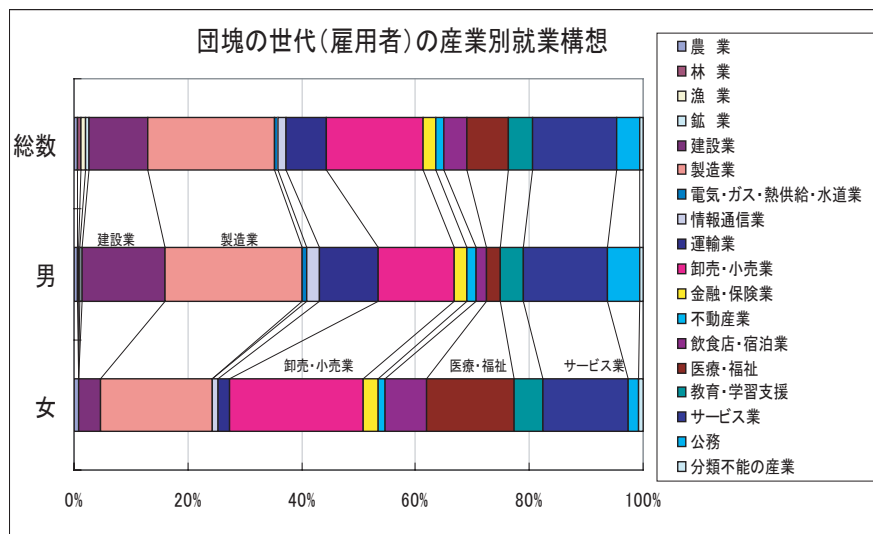


図 2.4: 団塊世代の産業別就業人数 2

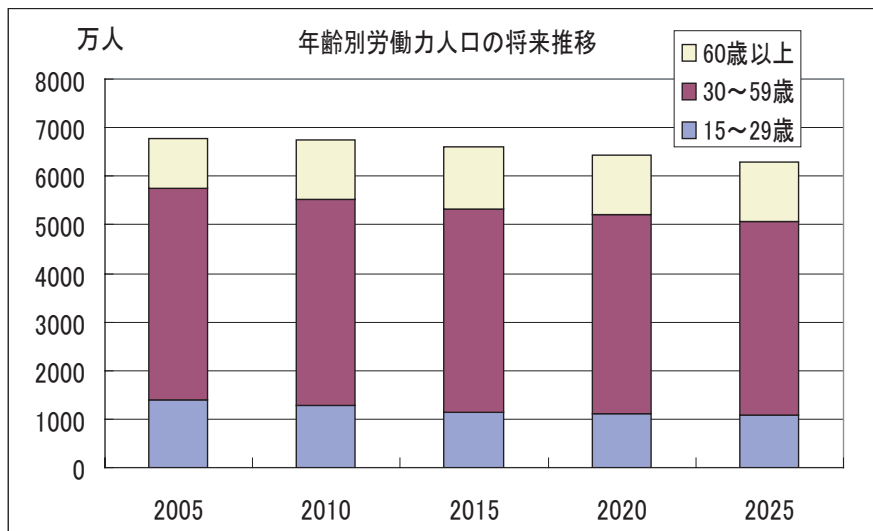


図 2.5: 年齢別労働力人口の将来推移

いて、一層のロボット化・自動化が不可欠である

ロボット技術の中でも、人間と同じ身体機能を持つヒューマノイドロボットは、社会と共存・協調しながら、「アシスタント」「製造」「福祉」などの領域で人間社会に新たな価値をもたらすロボットとして特に期待されている。

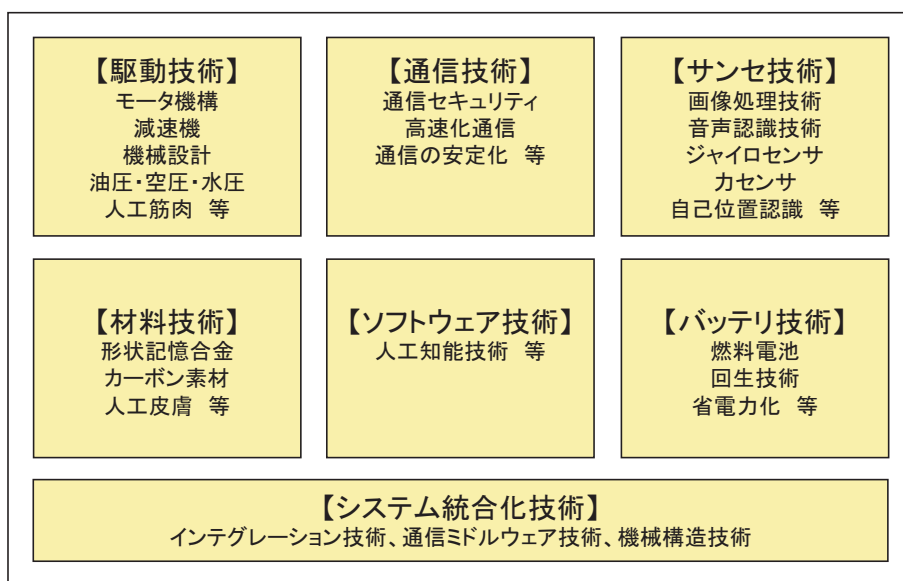


図 2.6: ロボットに関連する要素技術およびシステム統合化技術の例

2.2 ヒューマノイドロボット実現への技術的課題

2.2.1 ヒューマノイドロボットの情報処理システムの特徴

「産業用ロボット」と「ヒューマノイドロボット」の両者には大きな違いが存在する。現在の産業用ロボットの特徴は、床に固定されていること、労働安全の観点から柵で囲われ作動中は人間から隔離されていること、反復作業を行うことが多いことである。ロボットの作業内容は、基本的に現場の特定技術者が当日の作業内容に応じてティーチング・ボックスなどを使って1つ1つきめ細かく教示する。もちろん、次世代産業用ロボットの場合には、より人間に距離が近い環境で、複雑な作業を行うことが予想される。

一方、ヒューマノイドロボットは人間の生活環境の中に入ることを前提としており、移動性が強く、人との物理的距離が近く、加えて反復作業が少ないという特徴がある。家庭などにおける生活環境は大変複雑で、かつ、時々刻々と変動している。こうした環境変動を把握するために、ヒューマノイドロボットは視覚、聴覚、触覚などの機能を備えて、環境に関する大量の情報を実時間で処理する必要がある。また、ヒューマノイドロボットは、環境の変動に対応できるように、また多様な作業を実行できるように、より多くの運動自由度を備える必要がある。ヒューマノイドロボットの作業は、一般ユーザによる手軽な音声指令などで環境条件に応じて自律的に動作を決定しなければならない。以上のことより、ヒューマノイドロボットは産業用ロボットと比較して、高度な情報処理能力が特に必要になることが明らかである。

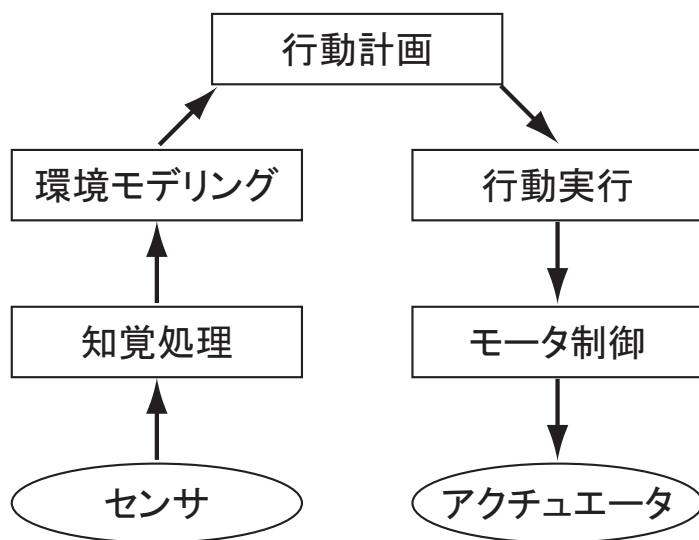


図 2.7: ヒューマノイドロボットの制御フロー

ヒューマノイドロボットが、複雑な作業環境下で、高い自律性を持って多様性に富んだ動作を柔軟に実現するためには、ヒューマノイドロボットが搭載する情報処理システムは、多種多様な情報の収集と処理、多くの関節の制御、環境に応じた多様な動作の計画と制御などの膨大な演算処理を (1) リアルタイム性を保証しつつ、(2) 同時並列的に行わなければならない。ヒューマノイドロボットが搭載する情報処理システムは、図 2.7 に示す「ロボットの状態を数十から数百個の内界・外界センサで認識し (知覚処理)、その認識結果を内部表現で表し (環境モデリング)、その表現を基に行動計画を立て (行動計画)、そして実際に数十個のアクチュエータを通して行動を起こす (行動実行)」という一連の制御フローを処理する必要がある。この制御フローは、数十 μsec 周期での処理を必要とする電流制御、PMW 制御などのタスクから、数秒周期での処理を必要とする環境認識に基づいた行動計画などのタスクまで幅広い時間制約を持つタスクから形成される。従って、この制御フローを処理する情報処理システムの性能がロボットの性能を左右することになる。

ここで述べたリアルタイム性とは、処理の真偽が結果の真偽だけでなく、時間にも依存する性質のことを言う。狭義には与えられた時間制約 (デッドライン) を守ることを意味する。リアルタイム性は、時間制約の種類により大きく 2 つに分けられる。

- ハードリアルタイム性
- ソフトリアルタイム性

ハードリアルタイム性とは、必ず時間制約を守る必要がある性質である、時間制約が守られなかった場合、処理結果の価値が直ちに 0 になる。ハードリアルタイム性を持つ処理

表 2.1: リアルタイム性

	ハードリアルタイム性	ソフトリアルタイム性
デッドラインミス時の影響	厳しい	比較的厳しくない
実行周期	短い	長い
演算量	少ない	多い

は時間制約が厳しいため、実行時間の見積もりの正確さ(実行時間の予測性)が重要になる。ハードリアルタイム性を持つ処理は、実行周期が $100\mu\text{s}$ から 10ms と短い、演算量が少ないといった特徴がある。

ソフトリアルタイム性とは、時間制約を破ることを許容する性質である。時間制約が守られなかった場合でも結果の値が直ちに0にはならず、時間経過とともに処理結果の値が減少していく。ソフトリアルタイム性を持つ処理は時間周期が 10msec から 1sec と比較的大きい、演算量が多いといった特徴がある。音声合成や画像処理のように、マルチメディア処理などがソフトリアルタイム性を持つ。ソフトリアルタイム性を持つ処理はデッドラインミスに関してはハードリアルタイム性を持つ処理よりも寛容だが、大量のデータを演算する。

以上、ハードリアルタイム性とソフトリアルタイム性について特徴をまとめると表 2.1 のようになる。ロボットが搭載する情報処理システムは、このように性質の異なるリアルタイム処理を実行する必要がある。具体例を挙げてみる。ヒューマノイドロボットが人間と対話をしながら前進するような場合、アクチュエータを動作させる処理、人間とコミュニケーションを行うための処理などが実行される。アクチュエータを動作させる処理では、一定時間毎にセンサなどからデータを取得し、アクチュエータの操作量を演算し、アクチュエータを制御する。センサが障害物を検知した場合、アクチュエータの操作が遅れるとロボットは障害物に激突してしまう。そのため、この種の時間制約が守られなかった場合、システムに与える影響は大きい。よって、このような処理はハードリアルタイム処理である。一方、コミュニケーションを行うための処理では、人間と会話を行うために人間を画像処理を用いて認識し、マイクからの入力を音声認識し、人間からの問いかけに対する返事を生成し、音声合成を行ってスピーカから返事を出力する。会話の応答が遅れると、人間とコミュニケーションをするのは難しくなる。しかし、時間制約が守られない場合でも、システムに与える影響は少ない。よって、このような処理はソフトリアルタイム処理である。これらの処理はシステム内で同時に実行され、どちらの処理も時間内に完了する必要がある。この例からもわかるように、ヒューマノイドロボットには、産業用ロボットと比較しても高度な処理能力が必要なことがわかる。

1章でも述べたように、初期のヒューマノイドロボットはリモートブレイン方式と呼ばれる、体外に情報処理システムを持つシステムアーキテクチャが主流であった。この方式は、大きくて重い当時の計算機をロボットへ搭載することが困難であった問題点を回避するために開発された手法である。プロセッサの発達によって、1996年に完全自律型のヒューマノイドロボット P-2 が実現された。P-2 の情報処理システムは、4 個の Micro SparcII プロセッサ (60MIPS) を VME バスで接続した構成であった。しかし、その情報処理システムの処理性能は、歩行制御程度しか実現できなかった。現在では、複数の高性能なプロセッサ、組み込みマイコン、DSP、FPGA などを搭載することが主流になっており、その情報処理システムは、歩行制御に加え、画像処理、音声処理も可能となった。しかし、現在のヒューマノイドロボットの性能でも、医療、福祉、介護、教育、娯楽、災害救助などの様々な分野で活躍できず、より高い情報処理能力が要求される。

Pentium 系のプロセッサの処理能力は、ムーアの法則に従うと、6年で20倍に向上することが期待される。しかし、2006年現在、プロセッサの消費電力は製造プロセスの微細化によるリーク電流の増加や、処理能力の向上の主手段である動作クロックの向上を主因にして現実的でないほど増加していくことが見込まれており、現状の設計思想のままでは、プロセッサの消費電力と冷却が非現実的なものとなり単独でのプロセッサの処理能力向上が難しくなっている。そこで、様々な半導体チップベンダが、プロセッサダイの上にそれぞれ単独に機能するプロセッサを実装し、並列動作させることによって、プロセッサチップ全体での処理能力を上げ、性能向上を果たすことを目的として開発されているマルチコアプロセッサ、メニーコアプロセッサの開発が進んでいる。マルチコアはコア毎に電圧やクロックの制御を行う機能を付加したり、より低いクロックでも処理性能が稼げることから、この問題への解決策として提案されている。また、大手半導体チップベンダである Intel 社は、将来的には同種の演算処理を行う複数の大小のコアを実装し、並列処理可能な命令群は多くの小さく消費電力の低いコアを用い、並列処理が苦手な処理は消費電力が大きいものの少数の大きなコアを用いるという手法の研究が進んでいる。しかし、その Intel 社のロードマップを見る限りでは、バッテリーで駆動するロボットにとっては、プロセッサの消費電力は依然大きいままである。従って、プロセッサの処理能力が向上したとしても、ロボットへ搭載できるとは限らないと言える。

以上のことから、プロセッサの処理性能向上を期待するだけでなく、ロボット専用プロセッサの開発や情報処理システムの構成自体を見直す必要があると言える。

2.2.2 ヒューマノイドロボットに必要な情報処理能力

Hans Moravec 氏は、人間の情報処理能力は1億 MIPS と試算した [Moravec 00]。ヒューマノイドロボットに必要な情報処理能力は、ロボットが行う身体的・知的作業の質と速さに依存するので、ロボットが行うタスクを定めれば、必要な能力を算出可能そうに思える。しかし、例えば、障害物を回避して移動する動作、飛んでくるボールを見て捕球する動作に必要な MIPS 値を正確に見積もることは、その戦略アルゴリズムが確立されていないため不可能である。だがしかし、どれくらいの情報処理能力を備えていればどれくらいのタスクが実行できるのかを把握することは、人間の生活環境下で人間と共存するヒューマノイドロボットを今後開発していくためには重要なことである。従来のヒューマノイドロボットが備えている情報処理システムと実現している動作を基準におき、何倍の能力があればどの程度のタスクが実現できそうかをおおよそ見積もることは可能である。本節では、図 2.7 に示すロボットの情報処理システムの環境認識 (知覚処理と環境モデリング)、行動計画、行動実行の3つの項目に分けて必要な情報処理能力を考察する。

環境認識処理

ヒューマノイドロボットが、変化する人間の生活環境下で自律的に動作を計画したり、安全に動作を行うには環境の認識が必要である。ヒューマノイドロボットの情報処理システムは、内界センサを利用したロボット自身の状態モデリング、距離センサを利用した環境認識、触覚センサを利用した物体把握、視覚センサを利用した顔認識、動体追跡、人との音声対話など様々な環境認識技術を行う必要があるが、その中でも最も大きな情報処理能力を必要とする処理は視覚処理である。

従来、画像理解のためにはエッジ、テクスチャ、面の法線など様々な特徴に基づく手法が提案されたが、3次元世界で動作するヒューマノイドロボットには、2眼立体視(ステレオ視)による距離画像を用いる方法が最も有力であるとされている。その理由は、レーザレンジファインダ、超音波センサなどの距離センサとは異なり、視覚センサを用いると距離情報、色情報を同時に取得できるためである。2眼の場合の距離画像生成の処理は、右画像の小領域と相関の高い画像小領域を左画像のエピポーラ制約にそった領域で探索するものである。320 × 240 の画像から 11 × 11 の小領域を対象として全画面の距離画像を生成する速度は、1.4GHz の PentiumIII プロセッサを利用して約 20msec 程度である。この処理能力は、50fps で距離画像生成が可能である。一般的な 35mm(35mm カメラ換算)の準広角レンズを用いると、320 × 240 の画像は人間における視力 0.1 に相当し、解像度が十分でないことは明らかである。例えば、4倍の 1280 × 960 にすれば視力 0.4 に相当するが、視力 0.1 の時と比較して処理能力は 16 倍必要となる。さらに、35mm の視野角も十分ではない。

聴覚処理は視覚処理に次いで大きな処理能力を必要とする。聴覚処理は、マイクロフォンアレイによる音源方向の同定、音声の周波数分析、HMMに基づく音声認識と音声合成である。高品質な音声合成で1秒間の音声を生成する速度は、1GHzのPentiumIIIを利用して約0.1-1秒程度である。

以上のことから、ヒューマノイドロボットの環境認識には莫大な情報処理能力が必要であることが明らかである。

行動計画

ヒューマノイドロボットが安定して動作するためには、目的地に近づくための経路計画と着地点の計画、自分の体にぶつからない手足のコリジョンフリーなリーチング戦略、足を蹴り上げるタイミングとトルク配分などを短時間で計算し、運動中は1-5msecの周期で全身の合計30個程度のアクチュエータをサーボする必要がある。歩行を含む移動技術に最低限必要なセンサは、各軸の角度を取得するエンコーダ、足の接地面を検出する触覚センサ、ロボットの傾き角を検出するジャイロなどである。さらに障害物と衝突せずロボットが動作するためには、先に述べた視覚処理、聴覚処理などを通じて環境モデルを獲得し、多次元空間の中で、他の物体と干渉しない姿勢の連鎖を探索するプランニングが必要である。探索空間は、ロボットを1つの剛体と見なすと6次元、関節数で数えると30次元となる。角度を1度ずつに分割し、x-y-z各軸を512に分割すると、6次元の場合での探索空間の大きさは、 6×10^{15} に達する。

ロボットの動作を考えるとこの探索空間の完全探索は全く不可能であり、遺伝的アルゴリズム、ランダムイズドプランナ、パターン分類など、ヒューリスティックな戦略アルゴリズムの導入が不可避である。ヒューマノイドロボットの姿勢と運動は、合計30個程度のアクチュエータに与える角度と速度で決まるが、接地足や把持腕を通じて環境から与えられる制約だけでは、全部の運動パラメータを決定することができない。すなわち、ヒューマノイドロボットは冗長な自由度を持つので、最も適当な動きになるように、ある規範に基づいて動作(関節角と角速度の時系列)を決めてやらなければならない。現在のヒューマノイドロボットが完全に平らな床でない歩行できないのは、1秒分の動作をあらかじめ作っているのに、接地時に受ける予測できない力に応じた制御をとることができないからである。動作中に受ける外乱を考慮したよりロバストな移動を実現するには、この10倍の能力によって常に短期計画を作成し続けることが必要である。

より長期の歩行計画は、全く障害物がない環境であれば瞬時に接地点が計算できるが、障害物が存在する環境ではそうはいかない。[Chestnut *et al.* 03]では、3m先の目的地に達するのに5mの障害物回避軌道になる場合、5m × 5mの平面を256 × 256に分割し、障害物を避けつつ目的地に到達するための足の接地点と歩行パターンを計算しており、1mあた

り5万の探索枝の処理を必要としている。その処理速度は、1.4GHzのPentiumIIIを利用して約1秒であるとしている。このように2次元での探索処理は、ほぼ経路長の2乗に比例する時間がかかり、例えば隣の部屋に行くようなより長い経路の探索では、より大きな情報処理能力が必要となる。

行動実行

[松井 他 04]によるとHRP-2が搭載する1.4GHzのPentiumIIIの情報処理システムは、移動パターン生成に20%、I/Oを含むサーボに20%、センサ情報を用いたパターン修正による歩行の安定化に20%を費やしている。歩行軌道は、5msec毎に関節目標値が生成され、この目標値を達成するべく、各関節を1msec毎に位置サーボを実行している。一般に、サーボ周期が短い方が制御系は安定し、サーボゲインを上げることができる。しかし、トルクに比して制御対象が大きな慣性を持つ場合は、周期を短くすることには大きな意味はない。従来のヒューマノイドロボットは、基本的には全身のアクチュエータに対して位置制御を行っている。ヒューマノイドロボットの減速比は50~150と大きいので、通常の力制御を適用することは困難である。しかし一般的に力をセンスすることと非常に短い周期での位置制御が可能であれば、力制御と同じ効果が得られることが知られている。現在の5倍程度に速度を上げることで、不規則な形状の表面に沿って一定の力を出力できるなら動作等も実現できると予測される。

2.2.3 ロボット技術のシステム統合化

前述の通り、ロボット技術自体は、駆動関連技術から材料技術、通信技術、ソフトウェア技術に至るまで、多様な要素技術領域を包含する技術領域である。このため、ロボット技術活用領域もロボットに限らず、自動車や情報家電といった分野と重なり合っている。近年は、ロボット技術と自動車技術や情報家電・情報通信技術との融合が更に進みながら、ロボット技術が急速に発展しつつある。複数の自動車メーカーがロボット開発に参入しているのもその例である。具体的には、車両の運動制御技術や自動車製造に使っていた技術をロボットに応用し、サービスロボットや人と協働する産業用ロボットの開発を目指す例や、車両直線走行技術や障害物回避技術、清掃技術を組合わせて業務用清掃ロボットの開発・実用を行っている例が挙げられる。また、情報家電分野では、家電製品等の技術を家庭用掃除ロボットに応用している例、音声・画像認識分野を中心にソフトウェアやデバイスを開発し、それを活用してコミュニケーションロボットを製造している例が挙げられる。このように、現在、自動車や情報家電分野等の技術とロボット技術が融合し、活用される領域も広がり、互いにオーバーラップしつつある。今後、サービス分野でのロボット技術活

用が広がれば、これら我が国が競争力を有する技術が、一気に開花することも予想される。実際、既に、自動車の ITS の進展、電子化 (先進前方照明システム (AFS)、電子制御式インジェクタ、横滑り防止装置 (ESC)、カーナビゲーションの高機能化等) を見ると、ロボット技術と他分野との融合が、急速に進んでいると言える。

このことから、今後ロボット技術の発展のために重要な要件の 1 つは、要素技術を磨くだけでなく、様々な分野で開発されている要素技術のうち優れた技術を自由に組み合わせることで新たなロボットサービスやロボットシステムの構築を可能とするシステム統合化技術を提唱することである。

20 世紀に最も発展し普及したと言われる自動車やパーソナルコンピュータは、決して万能な製品ではなく、あくまで一般の用途に、一般の利用環境下で、利用できる汎用製品である。そのような汎用品が普及した理由は、そのユーザが必要に応じてその機能や形態を自由に変更可能とすることが可能であった点も挙げられる。実際に、使用済みになるまで購入したままの自動車やパーソナルコンピュータを使っているユーザはほとんどおらず、少なからず、その機能や形態をオリジナルにカスタマイズしている。自動車やパーソナルコンピュータがユーザに対してその機能や形態を自由にカスタマイズできる再構成機構を提供できるのは、それらの製品を構成する技術がモジュール化・部品化されているからである。モジュール化・部品化は、部品毎の設計・実装、部品間の入れ替えを可能にする。その結果、それらモジュールや部品をユーザが交換することで、その機能や形態をオリジナルにカスタマイズすることができる。またモジュール化・部品化は、多数の研究機関・企業の開発者も開発に参加可能であり、その製品の発展に大きく貢献する。

ヒューマノイドロボットの場合も、自動車やパーソナルコンピュータと同様にユーザー一人のものとして、個人に所属することが想定される。さらに、医療、福祉、介護、教育、娯楽、災害救助など様々な利用が想定されることより、ヒューマノイドロボットは、ユーザの様々な要求や利用環境下でも適応して動作できる能力を備える必要がある。しかし、ユーザの様々な要求や利用環境下にも対応できる万能なヒューマノイドロボットを開発することは、現時点では難しいと言える。従って、自動車やパーソナルコンピュータと同様に、ヒューマノイドロボットも、それぞれのユーザの嗜好に応じて、ロボットの機能や形態をオリジナルにカスタマイズできる再構成機構を備えておく必要があると考えられる。

以上より、システム統合化技術を提唱する際には、モジュール化や部品化にも留意する必要があると考えられる。

2.3 本章のまとめ

本章では、ヒューマノイドロボットへの期待について述べた。さらに、その期待に応じることができるヒューマノイドロボットを実現するために必要となる技術について述べた。

ヒューマノイドロボットは、今後の製造業、サービス業の国際的な競争力や、少子高齢化・労働力の減少に対しての解決策として期待されている。しかし、従来のヒューマノイドロボットはその期待に応えられるレベルに達しているとは言い難い。

その原因の1つは、従来のヒューマノイドロボットが搭載する情報処理システムが、期待に応じた動作をするための処理能力を満たしていない点である。2つ目の原因は、従来のヒューマノイドロボットが、現在までに開発されてきた優れた電気・電子技術、センサ・アクチュエータ技術、システム・制御技術、計算機・情報技術、機械・製造技術などの電気・電子ハードウェアとソフトウェアを搭載でき、異分野の技術領域の融合による新たな技術を創出できるシステム統合化技術が提唱されていない点である。

以上より、ヒューマノイドロボットを実現するためには、高度な情報処理能力を備える情報処理システムと、要素技術をヒューマノイドロボットへ統合するためのシステム統合化技術の2つを実現できるシステムアーキテクチャの提案が必須であることがわかる。

次章では、従来の研究が提案する高度な情報処理能力を実現している情報処理システムの構成手法と要素技術をヒューマノイドロボットへ統合するためのシステム統合化技術について考察する。

第3章

関連研究

本章では，2章で述べたヒューマノイドロボット実現に向けて要求されるヒューマノイドロボットが搭載する情報処理システムの高度な情報処理能力，要素技術のシステム統合化技術の2点に着目して，従来のヒューマノイドロボットのシステム構成の比較・検討を行う．その検討より，ヒューマノイドロボット実現に向けた要求に対して，従来のヒューマノイドロボットのシステム構成が解決できていない点を明らかにし，情報処理システムの高度な情報処理能力，要素技術のシステム統合化技術の要求を解決できるシステムアーキテクチャの検討を行う．

3.1 情報処理システムの分類

現在のヒューマノイドロボットが搭載している情報処理システムの構成手段は，大きく下記の2種類に分類できる．

- 集中制御システム
- 分散制御システム

図3.1の左半分は集中制御システム，右半分は分散制御システムのシステム構成をそれぞれ示している．以降では，集中制御システムと分散制御システムの詳細を述べ，それぞれのシステムの特徴を明らかにする．

3.2 集中制御システム

集中制御システムとは，図3.2に示すように，システム全体のI/Oの管理や図2.7で示した制御フローを1つの高性能なプロセッサで処理するシステムのことである．

代表的なロボットとして，Robovie(ATR 知能映像通信研究所)[神田 他 02]，H7(東京大学)[Kagami *et al.* 02]，HRP-2(産業総合研究所)[Kaneko *et al.* 02]，WAKAMARU(三菱重

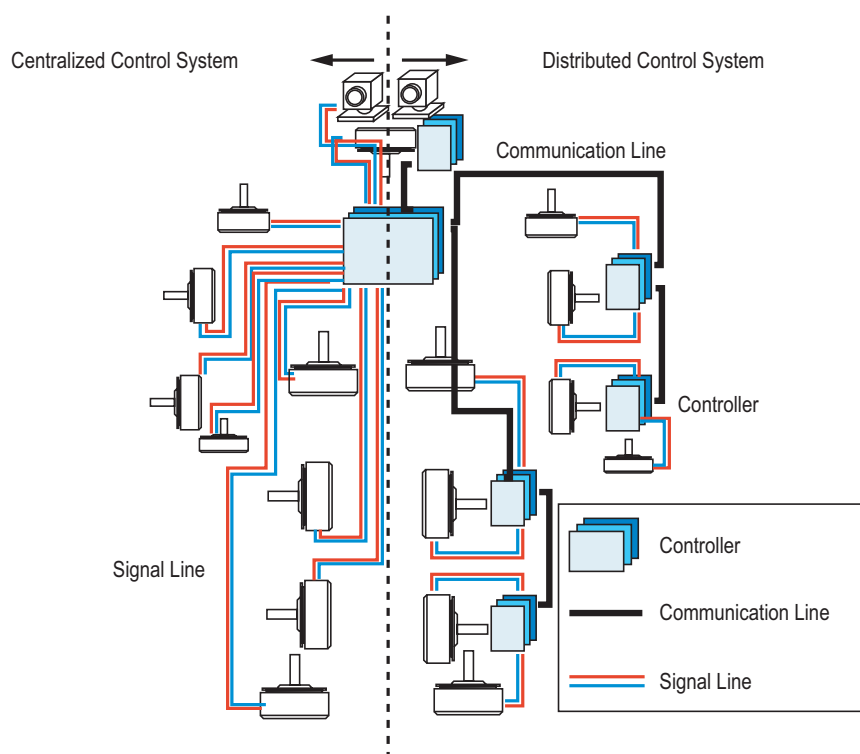


図 3.1: 集中制御システムと分散制御システムの比較

工)[川内 他 03], ROBISUKE(早稲田大学)[於久 他 02]などが挙げられる。以下では, 集中制御システムの代表例として Robovie, H7, HRP-2 を取り上げ, 集中制御システムの情報処理能力, 要素技術のシステム統合化技術の 2 点に関する利点・欠点を明らかにする。

3.2.1 Robovie

Robovie[神田 他 02] は, ATR 知能映像通信研究所で開発された 13 自由度の上半身型ヒューマノイドロボットであり, 人間とコミュニケーションするための視覚・触覚・聴覚センサを備えている。

Robovie の情報処理システムは, 図 3.3 で示すように, 1 つの 850MHz の PentiumIII プロセッサが, 胴体部に設置する PC マザーボードを介して, HDD, 無線 LAN カード, 画像キャプチャーカード, サウンドカード, 制御用 I/O ボードと接続する構成である。このような情報処理システムを備える集中制御システムは, 比較的シンプルな機械構造, 電気・電子ハードウェア, ソフトウェアでロボットシステムを設計できる利点がある。また, 集中制御システムでは, プロセッサを 1 つしか備えていないので, プロセッサ間通信や通信ミドルウェアを考慮する必要がない利点もある。

Robovie には, コミュニケーション能力を拡大するための二輪倒立振子機構を導入した

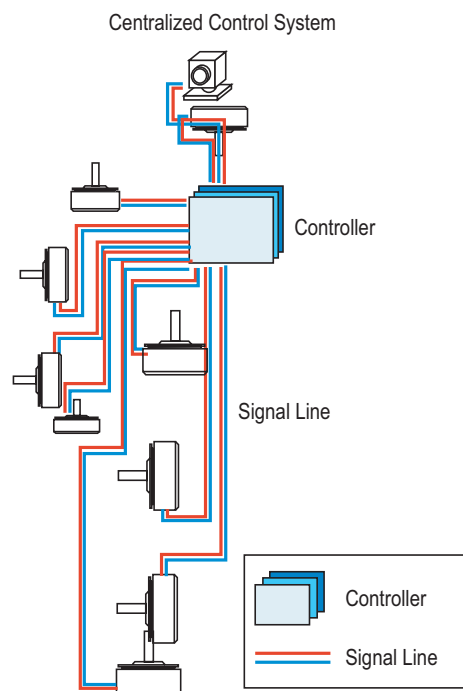


図 3.2: 集中制御システム

モデルがある [Miyashita *et al.* 04] . このモデルは通常の Robovie と比較して計算量が増加するため、その情報処理システムは、850MHz の PentiumIII プロセッサを 2 つ用いるシステム構成へと再設計された . その際には、現状の Robovie を改造するのではなく、全ての機械構造や電気・電子ハードウェアを再度設計し直し、初期段階から開発し直さなければならなかった . このことから、単一プロセッサの集中制御システムの構成では、より高度な行動を行ったり、新たな機能を追加する際にはプロセッサ資源が不足する可能性がある . その場合には、搭載する情報処理システムを高速プロセッサへ移行、またはプロセッサの増設などの対処が必要となる . つまり、集中制御システムでは、情報処理能力が既に限界に達していると言える . 集中制御システムは、プロセッサの増設やシステムに新たな要素技術、I/O、デバイスを追加する際、全ての機械構造や電気・電子ハードウェア、ソフトウェアを初期段階から再度設計・開発し直さなければならない事態を引き起こす可能性があるため、ヒューマノイドロボットのシステム構成には適していないと考えられる .

また、ヒューマノイドロボットはセンサやアクチュエータが体内に分散して配置してあるため、胴体に配置する制御用 I/O ボードから各 I/O までの配線が複雑になり、開発やメンテナンスにコストがかかってしまう . また、それらの配線は屈曲する狭い関節部分をメカや電気・電子ハードウェアと接しながら通り抜けていくため、ロボットの動作や外部からの衝撃によって断線等の障害を受けやすい構成である .

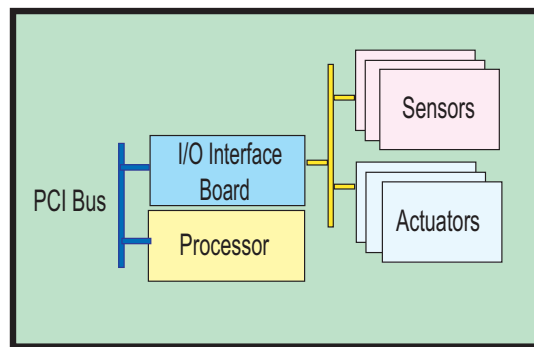


図 3.3: Robovie の電気・電子ハードウェア構成

Robovie は、制御用 OS として Linux のリアルタイム拡張である RT-Linux[Yodaiken *et al.*] を採用している。RT-Linux において、通常の Linux のカーネルは RT-Linux のプロセスの 1 つとして動作しており、RT-Linux はこの部分に変更を加えることなく、新たに別のスケジューラや割り込みハンドラを実装するという方法でリアルタイム処理を可能にしている。RT-Linux では、リアルタイムプロセスはカーネル空間で実行されるダイナミックロードブルモジュールとして記述される必要があり、リアルタイムプロセスはユーザ空間で実行される非リアルタイムプロセスと実時間 FIFO のみで通信可能である。非リアルタイムプロセスからは Linux の豊富なデバイスドライバが使用可能であるが、リアルタイムプロセスは非リアルタイムプロセスと通信することで間接的にのみ Linux のデバイスドライバを使用することができる。Robovie では、リアルタイム性の厳しいモータ制御をリアルタイムプロセスとして、画像処理、音声認識、行動計画を非リアルタイムプロセスとしては、設計している。非リアルタイムプロセスとして設計されている行動計画などの周期タスクは、Linux のタイマー機能を使用し、試行錯誤を繰り返し作り込むことで実現する必要がある。しかしながら、そのようなソフトウェア構成ではカーネル空間でのタスク実行に比べ周期精度が劣る、ハードウェアレベルの割り込みによる遅延が発生してしまう等のデメリットがあり、安定し信頼できるシステムの構築やトップダウンに設計実装することが困難であり、高度な情報処理能力を実現するためには相当なノウハウを要する。

Robovie は、要素技術をシステム統合化する技術として、図 3.4 で示すソフトウェアアーキテクチャである Situated Module Architecture[Ishiguro *et al.* 99] を提唱している。図 3.4 が示す Sensor Module, Actuator Module, Reactive Module は、ロボットの基本的な行動機能を提供するモジュール群である。ロボットの特定の機能は、Situated Modules を次々に追加することで実現する。Situated Modules は前提条件部 (Pre-Conditions) と行動部 (Behavior) から構成され、If-Then ルールに従って行動計画を行なう。Situated Modules は Sensor Module からの状態を基に次の行動を決定し、Actuator Module がその行動に従っ

て動作する．Robovie のソフトウェア構成は，Situating Modules を追加することで新たな行動（シナリオ）をロボットに実装することができ，容易に実験を行える利点がある．

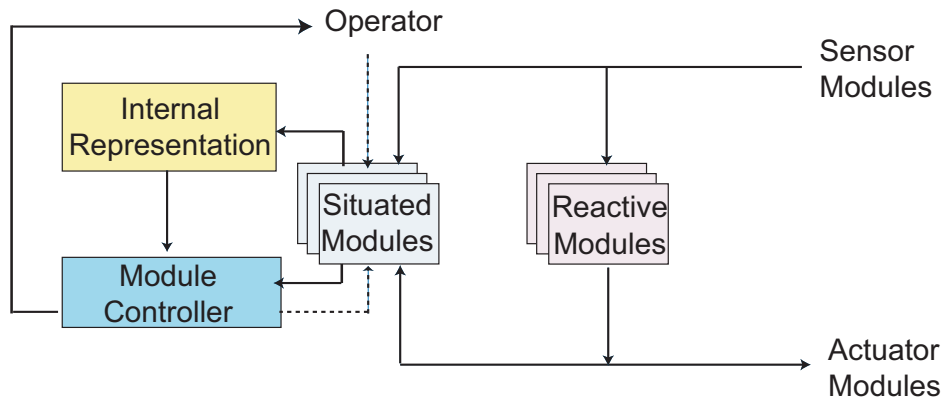


図 3.4: Robovie のソフトウェア構成

3.2.2 H7

H7 [Kagami *et al.* 02] は東京大学 井上・稲葉研究室で開発された 36 自由度の全身型ヒューマノイドである．H7 は，未来開拓学術研究推進事業「マイクロ・ソフトメカニクス統合による高度生体機能機械」プロジェクトの一環として，知能により実現される柔軟性と，柔軟分布触覚を統合し，ヒューマノイドロボットの知能ソフトウェア統合を目的とした統合研究のためのプラットフォームとして開発された．H7 の情報処理システムは，CPU カード (750MHz の PentiumIII プロセッサ) を 2 つ利用し，胴体部に設置してある CompactPCI を介して HDD，無線 LAN カード，IEEE1394 画像キャプチャーカード，サウンドカード，制御用 I/O ボードを接続する構成である．膨大な計算量の必要とするソフトウェアはネットワーク上の資源を利用して処理している．

H7 は制御用 OS として RT-Linux を採用し，要素技術をシステム統合化する技術として，以下のようなソフトウェアの階層化を提唱している．

- Application Layer :
インタラクション等のアプリケーションを設計する層
- Network Layer :
ネットワーク上の計算機を用いて膨大な計算量の必要とする制御ソフトウェアを設計する層

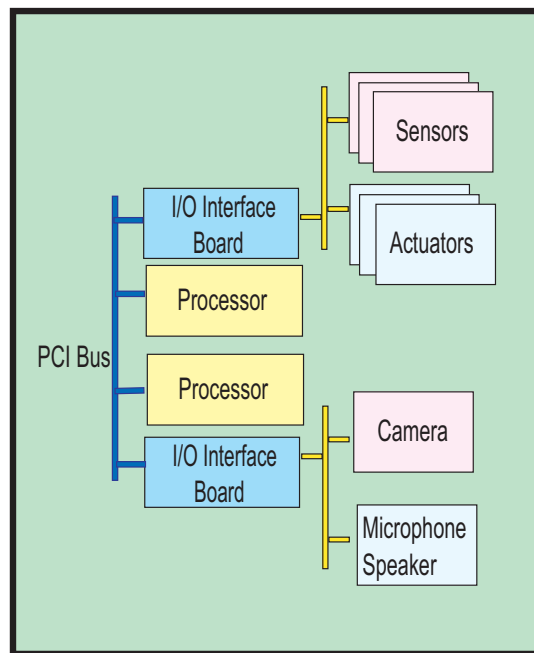


図 3.5: H7 の電気・電子ハードウェア構成

- On-body Layer :
比較的時間に余裕のある運動制御，画像処理を非リアルタイムプロセスとして設計する層
- Kernel Layer :
モータ制御，画像キャプチャをリアルタイムプロセスとして設計する層

以上のように，階層化したソフトプラットフォームは，各層を独立して並行に設計・実装できるため，ソフトウェア開発の効率がよいと言える．また，階層化したソフトプラットフォームは，各層毎に再利用可能であるために，異なったロボット間でも同じソフトウェアをそのまま使える利点がある．しかしながら，Robovie と同じく，集中制御型ロボットシステム固有の特徴が，H7 のシステム構成法の問題点として挙げられる．

3.2.3 HRP-2

HRP-2 [Kaneko *et al.* 02] は，経済産業省・新エネルギー・産業技術総合開発機構 (NEDO) で研究開発された屋外共同作業応用を実現するための人間型ロボットのプロトタイプである．HRP-2 では，不整地歩行，転倒制御，転倒回復動作を実現している．

HRP-2 の情報処理システムは，図 3.7 のように，運動制御用プロセッサ (1.2GHz の PentiumIII プロセッサ) と 3 次元画像処理用プロセッサ (1.2GHz の PentiumIII プロセッサ) が

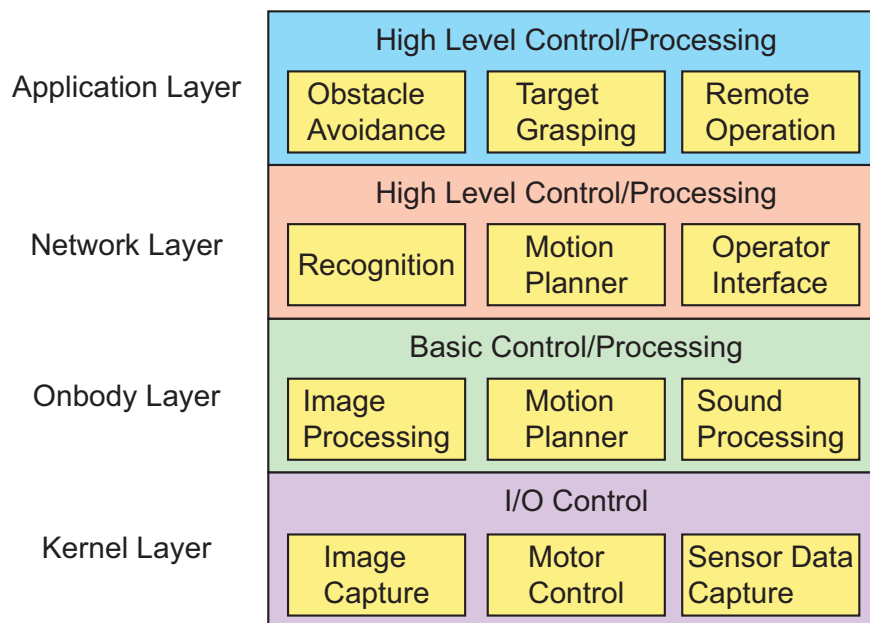


図 3.6: H7 のソフトウェア構成

別々のPC マザーボード上に実装され、2枚のマザーボードが100MbpsのEthernetで接続する構成である。RobovieやH7と同様に集中型ロボットシステムが引き起こす配線問題や、Ethernetを用いたリアルタイム性の欠如したプロセッサ間通信が問題点として挙げられる。

HRP-2は制御用OSとしてART-Linux [石綿 他 98][株式会社ムービングアイ]を採用している。ART-Linuxは、ロボットOSに特化して設計されており、ユーザ空間でリアルタイムプロセスを記述可能であり、リアルタイムプロセスからLinuxの豊富なデバイスドライバを利用可能という特徴を持ち、RT-Linuxの欠点を解決している。

HRP-2の要素技術をシステム統合化する技術として、ヒューマノイドロボットのシミュレータ及び制御ソフトウェアであるOpenHRP(Open Architecture Humanoid Robotics Platform)[Kanehiro *et al.* 02]を提唱している(図3.8)。シミュレータでは、ヒューマノイドロボットの動力学のシミュレーション、視野画像の生成を行うことができる。制御ソフトウェアでは、ヒューマノイドロボットの二足歩行、体操等の全身動作を制御することができる。運動制御システムはプラグインと呼ぶソフトウェアモジュールを基本単位として構成されており、プラグインマネージャがそれらの実行制御を行う。システム起動時にはアダプタプラグインと呼ぶ1つのプラグインを除いて他のプラグインは存在せず、ロボットの行う作業に応じて必要なプラグイン群が動的にロードされてシステムに組み込まれて制御を行う。マネージャは2つのスレッドからなり、これらのスレッドは非同期に並列実行される。1つ目のスレッドはインタプリタを実行し、プラグインの組込み/取り外しや

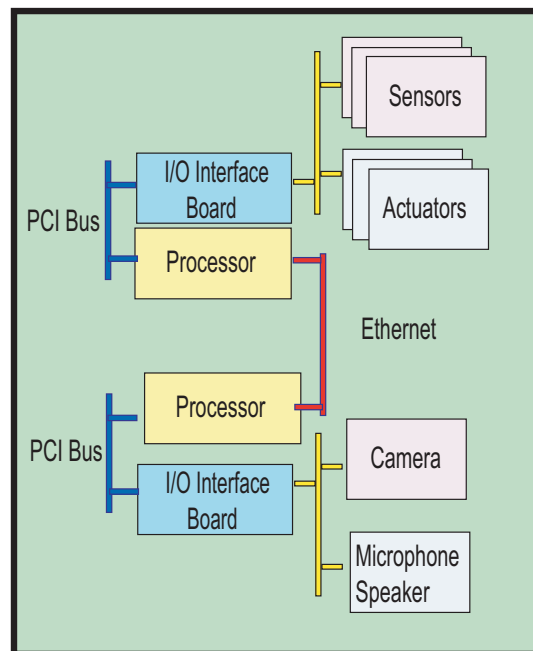


図 3.7: HRP の電気・電子ハードウェア構成

プラグインのパラメータ変更のようなプラグイン群の管理処理は、ユーザからの入力やスクリプトの解釈に基づいてインタラクティブに実行する。2つ目のスレッドはプラグイン群の制御計算を実行するスレッドであり、実行待ち行列に組み込まれたプラグイン群を先頭から順に繰り返し実行することでロボットを制御している。OpenHRPの特徴は、シミュレータ上で開発したソフトウェアが、バイナリ互換で実機に適用できる点と、標準的分散オブジェクトシステムCORBA Orbit/E [IONA]を利用することにより拡張が容易で、ネットワーク上に分散可能なアーキテクチャとなっている点である。しかし、CORBAを含む分散オブジェクトには、処理が大きくりアルタイム性を保証しにくい、通信に対するオーバーヘッドが大きい、タスクのスケジューリングの仕組みが無い等の問題点がある。

3.3 分散制御システム

集中制御システムで生じる問題の解決策の1つとして、分散制御システムの導入が挙げられる。分散制御システムとは、図 3.9 のように、システム全体が複数のプロセッサから構成されており、そのプロセッサ間で互いに通信を行いながらシステム全体の制御を行うシステムのことである。この際、各プロセッサは自身の管理下のセンサやアクチュエータなどを独立して制御を行う。

代表的なロボットとして、ASIMO(本田技研工業) [Sakagami *et al.* 02], HERMES(ドイ

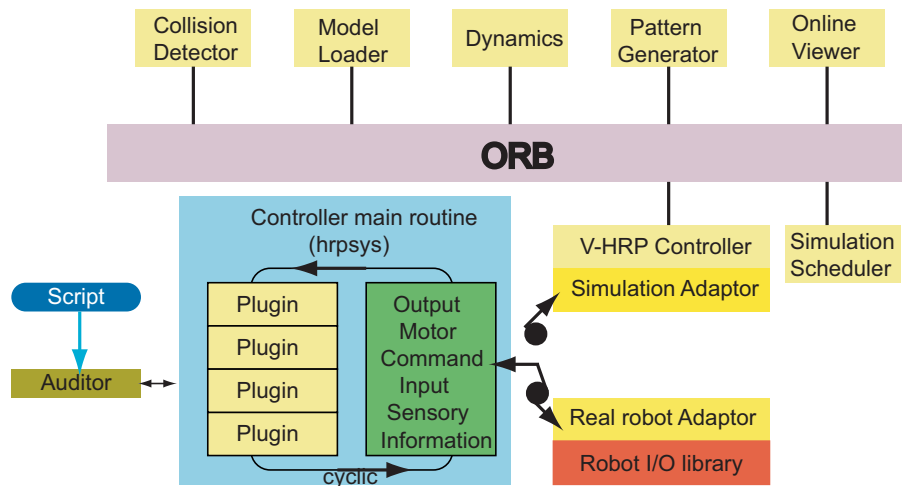


図 3.8: HRP のソフトウェア構成

ツ Bundeswehr 大学)[R.Bischoff *et al.* 99], SmartPal(安川電機)[株式会社安川電機], トヨタパートナーロボット(トヨタ自動車)[トヨタ自動車株式会社], QRIO(ソニー)[Ishida 04], EMIEW(日立)[Hosoda *et al.* 06], HRP-3(産業総合研究所)[Akachi *et al.* 05]などが挙げられる。以下では、分散制御システムの代表例として ASIMO, HERMES を取り上げて、分散制御システムの情報処理能力, 要素技術のシステム統合化技術の2点に関する利点・欠点を明らかにする。

3.3.1 ASIMO

ASIMO[Sakagami *et al.* 02]は本田技研工業で研究, 開発されている34自由度の全身型ヒューマノイドロボットである。

ASIMOの情報処理システムは, 図3.10に示すように, 音源定位用, フレームグラバ用, 画像処理用, 音声認識・音声合成用, ロボット制御・プランニング用, 外部サーバアクセス用の複数のプロセッサから構成する。また, ASIMOの情報処理システムのプロセッサ間通信は, 画像処理などの大容量転送を行う通信にはPCIを, 行動計画・動作実行を行う通信にはEthernetを用いている。

特に注目すべきは, 各アクチュエータが小型プロセッサを備え, その小型プロセッサ同士がリアルタイム通信規格であるアークネット[ARC]を用いて体内LANを構築している点である。このような形態の分散制御システムを導入することで, 情報処理システムは, 細粒度のリアルタイム処理でアクチュエータを制御できる, アクチュエータ関係の配線を省配線化ができる, メンテナンスや信頼性を向上できるなどの利点がある。また, 画像処理, 音声処理等の機能毎にプロセッサを割り当て, 並列処理することで, 高度な環境認識・

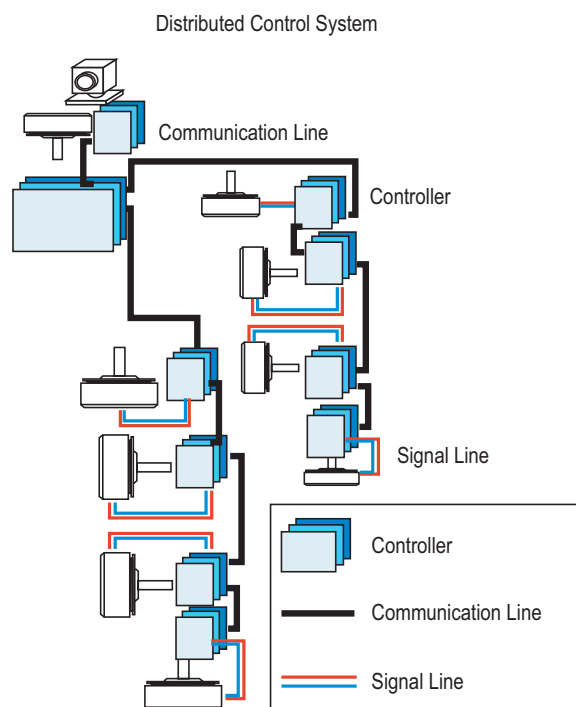


図 3.9: 分散制御システム

行動計画が、実現可能である。しかしながら、ASIMO が搭載する情報処理システムでは、アーム制御、歩行制御、画像処理、音声処理などを含めたロボット全体の行動計画を1つのプロセッサで集中的に処理している。そのため、システム全体の能力は、このプロセッサの情報処理能力に依存してしまう恐れがある。

ASIMO は制御用 OS として VxWorks [Wind River Systems] を使用している。VxWorks は組込分野では大きなシェアを占めており、ロボット分野でも多くのシステムで採用されている。

ASIMO は、要素技術をシステム統合化する技術として、Behavior-based Architecture [Arkin 98] の概念に基づいた分散エージェント指向アーキテクチャに基づいた図 3.11 に示す階層化したソフトウェアプラットフォームを提唱している。

3.3.2 HERMES

HERMES [R. Bischoff *et al.* 99] はドイツ Bundeswehr 大学で開発された 22 自由度の車輪移動式の上半身型ヒューマノイドである。HERMES は、屋内での運搬と対話などの対人サービスの研究のプラットフォームとして開発されている。

HERMES の情報処理システムは、図 3.12 に示すように、1つの汎用プロセッサ、5つの

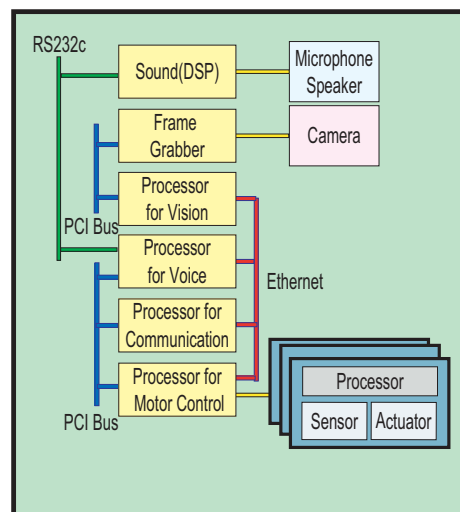


図 3.10: ASIMO の電気・電子ハードウェア構成

DSP(TI社製 TMS320C40) がリアルタイム通信規格である CAN バスを介して接続する構成である。また、ASIMO の情報処理システムと同様に、各アクチュエータが小型プロセッサを備え、その小型プロセッサ同士が CAN バスを用いて体内 LAN を構築している点である。特に注目すべきは、HERMES の全身のアクチュエータは、小型プロセッサ、CAN コントローラ、センサを備えたドライブモジュールとして、機械構造、電気・電子ハードウェア、ソフトウェアをモジュール化している点である。このドライブモジュールはモジュール間で Plug-and-Play が可能であり、システムのメンテナンスや再構成が可能である点で他のロボットよりも優れている。

しかしながら、HERMES の情報処理システムも ASIMO と同様に 1 つのプロセッサでロボット全体の行動計画を処理している点が問題として挙げられる。

HERMES は、要素技術をシステム統合化する技術として、図3.13に示す Situation-oriented Architecture を提唱している。ソフトウェア構成のコアは Behavior-based Architecture に基づいた Situation Assessment & Behavior Selection Module である。このソフトウェアモジュールは、タスクを実行するためのサブタスク実行の結果とタスク実行に関する知識をもとに、現在の situation を評価し、次の行動を生成・実行する。プランニングの際に必要なセンサ、アクチュエータ、マンマシーンインタフェース、知識データベースへのアクセスのためにデバイスの抽象化を行っている。図3.13中の Sensorimotor Skills のようにネットワーク内の全てのデバイス同士が Situation Assessment & Behavior Selection Module からの命令なしに協調できる機構を導入することで、行動の幅や性能がより向上すると考えられる。

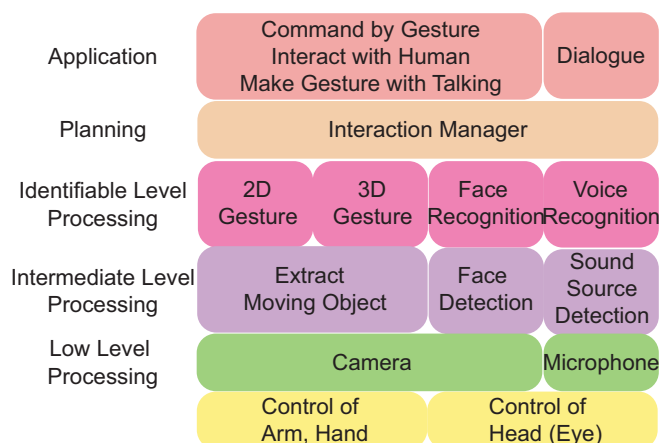


図 3.11: ASIMO のソフトウェア構成

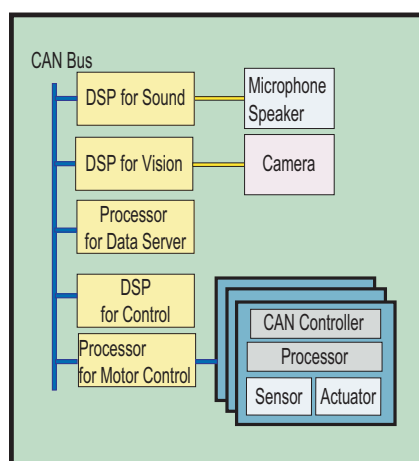


図 3.12: HERMES の電気・電子ハードウェア構成

3.4 本章のまとめ

本章では、従来のヒューマノイドロボットが搭載する情報処理システムの高度な情報処理能力、要素技術のシステム統合化技術の2点に着目して、従来のヒューマノイドロボットのシステム構成の比較・検討を行った。

ヒューマノイドロボットの情報処理システムの構成法は、関連研究で紹介したように、集中制御システムと分散制御システムの2種類に大きく分類することができ、それぞれ表 3.1 に示す特徴があることがわかった。

集中制御システムは、胴体部にプロセッサを置き、全てのアクチュエータやセンサの制御信号線を図 3.2 のように胴体部からスター型に配線するシステムであった。このシステ

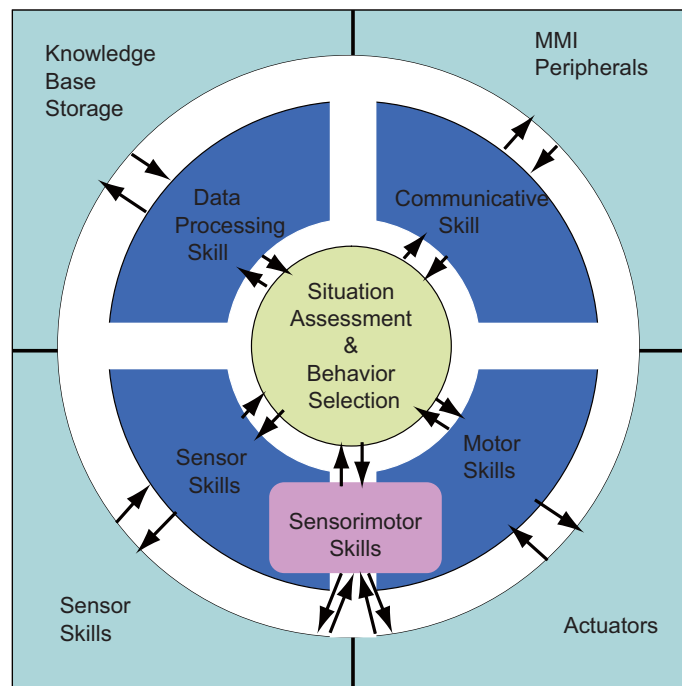


図 3.13: HERMES のソフトウェア構成

ムは、比較的シンプルな設計でシステムを構築できる、プロセッサ間通信を考慮しなくて良いなど利点がある。しかし、システムに新たな要素技術、I/O、デバイスを追加する際、機械構造や電気・電子ハードウェア、ソフトウェアを初期段階から再度設計・開発し直さなければならない点で、システム統合化技術の面に問題があった。

一方、分散制御システムは、図 3.9 に示すように、システム全体が複数のプロセッサから構成されており、そのプロセッサ間で互いに通信を行いながらシステム全体の制御を行うシステムのことであった。分散制御システムでは、画像処理部、音声認識部、行動計画部、各アクチュエータへそれぞれプロセッサを割り当てることで、集中型制御型システムに比べ処理能力を向上することができ、より複雑な行動や実験を可能としている。HERMESでは、アクチュエータ毎にモジュール化されており、他のロボットと比較して拡張性や柔軟性も実現している。しかしながら、分散制御システムにも複数の欠点が存在した。

1つ目は、ロボットの情報処理能力の向上には、分散化の粒度とプロセッサ間通信が重要になる点である。つまり、分散制御システムでは、制御アルゴリズムをできるだけ効率よく並列処理、分散処理を行うことできる情報処理システムを構築しなければならない。分散制御システムを構築するためには、まずシステム全体をどのように分割するのかを決定する必要がある。その際には、以下のようなファクターを元にして分割を行う。

- 分散制御アーキテクチャ(階層構造、フラット構造等)

表 3.1: 集中制御システムと分散制御システムの比較

	集中制御システム	分散制御システム
処理できるタスク量	少量	集中制御より向上
動作環境	少範囲の限定空間	集中制御より向上
システムの拡張性・柔軟性	難しい	集中制御より向上
メンテナンス性	難しい	集中制御より向上
耐故障性	低い	集中制御より向上
信頼性	低い	集中制御より向上
プロセッサの種類と個数	1つの高速なプロセッサ	複数のプロセッサ
プロセッサの消費電力	大きい	集中制御より向上
プロセッサの配置場所	胴体	胴体と各関節
プロセッサ間通信	必要なし	考慮する必要あり
ソフトウェア設計	比較的シンプル	複雑

- I/O の種類 / 数量
- 空間的な配置
- 演算量
- 通信量
- リアルタイム性
- 消費電力

システム分割の自動化は容易ではなく、システム設計者が上記のファクターを元にヒューリスティックに行う必要がある。上記のファクターにはトレードオフの関係になるものも多いため、システム設計者が最も重要だと考えるファクターから順に優先順位をつけ、分割方法を決定しなければならない。

従来の分散制御システムでは、画像処理部、音声認識部、各アクチュエータ、アーム制御、歩行制御などを含めるロボット全機能を管理する行動計画部は、1つのプロセッサ上で処理されている。つまり、従来の分散制御システムが搭載する情報処理システムの処理の流れは、集中制御システムが搭載する情報処理システムの処理の流れとなんら変わらないのである。この構成では、各要素技術の制御ソフトウェアをヒューマノイドロボットの行動計画部のソフトウェアへ組み込む際に、搭載しているプロセッサの余っている資源で制御性能を十分保証できるか、その制御処理を組み込むことで他のソフトウェアに影響を与えないかを、ボトムアップに調べる必要があり、ヒューマノイドロボットの制御ソフト

ウェアをトップダウンに設計・実装することが困難となってしまう恐れがある。また、行動計画部を1つのプロセッサで処理するため、そのプロセッサの肥大化や、そのプロセッサがボトルネックになってシステム全体のスループットの低下を招いてしまう恐れもある。

2つ目は、HERMESのようにロボット全体を機械構造、電気・電子ハードウェアを含めて、複数のサブシステムに分解できない点である。HERMESのようにシステムを複数のサブシステムへ分解できたとしたら、各要素技術を分離した研究・開発、新たな要素技術、I/O、デバイスの追加が容易に行うことができると考えられる。しかし、そのHERMESのシステム分割にも問題がある。HERMESは、システムをI/O毎にサブシステムとして分割している。そのため、各サブシステムは、非常に単純な機能しか備えておらず、サブシステム単体では動作することが不可能である。サブシステム単体でも利用や研究が可能なシステム分割手法を提案できるとしたら、サブシステムの発展とヒューマノイドロボットの性能向上が直結する、ヒューマノイドロボットの使用用途も広がるなど、ヒューマノイドロボットの性能が飛躍的に向上すると考えられる。

第4章

本研究の目的と動機

ヒューマノイドロボットは、少子高齢化・労働力不足の対策として期待されている。しかし現在までに開発されたヒューマノイドロボットはそのレベルに達していない。しかも現状は、その問題点を打開する技術も確立されていない状況である。

本研究は、(1) ヒューマノイドロボットが搭載する情報処理システムの情報処理能力不足、(2) 拡張性・柔軟性のあるシステム統合化技術の欠如というヒューマノイドロボットが抱えている2つの問題点に着目し、それぞれの問題に対して(1) 行動計画部の並列・分散処理、(2) システム再構成機構を解決策として提供するシステムアーキテクチャを提案し、問題点を解決することが目的である。

第5章

提案・設計方針

本章では、(1) ヒューマノイドロボットが搭載する情報処理システムの情報処理能力不足、(2) 拡張性・柔軟性のあるシステム統合化技術の欠如という問題点に対して、情報処理能力不足の原因と、拡張性・柔軟性を備える既存のロボットを調査・検討することで、これらの問題点を打開できるシステムアーキテクチャの設計方針を提案する。

5.1 並列分散処理化

5.1.1 ヒューマノイドロボットの情報処理能力向上

1980年代以降、マイクロプロセッサの性能向上は、主にスケールダウンによる低消費電力化、高速化、命令パイプラインの多段化等による命令レベルの並列化が主流であった。

しかし、2.2.3節でも述べたように、ゲート長が90nmに達するに及んでリーク電流の増大による低電力化の壁にぶつかり、命令レベルの並列度も4-8に達してこれ以上の並列化による効果が期待できない状況である。これに対して、マルチスレッディングのようなスレッドレベルでの並列化、クラスターのような粗結合並列化、プロセッサダイの上にそれぞれ単独に機能する複数のプロセッサを実装するマルチコアプロセッサ、メニーコアプロセッサが有効な手段と見なされつつある。しかし、そのIntel社のロードマップを見る限りでは、バッテリーで駆動するロボットにとっては、プロセッサの消費電力は依然大きいままである。従って、プロセッサの処理能力が向上したとしても、ロボットへ搭載できるとは限らないと言える。仮に低消費電力で処理能力の高い計算機が開発され、その計算機を複数個用いて情報処理システムを構成したとしても、処理能力を向上できるかどうかは、対象問題(ロボットの制御アルゴリズム)の並列性に依存する。

そこで、先に述べた環境認識処理、行動計画、動作実行の並列性について検討を行う。

視覚処理における距離画像の算出においては、多数の画像小領域のペアについて相関演算を施す処理が並列に実行できるだけでなく、相関演算そのものがMMX [Peleg *et al.* 97]のようなSIMD命令 [Raman *et al.* 00]によって並列化が可能である。実際に現在の多くのロボッ

トの視覚処理には、SIMD命令が使われることが一般的である。また、[Komuro *et al.* 03][神田 他 06]のような専用ボードの開発も行われている。同様に聴覚処理における音素系列の探索処理、音源定位処理も並列性があり、[黒滝 他 04][於久 他 02]のような専用ボードの開発も行われている。行動計画の中で大きな時間を占める計算は、物体同士の干渉を判定する幾何計算である。数十～数千の物体 N に対して N^2 のペアの距離計算が必要であるが、これらは相互依存関係がないので並列計算可能である。また、腕、足、頭など部位毎の順運動学、逆運動学も並列計算可能である。

このようにロボットの情報処理には、並列に動作できる機能が多く存在する。「並列処理」は専門用語で「コンピュータで、一連の処理を複数台の処理装置で同時に並行して行うこと」(大辞林)という意味である。並列処理と同じような言葉に並行処理がある。複数のプロセッサを使って複数のプログラムを(真に)同時に実行することを並列(Parallel)実行、1つのプロセッサで擬似的に実行することを疑似並列(Pseudo-Parallel)とか並行(Concurrent)実行と呼ぶ。

従来のヒューマノイドロボットが搭載する情報処理システムは、視覚処理、音声処理、行動計画を複数のプロセッサを利用して並列処理している。特に視覚処理、音声処理に関しては、先に述べたようにアルゴリズムをFPGAやASICなどの専用ハードウェアへ実装し、高速な並列処理を実現している。しかし、従来のヒューマノイドロボットは、行動計画処理を1つのプロセッサ上で並行処理している。つまり、行動計画処理を並列化することがロボットの情報処理能力向上に繋がる。そこで、本研究では、ロボット機能の並列性を向上することができる制御手法を提案することが目標である。

5.1.2 行動計画部の並列・分散処理化

本節では、行動計画部の並列・分散処理化について検討する。

前章でも述べたように、分散制御システム設計する際には、システム分割の粒度をどの程度に設定するかが大きな問題となる。分割の粒度が小さすぎると、プロセッサ間の通信量の増加、制御システムの複雑化を招いてしまう。一方、分割の粒度が大きすぎると、高性能な計算能力の必要性、柔軟性の損失を招いてしまう。

ASIMOやHERMESのように、ロボット全体の行動計画を1つのプロセッサで処理する必要があるのかどうか検討するために、本節では人間の動作に着目してみる。人間の行動は、(1)「友達と会話しながら、移動する」や(2)「移動しながら、読書する」など、複数の動作を同時に行っている。(1)の動作は、A「目や首を動かし相手を認識しつつ、耳の機能を働かせて相手の話を聞く」、B「移動する経路を考える」、C「目的地へ移動する」の大きく3つ分類することができる。(2)の動作も、D「移動する経路を考える」、E「目的

地へ移動する」, F「手を動かす」の大きく3つに分類することができる。

これらの動作は, ヒューマノイドロボットの動作に置き換えて考えてみると, (1)-Aは頭部(環境認識処理, 目や首の自由度の行動計画, 行動実行), (1)-Bと(2)-Dはシステム全体の行動計画部, (1)-C, (2)-Eは移動部(環境認識処理, 行動計画, 行動実行), (2)-Fはアーム部(環境認識処理, 行動計画, 行動実行)のタスクとして捉えることができる。つまり, ヒューマノイドロボットが搭載する情報処理システムは, 従来のヒューマノイドロボットの情報処理システムのように視覚処理, 音声処理, 行動計画部を並列処理する構成よりも, 頭部, システム全体の行動計画部, 移動部, アーム部という粒度で並列処理する構成の方が適していると解釈することができた。この考察で明らかになったことは, 従来のヒューマノイドロボットではシステム全体の行動計画部を並行処理しているが, 実際には複数の行動計画が並列で処理可能であるということである。この考察に基づいて, 行動計画を並列処理できる情報処理システムを確立することができれば, ヒューマノイドロボットの情報処理能力は向上することができると考えられる。

5.2 システム再構成機構

5.2.1 システム再構成機構とは

この節では, 現在のヒューマノイドロボットが抱えている問題の解決につながると考えられるシステム再構成機構について述べ, ヒューマノイドロボットに適應できるかどうか検討を行なう。

システム再構成機構とは, システムが動作しながらにして, そのシステムを構成する電気・電子ハードウェアおよびそれに伴うソフトウェアを部分的に, 環境や用途に応じて自由に変更可能とする機構のことである。この機構は, コンピュータの分野において, USB [USB] や IEEE1394 [Association] などの規格に対応した周辺機器をコンピュータに挿すだけでシステムがそれを自動認識して使えるようにする「Plug-and-Play」や, またそれを電源が入った状態で可能とする「Hot Plug」としてよく知られる。システムの拡張性や柔軟性, ユーザに対する利便性の向上などの理由から, この機構はパーソナルコンピュータに限らずニーズは高い。システム再構成機構の考え方は, 未知の環境あるいは動的に変化する環境に柔軟に対応しながら, 継続した動作や作業を行うことが求められる惑星探査ロボットやレスキューロボットなどで既に採用されている。以下では, 実用化レベル, 研究レベルのシステム再構成機構の例を挙げ, その特徴と技術的課題を明らかにする。

表 5.1: Plug-and-Play に必要な条件

対応に必要なもの	条件	説明
電気・電子ハードウェア	機械的，電氣的に接続できること	コネクタの機械的な形状の規定や，信号電位，タイミング，ハードウェアプロトコルなど電氣的な規定が必要である
電気・電子ハードウェア or OS	接続の有無が判別できること，分離できること	機械構造，電氣的な方法，ソフトウェアにより，接続/分離の有無を判別できることが必要である．
電気・電子ハードウェア and OS	何が接続されているか識別できること	複数の識別子を用いて，システムへ接続されているデバイスを一意に識別する方法が必要である
電気・電子ハードウェア and OS	必要な情報について規定されていること	接続されている機種に依存する必要情報の内容の規定と，必要な情報を取り出す方法（プロトコル）が必要である
OS	リソースの割り当てができること	接続機器にリソースが必要な場合，既存リソースと衝突しないように割り当てる必要があるとなる
OS	デバイスドライバの割り当てができること	接続された機器に対して必要なデバイスドライバを自動的に見つけることが必要である．デバイスドライバと割り当てリソースが一致し，デバイスドライバが動作可能でなくてはならない

5.2.2 Plug-and-Play

システム再構成機構を実現した例として，Plug-and-Play がある．Plug-and-Play とは，コンピュータの分野における周辺機器を拡張する際の自動コンフィギュレーション機能のことを指す．ユーザが電気・電子ハードウェアを挿す (Plug) だけで使用可能 (Play) とするという機構である．この概念はそもそも Microsoft 社と Intel 社により提案され，Windows95 で初めて導入された．Plug-and-Play は，拡張デバイスのための I/O アドレス，拡張メモリ，割り込みチャンネル，DMA チャンネルなどのデバイスを管理する機構である．対象としては ISA バスや PCI バス等数多くのバスインタフェースが含まれる．文献 [ISA94] では，Plug-and-Play を実現するためにシステムが備えなければならない条件を表 5.1 のようにまとめている．表からも容易に推察できるように，Plug-and-Play を実現するためには電気・電子ハードウェア及びソフトウェア両面からのサポートが必要である．そしてこのような Plug-and-Play や Hot Plug はその利便性から利用される領域が拡大している．

5.2.3 モジュールロボット

Plug-and-Play の概念をロボットへ適応した，モジュールロボットというロボット技術がある．このモジュールロボットとは，「モジュール」と呼ばれるサブシステムの結合からなるロボットであり，周囲の環境に合わせてその機能や形態を変化させながら，その環境に応じた移動方法や作業方法を行うことができるロボットの総称である．更に，モジュールロボットは，物理的にもモジュール毎に着脱を容易な構造にすることによって，モジュール単位での再利用や交換が可能になり，故障や仕様変更，システム拡張に対して柔軟に，低コストで対応できる利点がある．

モジュールロボットに関する研究は次の2種類に大別することができる．1種類目は，システムを構成するモジュールは基本的に1種類，あるいはあっても数種類で，非常に単純な機能しか備えておらず，これらを多数接続することで，複雑な形態や動作を可能にするロボットである．その代表的なロボットとして，M-TRAN [Kamimura *et al.* 01]，Cebot [Kawauchi *et al.* 94]，CONRO [Castano *et al.* 00]，PolyBot [Yim *et al.* 00] が挙げられる．2種類目は，システムを構成する各モジュールはロボットを構成するある機能を備えており，モジュールの追加や交換によって機能拡張や変更に対し，柔軟且つ低コストに対応することを目指したロボットである．その代表的なロボットとして，OPEN-R [Fujita *et al.* 99] や MFMRS [Matsumoto *et al.* 02] が挙げられる．この種類のモジュールロボットにおいてモジュールの組み換え，すなわちシステム再構成は，新しい機能を持つモジュールを追加したり，既存のモジュールを取り外したりすることによって，システム全体の機能を再構築することを意味する．

これら2種類のモジュールロボットは，次の6つの項目によって，表5.2のように比較することができる．

- ロボットの種類
- 均質性:ロボットが同一のモジュールから構成されているか
- タイミング:システム構成の変更タイミングがオンラインかオフラインか
- 再構成能力:システム構成をロボット自身で変更できるかユーザが手動で変更するか
- 制御手法:モジュールロボットが集中制御システムか分散制御システムか
- 自律性:モジュール単体で自律して動作を行うことができるか

以下では，M-TRAN，Cebot，OPEN-Rの詳細を述べ，モジュールロボットの利点，欠点を明らかにする．

表 5.2: 再構成可能なロボットの比較

	タイプ	均質性	タイミング	再構成能力	制御手法	自律性
M-TRAN [Kamimura <i>et al.</i> 01]	移動ロボット	同形	オンライン	自動	集中制御	×
Cebot [Kawauchi <i>et al.</i> 94]	マニピュレータ	異形	オンライン	自動	分散制御	×
CONRO [Castano <i>et al.</i> 00]	移動ロボット	同形	オフライン	手動	集中制御	×
PolyBot [Yim <i>et al.</i> 00]	移動ロボット	同形	オフライン	手動	集中制御	×
MFMRs [Matsumoto <i>et al.</i> 02]	移動ロボット	同形	オフライン	手動	集中制御	×
OPEN-R(AIBO) [Fujita <i>et al.</i> 99]	4足ロボット	同形	オフライン	手動	集中制御	×
HERMES [R.Bischoff <i>et al.</i> 99]	ヒューマノイド ロボット	同形	オフライン	手動	集中制御	×

M-TRAN

M-TRAN(Modular Transformer) は、産業技術総合研究所と東京工業大学が1998年から開発している、自己再構成型モジュールロボットである。

M-TRANは、図5.1のように、立体的な構造を変えることで環境に適応することができ、例えば、段差のある環境下では、脚型に変形して歩き、隙間では、へび型構造に変形して動作することができる。各モジュールが、電気・電子ハードウェア、制御ソフトウェア、他のモジュールと着脱可能な機械構造、通信機構を備えることで、モジュール単体でも動作可能であり、複数組み合わせても動作可能であるという拡張性と柔軟性を実現している。このことから、モジュールロボットは、自動車やパーソナルコンピュータのように機能や形態をオリジナルにカスタマイズできる機能をユーザへ提供できると考えられる。

Cebot

Cebot(Cellular Robotic System) は、様々な環境下で様々なタスクを実行するためにロボットの電気・電子ハードウェアとソフトウェアともに再構成可能である「cell」と呼ばれるモジュールから構成されるモジュールロボットシステムである。各 cell は、独立したセンサシステム、通信システム、知識データベース、動く、曲げる、回転する等の基本動作

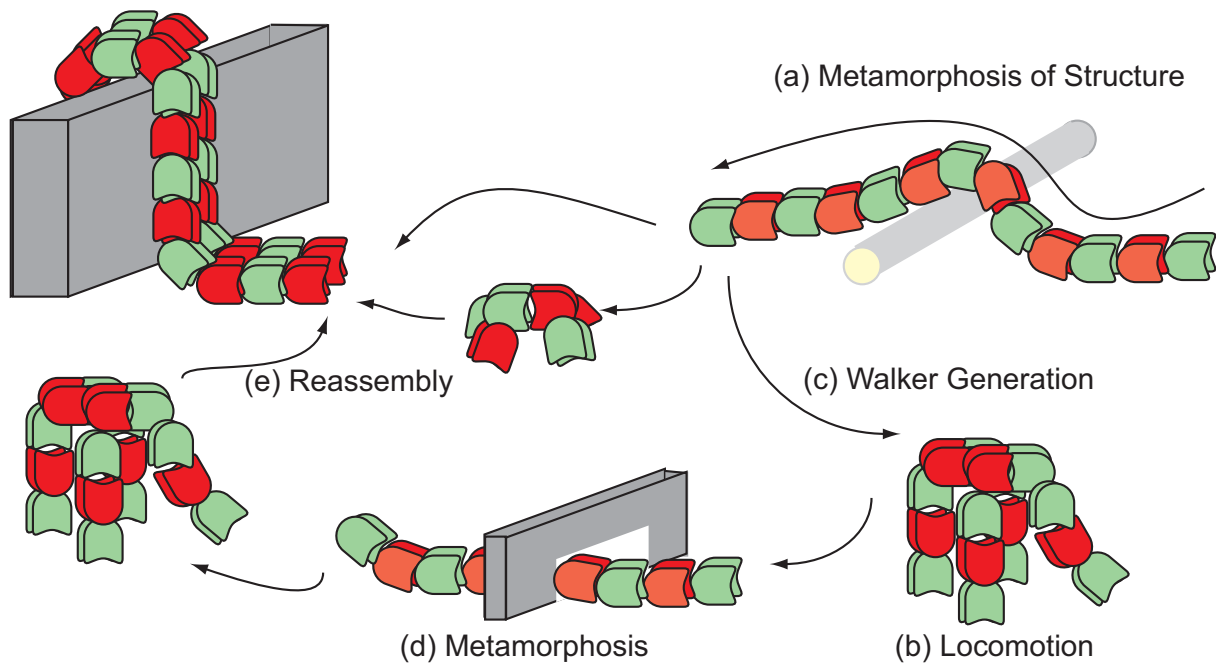


図 5.1: M-TRAN のシステム構成

機能を備えている．ユーザは cell を任意に組み合わせることでマニピュレーション機能を実現することが可能である．

OPEN-R(AIBO)

OPEN-R [Fujita *et al.* 99] は，ロボットシステムの拡張性・柔軟性・移植性・操作性に注目した，エンターテインメントロボット用のシステムアーキテクチャである．OPEN-R では，これらの性質を実現するために以下の技術を提唱している．

- 用途に応じたモジュール化
基本モジュールと拡張用モジュールから構成されるモジュールロボットを定義する
- Configurable Physical Components(CPC)
各モジュールを CPC と定義し，お互いに Plug-and-Play を可能とする
- レイヤ構造
Hardware Abstraction Layer(HAL) , System Service Layer(SSL) , Application Layer(APL) の3つのレイヤにソフトウェアを分割する
- オブジェクト指向プログラミング
オブジェクト指向 OS である Aperios [Fujita 00] 上で，再利用可能なソフトウェアコ

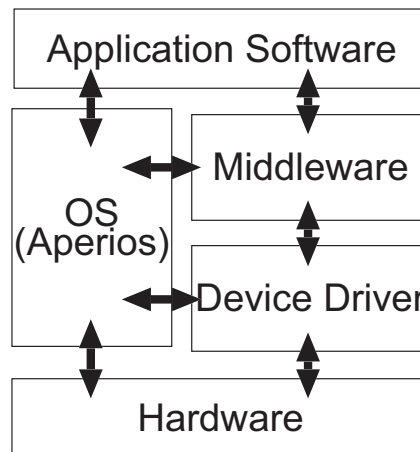


図 5.2: OPEN-R のシステム構成

ンポーネントを実現する。

図 5.2 は OPEN-R のシステム構成を示している。図 5.2 から明らかなように、OPEN-R は、電気・電子ハードウェアとソフトウェア両面の再構成機構を考慮している。OPEN-R の電気・電子ハードウェアは、各モジュール間を接続する専用のソケットを提案している。このソケットの構造は USB の構造に類似しており、Plug-In 時において信号線が接続される前に電源線が接続できるように、電源線を信号線より長くしている。OPEN-R のソフトウェアは、図 5.2 に示す階層化した構造を採用している。この階層化では、上位レベルが電気・電子ハードウェア構成の変化を意識することがないように、下位レベルでその変化を吸収し、抽象化している。この抽象化により、アプリケーションに電気・電子ハードウェアを隠蔽した制御を可能としている。

このように、OPEN-R は拡張性、柔軟性、移植性、操作性に重点を置いたシステムアーキテクチャであり、システム全体として持つ機能が、新しい機能を持つモジュールを追加したり、取り外したりすることによって再構築可能である。

5.2.4 モジュールロボットに関するまとめ

システム全体をモジュール化・部品化しているモジュールロボットは、新しい機能を持つモジュールを追加したり、取り外したりすることによってシステム全体として備える機能の再構築が可能である点、使用環境に応じて任意の組み合わせで動作可能である点が利点であった。このようなモジュールロボットを実現するためには、各モジュールが、モジュール単体として動作するための電気・電子ハードウェア、制御ソフトウェアと、他のモジュールと着脱するための機械構造、他のモジュールと協調動作するための通信機構を備える必



図 5.3: 要素研究の研究体系

要がある。一方，モジュールロボットは，各モジュール毎にモジュールと着脱するための機械構造を要するために，一般的なロボットと比較して重量が増加してしまう欠点があることも明らかになった。

5.3 設計方針

ヒューマノイドロボットの動作は，複数の機能が並列に動作していることが考察できた。モジュールロボットは，複数のモジュールを組み合わせることでシステムを構成することで，環境や用途に応じてシステムの状態を変化できる機能をロボットへ付加するアプローチを通して，柔軟性や拡張性の向上を実現していることが考察できた。

以上より，ヒューマノイドロボットの各機能の機械構造，電気・電子ハードウェア，ソフトウェアをモジュール化し，それらモジュールを組み合わせることでシステムを構築するモジュール型のヒューマノイドロボットを実現することができれば，情報処理能力の向上と，ヒューマノイドロボットもモジュールロボットと同等の柔軟性や拡張性の向上を実現できるのではないかと考えられる。

一方で，図 2.6 に示したロボット技術に着目すると，図 5.3 に示すように移動ロボット，頭ロボット，マニピュレータなどの単機能を持つロボット毎の開発が最も盛んに進められており，それら単機能ロボットを構成する電気・電子ハードウェア設計 [鏡 他 03][前野 他 98] やソフトウェア [Breazeal 99][Nakadai *et al.* 02]，アルゴリズム [加賀美聡 他 02][小林 他 03] などの研究が十分な成果を挙げている。これらの要素技術は，先に述べたヒューマノイドロボットの並列して動作している機能と一致していることがわかる。そこで，ヒューマノイドロボットを複数のモジュールに分割する際の粒度は，要素研究の粒度で分割するのが適していると考えられる。

以上のことから，本研究では，図 5.4，図 5.5 に示すように「通常は，単体の要素技術ロボットとして動作し，ヒューマノイドロボットに統合化された際には，ヒューマノイドロボットの部品として動作する」という設計概念を持つ再構成可能なモジュール型ヒューマノイドロボットのシステムアーキテクチャを提案する。

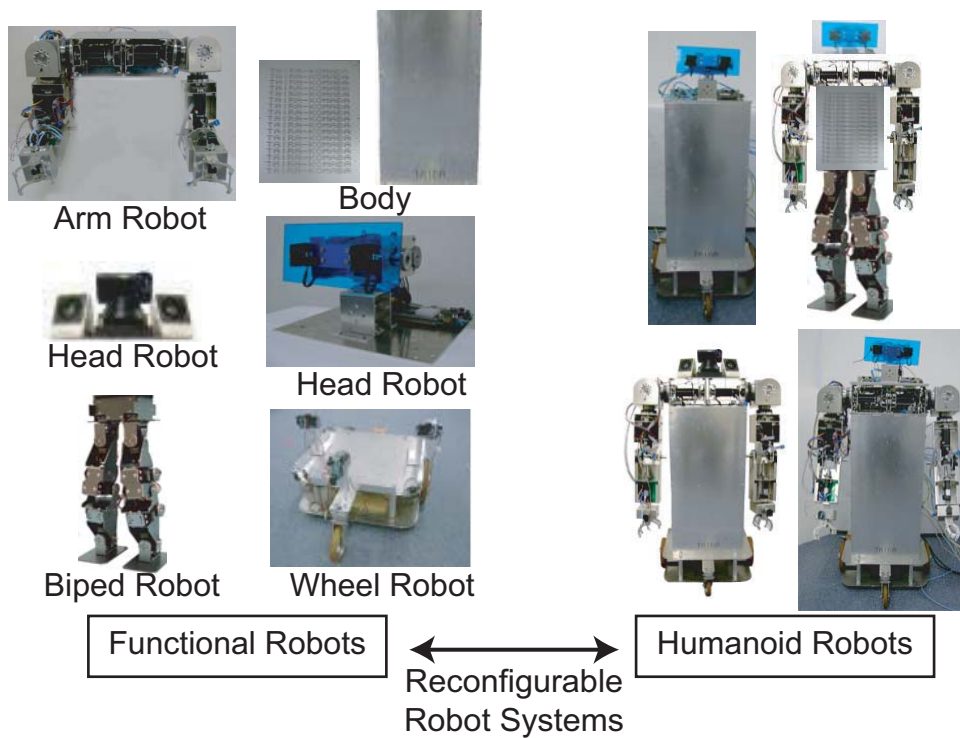


図 5.4: 再構成可能なモジュール型ヒューマノイドロボット

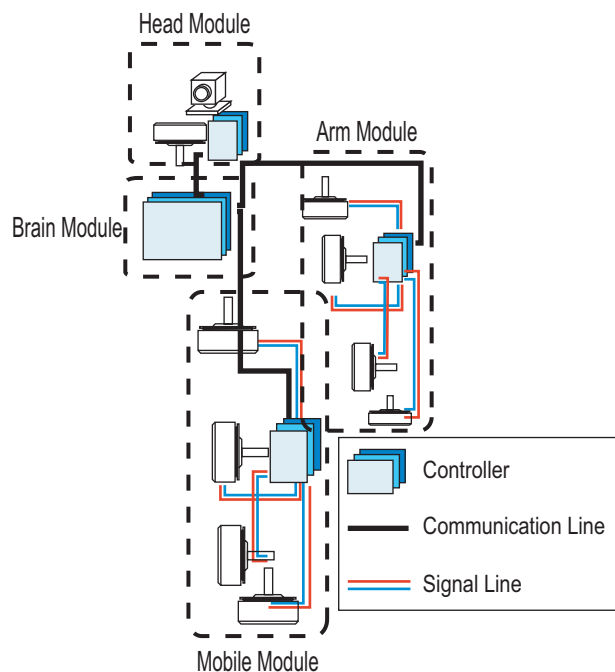


図 5.5: 再構成可能なモジュール型ヒューマノイドロボットの構成図

表 5.3: 再構成可能なモジュール型ヒューマノイドロボットの位置付け

	再構成可能なモジュールシステム	集中制御システム	分散制御システム
処理できるタスク量	多量	少量	集中制御より向上
動作環境	広範囲	少範囲	集中制御より向上
システムの拡張性・柔軟性	再構成機構により実現	難しい	集中制御より向上
メンテナンス性	モジュール毎に行える	難しい	集中制御より向上
耐故障性	分散制御より向上	低い	集中制御より向上
信頼性	分散制御より向上	低い	集中制御より向上
プロセッサの種類と個数	要素技術の処理に必要な計算機分	1つの高速なプロセッサ	複数のプロセッサ
プロセッサの消費電力	要素技術の処理に必要な計算機分	大きい	集中制御より向上
プロセッサの配置場所	モジュール毎	胴体	胴体と各関節
プロセッサ間通信	×考慮する必要あり	必要なし	考慮する必要あり
ソフトウェア設計	×複雑	比較的シンプル	複雑
ロボットの重量	×増加	-	-

本システムアーキテクチャに基づいたヒューマノイドロボットと従来のヒューマノイドロボットとの違いは、腕ロボット、移動ロボット、頭ロボットなどの複数の要素技術からシステムを構成する点である。本アーキテクチャに基づく要素技術は、単体の自律ロボットとして動作するために専用の機械部、電子部、ソフトウェア部を備え、ヒューマノイドロボットの一部としても動作する可能である。このような機能別ロボットの組み合わせでシステムを構成するモジュール型ヒューマノイドロボットは、ユーザに対して自動車やパーソナルコンピュータのように使用用途や環境に応じてシステムの構成や機能を変更できる再構成機構を提供する。さらには、モジュール型ヒューマノイドロボットは、新たに開発した機能・アルゴリズム・デバイスを備えた機能別ロボットをシステムへ容易に組み込むことができ、従来のヒューマノイドロボットと比較して拡張性と柔軟性の高いシステムを実現する。表 5.3 と表 5.4 は、提案する再構成可能なモジュール型ヒューマノイドロボットシステムの特徴を示している。

しかしながら、モジュール型ヒューマノイドロボットを実現するためには、下記に挙げる問題点を解決する必要がある。

- 問題点 (1)：制御アルゴリズムのモジュール型ヒューマノイドロボットへの適用
 - － 全身運動
 - － モジュール間通信 (体内 LAN)
- 問題点 (2)：ヒューマノイドロボットの再構成機構
 - － 制御ソフトウェアの管理
 - － モジュールの管理
 - － モジュールロボットの欠点である重量の増加への対応

そこでまず初めに，問題点 (1) を解決する手法を提案し，ヒューマノイドロボットの制御アルゴリズムを並列分散処理化することが可能なのか検討する．その検討結果上に再構成機構を設計することで，再構成可能なモジュール型ヒューマノイドロボットを実現する．

5.4 前提条件

本研究が提案する複数の自律ロボットを組み合わせることでシステムの機能や形状を再構成できるヒューマノイドロボットは現在提案されていない．システム構成の変更タイミングがオンラインで，システム構成をロボット自身で変更できる「動的再構成可能なヒューマノイドロボット」が最も拡張性や柔軟性があることは明確であるが，本研究では，再構成可能なモジュール型ヒューマノイドロボットが，従来のヒューマノイドロボットが抱える問題点を解決できるのかどうか検討することが目的であるため，システム構成の変更タイミングがオフラインで，システム構成をユーザが手動で変更する「静的再構成可能なヒューマノイドロボット」の実現を目指す．静的再構成可能なヒューマノイドロボットでその優位性を明確にしたのち，得られた知見を基に動的再構成可能なヒューマノイドロボットを実現する予定である．

従って，本研究では，静的再構成可能なヒューマノイドロボットを実現するために，図 5.6 が示すように，自律ロボットを複数組み合わせることでヒューマノイドロボットを構築できるシステムアーキテクチャを設計・実装することが目的となる．

静的再構成可能なヒューマノイドロボットの位置付けを表 5.4 に示す．

5.5 本章のまとめ

本章では，従来のヒューマノイドロボットが課題としていた問題点を解決できる静的再構成可能なモジュール型ヒューマノイドロボットの設計方針について述べた．次章では，この設計方針を具体化した詳細設計について述べる．

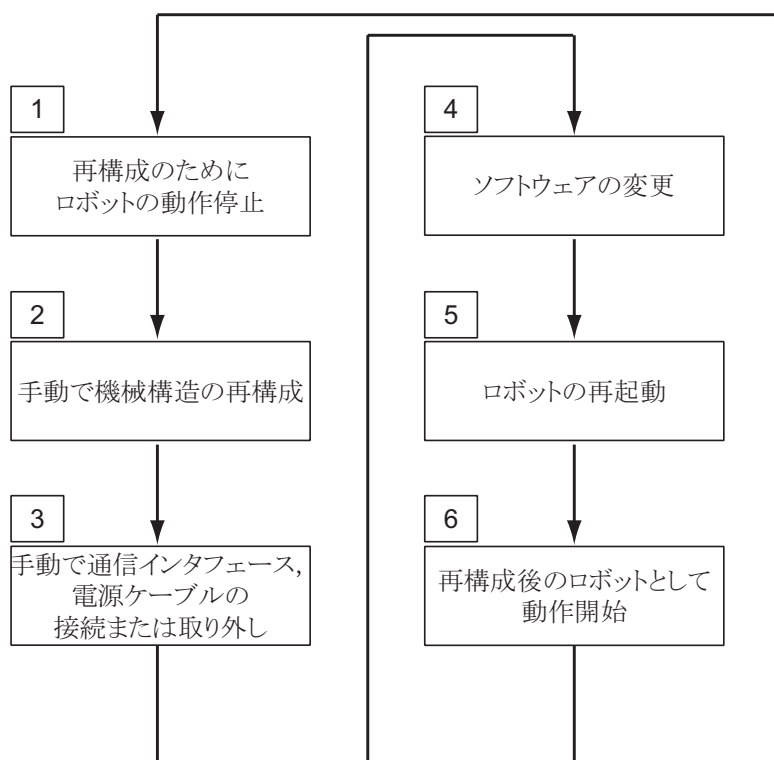


図 5.6: モジュール型ヒューマノイドロボットの再構成機構

表 5.4: 再構成可能なモジュール型ヒューマノイドロボットの位置付け

	タイプ	均質性	タイミング	再構成能力	制御手法	自律性
静的再構成可能なヒューマノイドロボット	ヒューマノイドロボット	異形	オフライン	手動	分散制御	
動的再構成可能なヒューマノイドロボット	ヒューマノイドロボット	異形	オンライン	自動	分散制御	
M-TRAN [Kamimura <i>et al.</i> 01]	移動ロボット	同形	オンライン	自動	集中制御	×
Cebot [Kawauchi <i>et al.</i> 94]	マニピュレータ	異形	オンライン	自動	分散制御	×
CONRO [Castano <i>et al.</i> 00]	移動ロボット	同形	オフライン	手動	集中制御	×
PolyBot [Yim <i>et al.</i> 00]	移動ロボット	同形	オフライン	手動	集中制御	×
MFMRs [Matsumoto <i>et al.</i> 02]	移動ロボット	同形	オフライン	手動	集中制御	×
OPEN-R(AIBO) [Fujita <i>et al.</i> 99]	4足ロボット	同形	オフライン	手動	集中制御	×
HERMES [R.Bischoff <i>et al.</i> 99]	ヒューマノイドロボット	同形	オフライン	手動	集中制御	×

第6章

システムアーキテクチャの設計

本章では，設計方針に基づいて再構成可能なモジュール型ヒューマノイドロボット用のシステムアーキテクチャの設計について述べる．

6.1 制御アルゴリズムの適用検討

本節では，まず始めに機能別ロボットと脳ロボットの提案を行う．続いて，それらのロボットを複数個組み合わせる構成するヒューマノイドロボットが，従来のヒューマノイドロボットと同様の制御アルゴリズムを実現できるのか検討する．

6.1.1 機能別ロボットの提案

設計方針に従って，本システムアーキテクチャは，1体のヒューマノイドロボットを，移動部，頭部，アーム部の要素技術の粒度で，複数のモジュールに分割する．本システムアーキテクチャでは，このようなモジュールを「機能別ロボット」として定義する．各機能別ロボットは，単体のロボットとして動作するために必要な機械機構，電気・電子ハードウェア，ソフトウェアを備える必要がある．また，機能別ロボットは，ヒューマノイドロボットへ統合される際に他の機能別ロボットと接続するために必要な機械構造，モジュール間通信機構，ソフトウェア機構も備える必要がある．このように機能別ロボットは，「一般的なロボットに必要なもの」と「統合化に必要なもの」から構成される．モジュールの汎用性や開発の自由度を向上させるために，「一般的なロボットに必要なもの」に関しては，そのユーザや研究者が自由に選択できることとする．再構成可能なモジュール型ヒューマノイドロボットが，他のヒューマノイドロボットと異なる点は「統合化に必要なもの」を備える一般的なロボットが複数組み合わせることでシステムを構成している点である．

以上より，本システムアーキテクチャでは，各機能別ロボットは以下の要素から構成すると定義する．

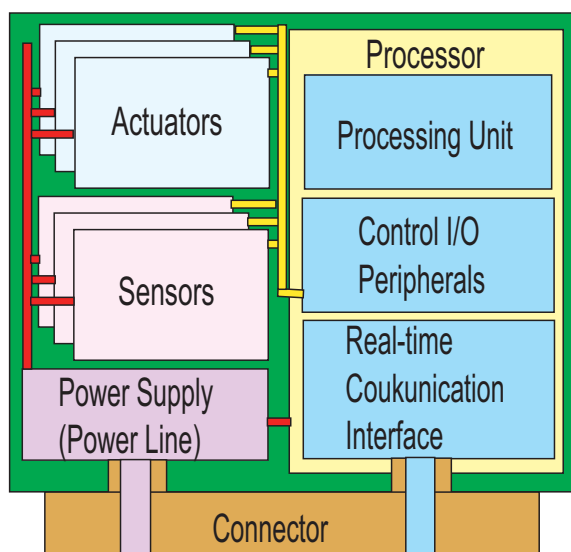


図 6.1: 機能別ロボット

- 機械構造：ボディ，モジュール間接続用コネクタ
- 電気・電子ハードウェア：プロセッサ，メモリ，I/O，モジュール間通信インタフェース
- ソフトウェア：リアルタイム OS，デバイスドライバ，アプリケーションプログラム，モジュール間通信ミドルウェア，モジュール管理ソフトウェア

図 6.1 は機能別ロボットを示している．図 6.2 は提案する機能別ロボットから構成するモジュール型ヒューマノイドロボットの例を示している．

6.1.2 脳ロボットの提案

本システムアーキテクチャでは，本アーキテクチャに基づくヒューマノイドロボットは，機能別ロボットの1つとして脳ロボット（脳モジュール）を必ず備えることと定義する．脳ロボットは人間の脳に相当する機能を処理するモジュールであり，各機能別ロボットの制御までは管理せず，機能別ロボットの制御値の入出力を利用してシステム全体の行動計画や管理を行なうアービタ用モジュールである．以上より，再構成可能なモジュール型ヒューマノイドロボット全体の構成は，脳ロボットを上位層として置き，それ以外の機能別ロボットを脳ロボットの下位層として位置づけた，階層構造の並列・分散システムである．

また，脳ロボットの導入は，行動計画や人工知能等の要素技術を，ヒューマノイドロボットから分離して開発できる利点もある．

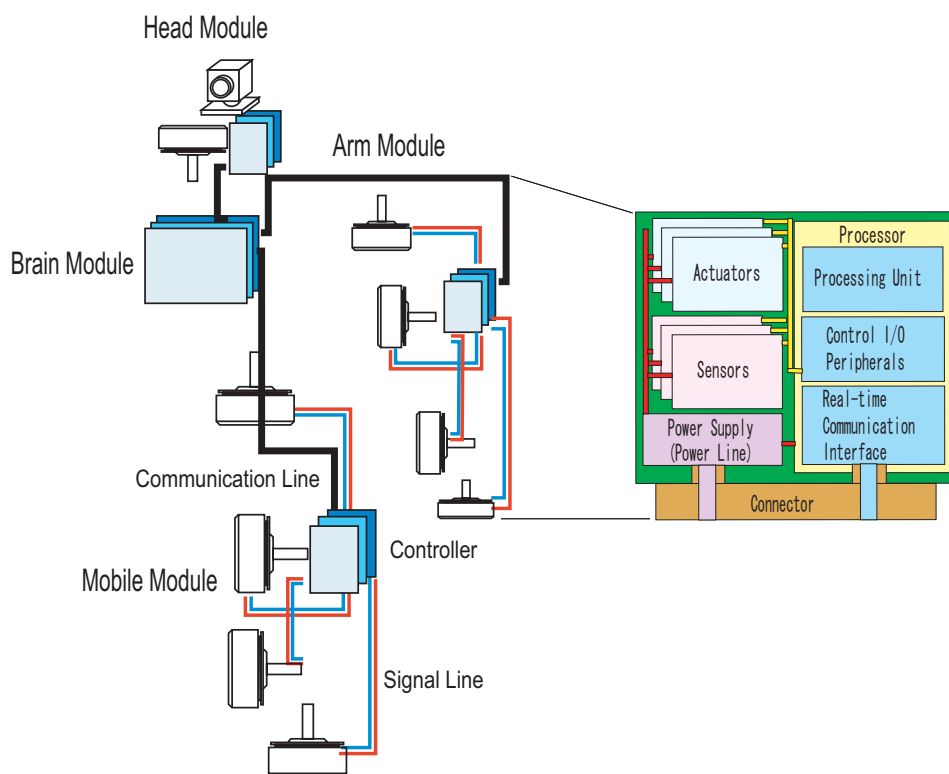


図 6.2: 機能別ロボットから構成するヒューマノイドロボット

6.1.3 再構成可能なモジュール型ヒューマノイドロボットの制御方式

本節では、従来のヒューマノイドロボットの制御アルゴリズムを再構成可能なモジュール型ヒューマノイドロボットへの適用する手法について検討する。

モジュール型ヒューマノイドロボットを構成する各機能別ロボットの制御は、従来のロボットの制御と同様に図 6.3 のように処理される。本システムアーキテクチャでは、各機能別ロボットが処理する制御のことを Local Control と定義する。

機能別ロボットの制御アルゴリズムとはその規模や情報量は異なるものの、ヒューマノイドロボットの制御アルゴリズムも図 6.3 の順番で処理される。再構成可能なモジュール型ヒューマノイドロボットの脳ロボットの制御では、図 6.3 のセンサ入力、アクチュエータ出力部が各機能別ロボットになる。また、再構成可能なモジュール型ヒューマノイドロボットの各機能別ロボットの制御では、図 6.3 のタスク入力が脳ロボットからのタスク指示、環境モデリングの結果が脳ロボットへの入力へと考えることができる(図 6.4)。本システムアーキテクチャでは、この脳ロボットと各機能別ロボットが構成する制御ループを Global Control と定義する。また、本システムアーキテクチャでは、Global Control による行動と並列な動作実行の実現、脳ロボットの負荷軽減、反射的行動、耐故障性の役割を

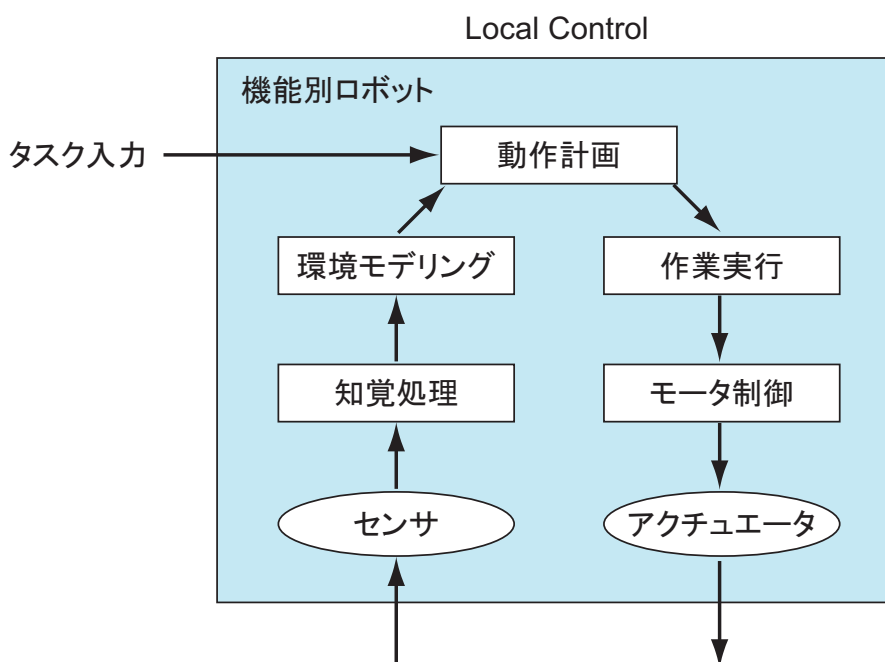


図 6.3: Local Control

果たすために，Cooprative Controlを導入することとする．

以上より，再構成可能なモジュール型ヒューマノイドロボットは，以下の3つの制御方式を導入することで従来のヒューマノイドロボットの制御アルゴリズムを実現する．

- Local Control

機能別ロボット内に閉じたマニピュレータ，歩行，画像処理のような要素技術を処理する局所的な制御のこと．本制御は，各機能別ロボットに割り当てるプロセッサ上で処理される

- Global Control

脳ロボットの行動計画によるロボットシステム全体を制御する大局的な制御のこと．各機能別ロボットからの入力イベントに対して脳ロボット内の行動計画部がシステム全体の行動計画を行い，各機能別ロボットへタスクを依頼することで，ロボット全体の制御を実現する

- Cooperative Control

複数のモジュールが脳ロボットの行動計画なしに自律的に協調してタスクを実行する制御機構．Cooperative ControlはGlobal Controlとは別の制御ループで形成され，Global Controlとは並列に動作する

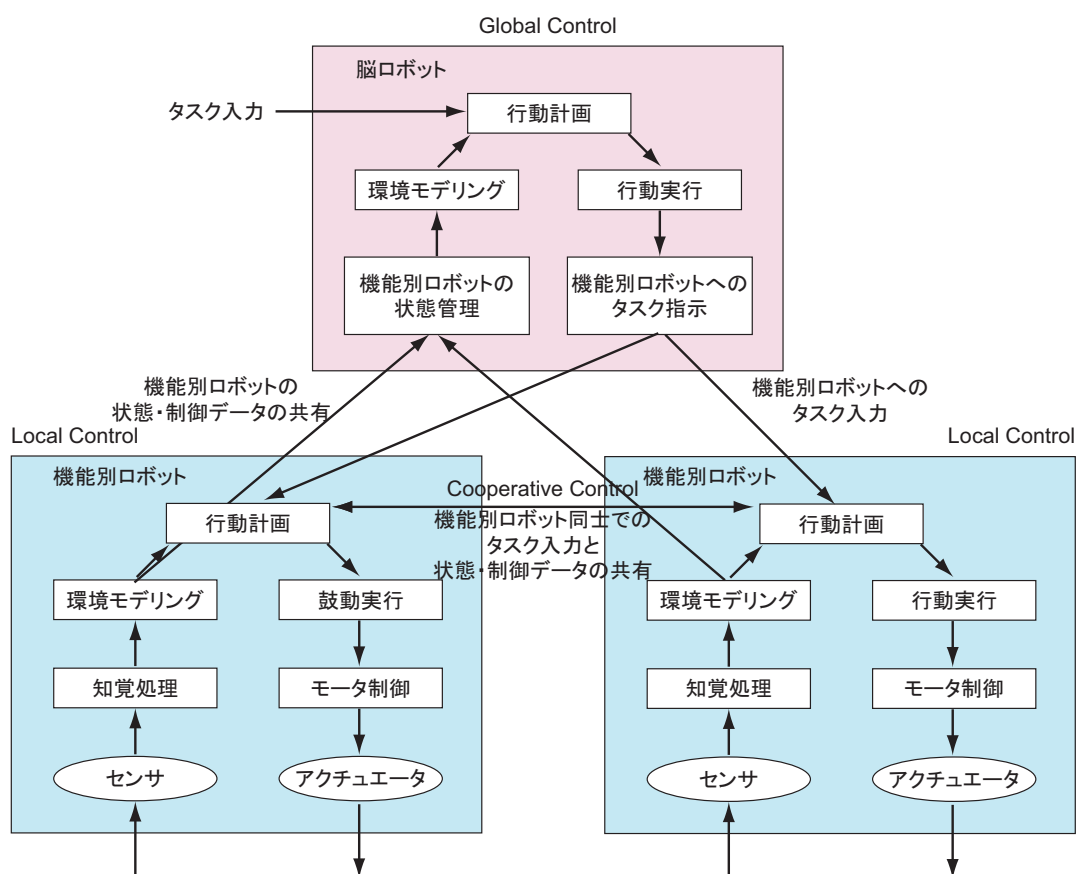


図 6.4: Global Control

図 6.4 から明らかなように，この制御方式は，設計方針通りに行動計画部の並列・分散処理化を行えている．

6.1.4 階層化制御ソフトウェアプラットフォーム

再構成可能なモジュール型ヒューマノイドロボットの制御ソフトウェアは，脳モジュールのシステム全体の行動計画ソフトウェア，各機能別ロボットの要素機能ソフトウェア，それらを接続するモジュール間通信機構（体内 LAN）から構成される．具体的には，再構成可能なモジュール型ヒューマノイドロボットのソフトウェアは，脳モジュールの管理の下，要素技術のソフトウェアがモジュール間通信機構で相互に接続することでロボット全体を制御する構成である．そこで，H7，ASIMO が採用しているようにモジュール型ヒューマノイドロボットのソフトウェアも階層化したソフトウェアプラットフォームを提唱することで，Global Control(脳ロボットの制御ソフトウェア)，Cooperative Control(機能別ロボットの協調制御ソフトウェア)，Local Control(機能別ロボットの制御ソフトウェア)を別々に

設計することができる。

本システムアーキテクチャでは、以下の要件に応じて、ロボットの制御ソフトウェアを階層化する。

1. ユーザやロボットを用いて実験を行う開発者には、モジュール間通信機構や機能別ロボットの詳細情報を熟知しなくとも、ヒューマノイドロボットの行動やタスク等のアプリケーションを記述できる環境が必要である
⇒ ヒューマノイドロボットのアプリケーションや行動を記述できる層の必要性
2. 複数の機能別ロボットを1つのヒューマノイドロボットとして扱うためには、システムの統合化が必要である
⇒ システム統合化ソフトウェアを開発できる層の必要性
3. 機能別ロボット同士を接続するための通信機能を提供する必要がある
⇒ モジュール間通信機構を開発できる層の必要性
4. ヒューマノイドロボット全体やモジュール間通信機構のことを意識せず、要素技術のソフトウェアを開発できる環境が必要である
⇒ 機能別ロボット(要素技術)のソフトウェアを開発できる層の必要性
5. 新たなI/O制御用ソフトウェアを導入できる必要がある
⇒ I/O制御など基礎的なソフトウェアを開発・導入できる層の必要性

以上の要件から、ロボットシステム全体のソフトウェア構成を、図 6.5 に示す 5 階層のアーキテクチャとして設計する。本アーキテクチャは、上位層からアプリケーション層、統合層、ネットワーク層、機能別ロボット層、機能制御層と定義する。ソフトウェアの特徴から、上位 2 層は脳モジュールに、下位 2 層は各機能別ロボットに分離する。

- アプリケーション層

モジュール型ヒューマノイドロボット全体の行動計画や管理を行う層

下位層である統合層が提供する機能 API を用いてヒューマノイドロボットの制御ソフトウェアを設計する層

- 統合層

機能別ロボットが提供する機能を統合化し、ヒューマノイドロボット制御用 API を上位層へ提供する層

- ネットワーク層

モジュール間通信機構を提供する層

モジュール間通信機構の詳細設計については後述する

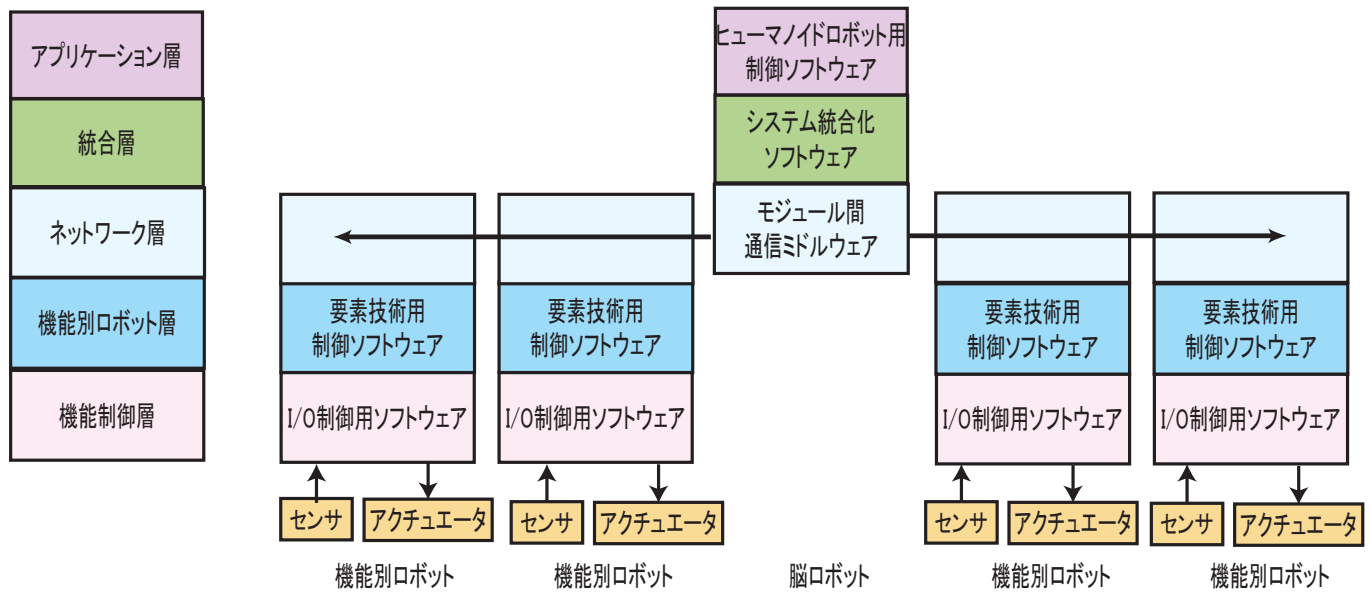


図 6.5: 5階層のソフトウェアアーキテクチャ

- 機能別ロボット層

頭ロボットや移動ロボットなどの機能・要素技術を研究・開発する層

拡張性を実現するために、各機能別ロボットはこの層において独自のOSやスケジューリング機構、アルゴリズムを使用できる

- 機能制御層

機能別ロボット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層
本層は、上位層からのタスクを基に、デバイスを直接制御し、その結果を上位層へ提供する

機能別ロボット用制御ソフトウェア

図 6.6 は、機能別ロボット単体での利用、開発する際のソフトウェア構成を示す。機能別ロボット単体は、アプリケーション層で機能別ロボット層が提供する機能 API を用いてロボットのアプリケーションを設計することができる。定義したアプリケーションは、モジュール間通信が必要ないので、ネットワーク層がそのまま機能別ロボット層へタスク入力し、下位層がそのタスクに従って処理を行うことになる。

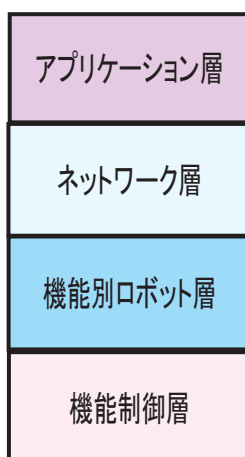


図 6.6: 単体ロボットのソフトウェア構成

複数の機能別ロボットから構成する協調動作制御ソフトウェア

本モジュール型ヒューマノイドロボットでは、左アームモジュールと右アームモジュールの協調動作や、アームモジュールと頭モジュールのビジュアルフィードバックなどの、Cooperative Control を開発するための環境が必要である。本提案のシステムアーキテクチャは脳モジュールがシステム内になくとも、モジュール情報を基にネットワークを形成することで、図 6.7 のようにして協調制御のソフトウェア開発を行うことができる。

このような複数の機能別ロボットが脳ロボットの動作計画なしに自律的に協調してタスクを実行する制御機構は、ロボットの反射的行動や、動作の並列度抽出、脳ロボットの負荷軽減、制御システムの耐故障性向上の面で大きな利点がある。複数の機能別ロボットが協調して制御する点も、本アーキテクチャの大きな特徴でもある。

6.1.5 モジュール間通信機構 (体内 LAN) の設計

再構成可能なモジュール型ヒューマノイドロボットのモジュール間通信機構 (体内 LAN) について検討する。まず始めに再構成可能なモジュール型ヒューマノイドロボットに適した通信インターフェースを決定する。続いて、通信プロトコルと通信ミドルウェアの設計を行う。

モジュール間通信機構の要件

モジュール型ヒューマノイドロボットを構成する機能別ロボット間の通信には、大きな問題が 3 つ存在する。

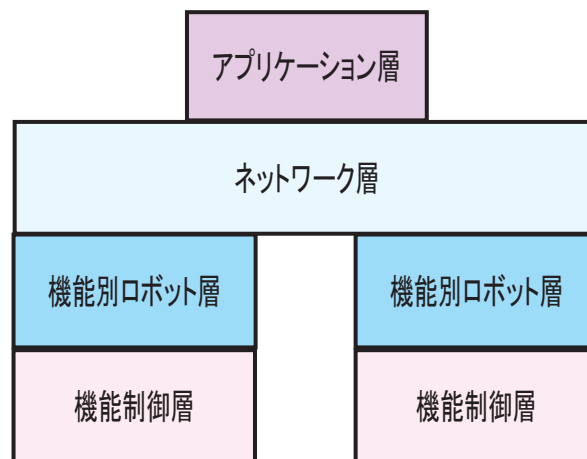


図 6.7: 協調制御のためのソフトウェア構成

1つ目は、機能別ロボットが処理する各機能間の同期・協調のリアルタイム性を保証しなければならない点である。情報処理分野の並列分散処理では、ある機能別ロボットが他の機能別ロボットのデータを参照する際に、予想以上の時間がかかってしまったとしても、システム全体のタイミングさえ合っていれば、安定した処理を行うことができる。しかし、制御分野では、モジュール間の通信路がリアルタイム性を有していないと安定で正確な制御を行うことができない。アクチュエータモジュールの制御ループの周期（例えば 1msec）以内に、センサモジュールからセンサデータを取得して、それを元にアクチュエータに対する次の指令値を計算し、アクチュエータの制御を行うモデルを例に挙げて考える。このモデルでは、以下の式を満たせば、分散制御可能であると判断することができる。 T_{sensor} はセンサモジュールがセンサデータを取得するのに必要な時間、 $T_{communication}$ は通信時間、 $T_{calculation}$ は次の指令値の計算時間、 $T_{actuator}$ はアクチュエータモジュールが処理を行うのに必要な時間、 T_{cycle} は時間制約（実行周期）を表す。

$$T_{sensor} + T_{communication} + T_{calculation} + T_{actuator} < T_{cycle} \quad (6.1)$$

ここで、 T_{sensor} 、 $T_{calculation}$ 、 $T_{actuator}$ の最悪実行時間を一定時間内にバウンドすることは容易である。しかし、 $T_{communication}$ はネットワークの状況によって変動するため、その最悪実行時間を一定時間内にバウンドすることは容易でない。従って、 $T_{communication}$ をバウンドすることが可能であるネットワーク、つまりリアルタイムネットワークの場合、そのモデルが分散制御可能かどうかを判定することができ、分散制御システムをトップダウンに設計することができる。一方、 $T_{communication}$ をバウンドできないネットワーク、つまり、非リアルタイムネットワークの場合は、まず機能別ロボット間を通信インターフェースで配線して実際に動作させ通信が間に合うかどうかをボトムアップに調べる必要があり、

分散制御アルゴリズムをトップダウンに設計・実装することは困難である．以上より，再構成可能なモジュール型ヒューマノイドロボットには， $T_{communication}$ をバウンドすることができるリアルタイムネットワークが必要不可欠である．

このようなリアルタイム通信は，大きく以下の2つに分類することができる．

- ソフトリアルタイム通信
- ハードリアルタイム通信

ソフトリアルタイム通信は，時間粒度が比較的大きく，バンド幅保証をする通信であり，主として大量のセンサデータ等の通信に用いられる．それに対し，ハードリアルタイム通信は，時間粒度が小さく，低レイテンシを保証する通信であり，主として制御コマンドの通信に用いられる．機能別ロボットの並列動作と統合化を実現するためには，これら2つのリアルタイム通信を保証する必要がある．

2つ目は，ネットワークの柔軟性の点である．研究・開発が行われている体内 LAN の多くは，そのシステム毎に独自の仕様で作り込まれている場合が多い．また，体内 LAN に利用されている多くの通信インタフェースには，構成できるネットワークのトポロジに制約がある．その結果，ロボットシステム上に制御アルゴリズムを，トップダウンに設計・実装することが困難となってしまう恐れがあり，従来のヒューマノイドロボットが抱えている拡張性の問題点を解決することができない．従って，体内 LAN には，ブロックを組み立てるようにロボットの機能を自由に変更した場合でも，ロボット機能のリアルタイム性を理論的に保証できるような柔軟性を有する必要がある．

3つ目は，消費電力・耐ノイズ性の点である．一般的に通信速度(動作周波数)を速くすれば消費電力が大きくなり，遅くすれば小さくなる．また，ロボット内はノイズ源(例えば，大型 AC モータやバッテリーの近傍)が多く，通信速度を速くすると伝送路にノイズが載り過ぎて，エラー訂正が間に合わない状況も考えられる．従って，ロボットの体内 LAN を構築するには，消費電力・耐ノイズ性についても検討する必要がある．

関連研究

本節では，現在までに開発された体内 LAN 機構の特徴を先の3点に着目して利点と欠点を明らかにし，再構成可能なモジュール型ヒューマノイドロボット用の体内 LAN 機構の設計に反映する．

現在までに開発された体内 LAN 機構には，Ethernet [Akachi *et al.* 05]，USB [USB]，IEEE1394 [Association]，レスポンスブリンク [山崎 04]，I²C [小屋迫光太郎 他 96]，VME バス [HONDA]，CAN バス [Kaiser *et al.* 99]，OPEN-R バス [Fujita *et al.* 99] 等，様々なインタフェースが利用されてきた．

安価で高い通信速度を持ち、最も汎用的である Ethernet は、CSMA/CD 方式によりネットワークにアクセスするため、リアルタイム通信を実現することは一般的に難しいとされている。近年では、Store-and-Forward 方式、全二重通信を採用した Ethernet スイッチ、Ethernet スイッチ内部のパケット送信のタイミングをスケジューリングアルゴリズムを利用してを管理する手法 [Hoang *et al.* 02],[Chen *et al.* 02]、リアルタイム OS の機能を利用して、アプリケーション側で通信の送信タイミングを制御する手法 [佐藤秀雄 他 01]、[石綿陽一 他 04] 等、Ethernet の問題解決に取り組む研究が様々行われている。また、OSI 基本参照モデルの物理層とデータリンク層の一部を規定し、リアルタイム性を保証している ARCNET [ARC] などもある。

IEEE1394 と USB のアイソクロナス転送は、 $125\mu\text{sec}$ と時間粒度の小さいリアルタイム通信を実現することが可能であり、ロボット制御用にも十分である。その反面、IEEE1394 を利用したネットワークは、最大通信可能なプロセッサ数が 63 個である、サイクルマスタが必要である、ループトポロジが不可能である等の制約がある。USB を利用したネットワークも最大通信可能なプロセッサ数が 127 個である、必ずルートコントローラが必要である、トポロジがツリー構造のみである等の制約がある。ロボットの制御に関するデータ転送に誤りが生じた場合、そのデータは制御系へ多大な影響を与えることになる。ハードウェアでエラー訂正を行わないアイソクロナス転送を制御用途に使用するためには、ソフトウェアでエラー訂正を行わなければならない欠点がある。

レスポンスリンクは、ソフトリアルタイム通信(データリンク)とハードリアルタイム通信(イベントリンク)の分離し、優先度付きパケットを用いることで、ハードリアルタイム通信とソフトリアルタイム通信の両立を実現している。また、レスポンスリンクは、トポロジフリーであるためネットワークの制約はなく、ハードウェアによる前方エラー訂正機能も備えている。

CAN(Controller Area Network) は、自動車や FA 等に用いられているリアルタイム通信規格である。CAN ではトークンによるネットワークへのアクセス制御により、データの衝突を回避し、通信レイテンシの予測を行うことが可能である。しかしながら、CAN では、プロセッサの数が増加した場合、各プロセッサがネットワークにアクセス可能となるまでのオーバヘッドが長くなる、通信の高速性を満たさない等の欠点がある。

以上のことから、再構成可能なモジュール型ヒューマノイドロボットのモジュール間通信に利用できそうな通信インタフェースの転送速度、リアルタイム性、トポロジに関する特徴を表 6.1 にまとめた。

CAN の後継として期待されている FlexRay [FlexRay Consortium] は実用段階ではないため、TITechWire [清水正晴 他 01]、OPEN-R バス [藤田雅博 他 01] 等の独自規格の通信インタフェースは標準化されていないため、モジュール間通信機構の通信インタフェース

表 6.1: 通信インタフェースの比較

	転送速度	リアルタイム性	トポロジ	採用例
CAN [Kaiser <i>et al.</i> 99]	1 Mbps		バス	HERMES
ARCNET [ARC]	10 Mbps		バス, スター, ツリー	ASIMO
レスポンスリンク [山崎 04]	67 Mbps		フリー	-
CompactPCI [Group]	133 Mbps	×	バス	H7
IEEE1394 [Association]	400 Mbps		スター, ツリー, ディ ジーチェーン	SmartPal
USB [USB]	480 Mbps		ツリー	Fujitsu
Ethernet	1000 Mbps	×	フリー	EMIEW , HRP

の候補の対象外とした。この表の中で、実際に利用できる通信インタフェースを選択するために、まず、再構成可能なモジュール型ヒューマノイドロボットが必要とする転送速度を見積もることとした。

- 仮定条件

- 40 自由度，各データは 16 ビットで表現される
- 各通信は 1msec 以内に完了する
- モジュール数は 20 個 (頭部 (聴覚, 視覚), 脳部, 右/左腕部, 右/左手部, 右/左脚部といった 9 個の機能別ロボットからシステムが構成されると想定しているが, 拡張性のことを考慮して, ここではその約 2 倍の 20 個としている)

- 脳ロボットから機能別ロボットへの通信

- 動作指令, 位置指令, トルク指令
- $16 \times (20 + 20 + 20) / 0.001 = 960,000$ bps

- 機能別ロボットから脳ロボットへの通信

- 関節角度・トルク・温度, 角速度センサ (3 個), 加速度センサ (3 個), 6 軸力センサ (3 個)

表 6.2: レスポンシブリンクと IEEE1394 の比較

	レスポンシブリンク	IEEE1394
転送速度	6.25 ~ 67 Mbps(動的に変更可能)	100 ~ 400 Mbps(動的に変更可能)
リアルタイム性	イベントとデータの分離, パケットの追い越し (パケットの優先度)	アイソクロナス転送 (アシンクロナス転送 (非リアルタイム通信) との時分割通信)
結合方法	Point-to-Point	バス結合
トポロジ	フリー	スター, ツリー, デイジーチェーン
ノード数	256	63
エラー訂正	あり	なし
消費電力	0.2 W	0.3 W
通信プロトコル	ハードウェアによる転送, ソフトウェアはルーティング設定のみ	ハードウェアによる転送, ソフトウェアは資源割り当てのみ

$$- 16 \times (40 + 40 + 40 + 3 + 3 + 6 \times 3) / 0.001 = 2,304,000 \text{ bps}$$

- 機能別ロボットから他の機能別ロボットへの通信

- 関節角度・トルク・温度, 角速度センサ (3 個), 加速度センサ (3 個), 6 軸力センサ (3 個)

$$- 16 \times (40 + 40 + 40 + 3 + 3 + 6 \times 3) / 0.001 \times 19 = 43,774,000 \text{ bps}$$

以上より, 表 6.1 の中から 40Mbps 以上の転送速度を持つレスポンシブリンク, Compact-PCI, IEEE1394, USB, Ethernet が, モジュール間通信機構の候補として残る. さらに, これら 5 つの通信インタフェースをリアルタイム性, トポロジの 2 点で比較すると, レスポンシブリンクと IEEE1394 の 2 つが候補として残る. 表 6.3 は, これら 2 つの通信インタフェースの特徴を示している. しかし, レスポンシブリンクではトポロジ, ノード数, エラー訂正, IEEE1394 では通信速度がお互いに優れているため, この比較だけではどちらの通信インタフェースがモジュール間通信機構に適しているのか判断するのが困難である. そこで, モジュール間通信機構に利用する通信インタフェースは, 実際にレスポンシブリンクを利用したネットワーク, IEEE1394 を利用したネットワークを構築し, その性能を, リアルタイム通信, 拡張性, 消費電力・耐ノイズ性の 3 点で比較することで決定することとした.

レスポンスリンクと IEEE1394 の通信評価

本節では，レスポンスリンクと IEEE1394 のリアルタイム性と拡張性について評価を行う．リアルタイム性は，レスポンスリンクと IEEE1394 をそれぞれ利用したネットワークを用いて，通信のレイテンシとジッタを評価する．拡張性は，レスポンスリンクと IEEE1394 がともにトポロジ制約に関してはほとんど差がないため，ネットワーク上を過負荷にした状態でも，リアルタイム性を維持できるのかを評価する．

本研究が提案する再構成可能なモジュール型ヒューマノイドロボットが，頭部(聴覚，視覚)，脳部，右/左腕部，右/左手部，右/左脚部といった9個の機能別ロボットからシステムが構成されると想定している．そこで図 6.8 に示すようにレスポンスリンクで接続されたネットワーク(以降，レスポンスリンクネットワークと呼ぶ)と，図 6.8 と等価である図 6.9 に示す IEEE1394 で接続されたネットワーク(以降，IEEE1394 ネットワークと呼ぶ)を，評価用のネットワークとして利用する．図 6.8 と図 6.9 中の C1 は機能別ロボット 1 を示す．

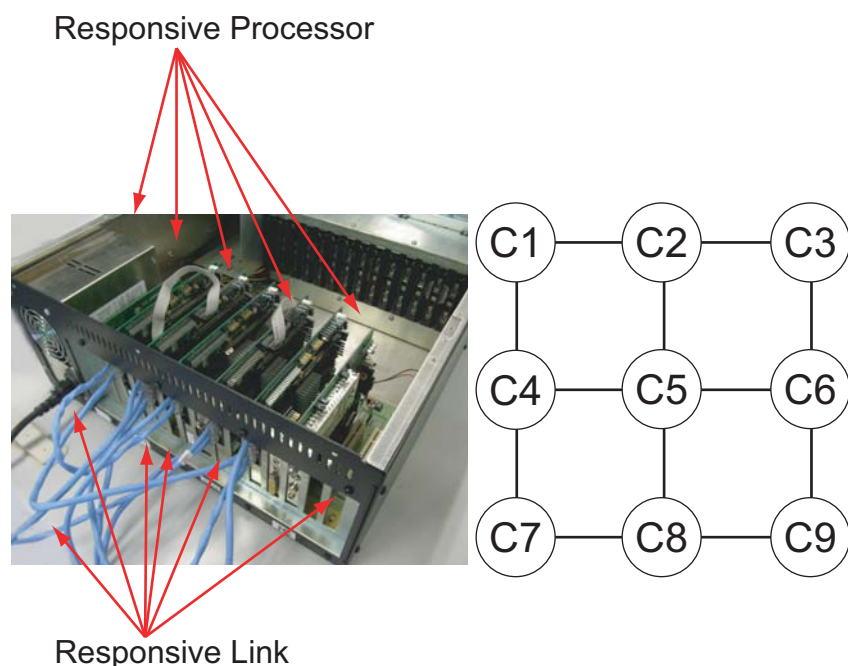


図 6.8: RL-Net の設計・評価環境

表 6.3 は，レスポンスリンクネットワークと IEEE1394 ネットワークの構成，通信遅延時間の測定方法を示す．以降では，これらの環境を利用して，レスポンスリンクと IEEE1394 の評価を行う．

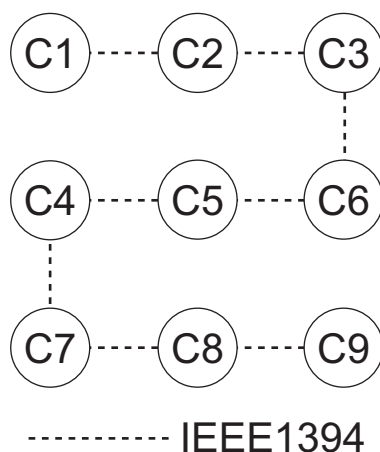


図 6.9: IEEE1394 の実験環境

表 6.3: レスポンシブリンクと IEEE1394 の比較

	レスポンシブリンク	IEEE1394
プロセッサ	レスポンシブプロセッサ	Pentium 4
動作周波数	100 MHz	2.4 GHz
メモリ	SDRAM 16M バイト	DDR SDRAM 512M バイト
リアルタイム OS	RT-Frontier [Kobayashi <i>et al.</i> 04]	RT-Linux Free
測定方法	解像度 0.05 μ sec の内部タイマ とロジックアナライザ	Linux が提供する gettimeofday 関数と パケットアナライザ
備考	67Mbps の設定で通信	400Mbps の設定で通信 . TSB43AB22A を 1394 コントローラとして利用

通信プロトコルに関する評価

通信プロトコルの性能の評価は、レスポンシブリンクのイベントリンク (ペイロードサイズ:16 バイト)、データリンク (ペイロードサイズ:56 バイト)、IEEE1394 のアイソクロナス転送 (ペイロード:56 バイト) のパケットの送受信にかかるソフトウェアオーバーヘッドを測定することで行った。表 6.4 はその結果を示す。レスポンシブリンクネットワークでの受信時間は、受信割り込み発生から受信先のバッファにパケットデータのコピーが終わるまでの時間、送信時間は、パケットデータを SDRAM にデータを作成後から書き終わるまでの時間とした。IEEE1394 ネットワークでの送信時間は、パケットを生成しバスに値

表 6.4: 通信プロトコルのソフトウェアオーバーヘッド

	平均値 (μsec)	最悪値 (μsec)	標準偏差
イベント受信	23	26	0.31
イベント送信	13	15	0.42
データ受信	68	71	0.79
データ送信	36	38	0.91
IEEE1394 受信	17	21	0.65
IEEE1394 送信	13	16	0.61

を書き込むまでの時間、受信時間は、バスに値が書き込まれてから取得するまでの時間とした。

レスポンスリンクネットワークと IEEE1394 ネットワーク通信の性能は、マニピュレータシステムに利用されている Ethernet を用いたリアルタイム通信機構である RT-Messenger [佐藤秀雄 他 01] の通信性能 (パケットサイズ:48 バイト, 最悪受信時間:35 μsec , 最悪送信時間:8 μsec), HRP-3P の体内 LAN に利用されている Ethernet を用いたリアルタイム通信機構である実時間 Ethernet [石綿陽一 他 04] の通信性能 (パケットサイズ:46 バイト, 最悪受信時間:120 μsec , 最悪送信時間:20 μsec) と比較しても同等の性能であることがわかる。また, これらの通信は高性能プロセッサで処理している事を考えると, 組込みプロセッサであるレスポンスプロセッサで処理するレスポンスリンクネットワークの通信プロトコルの性能は, 十分な性能であると考えられる。

また, レスポンスリンクネットワークの通信性能は IEEE1394 ネットワークと比較して, 標準偏差が小さく, 送受信処理にかかるオーバーヘッドにバラツキの少ない, 安定した送受信機構を実現している。

通信レイテンシに関する性能評価

本節では, 通信レイテンシに関する性能を示す。

レスポンスリンクネットワークの通信レイテンシに関する評価は, 図 6.8 の C1 から C4 を経由して C5 へレスポンスリンクのイベントリンク, データリンクの 1 パケットを 1msec の周期で送信する際のレイテンシを計測することで行った。計測は, (1) ネットワークが無負荷状態, (2) C1 と C5 以外の全コントローラが, C4 を経由して C5 へ, C1 から C5 へのパケットより低優先度の外乱パケットをベストエフォートで送信している最悪のネットワーク状態の 2 通りの状態で各々 500 回の通信を行った。実験結果は図 6.10, 図 6.11 であり, 横軸は通信回数, 縦軸は通信レイテンシを示す。

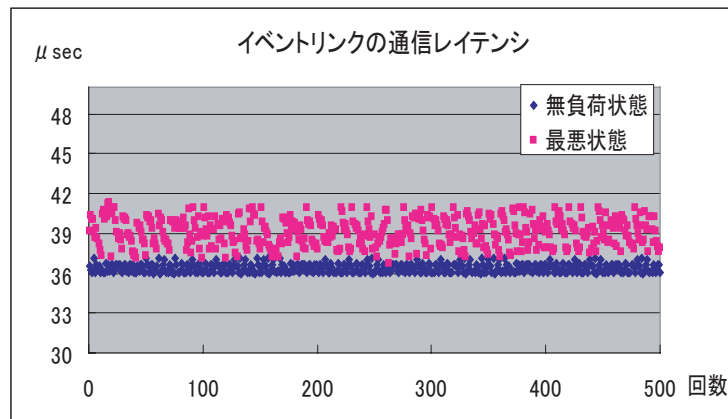


図 6.10: レスポンシブリンクネットワークの実験結果 (イベントリンク)

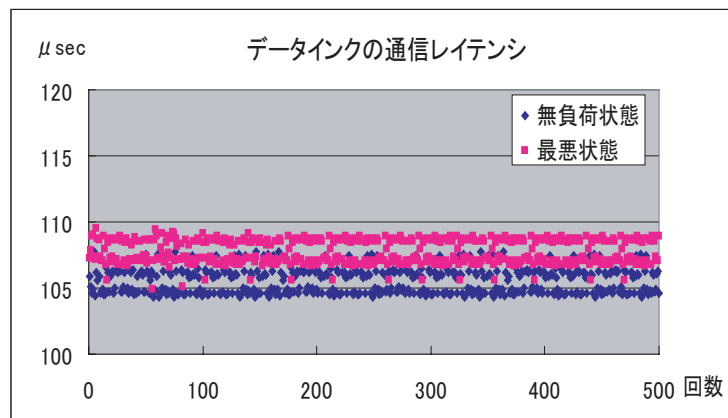


図 6.11: レスポンシブリンクネットワークの実験結果 (データリンク)

図 6.10, 図 6.11 より, 外乱パケットがレスポンシブリンクネットワーク上に存在した場合でも, 無負荷状態と同等の低レイテンシを実現している事がわかる. 外乱パケットの送信開始後平均して数マイクロほどレイテンシは増大しているが, このレイテンシはソフトウェアでの受信処理において影響を受けてしまった結果だと考えられる. この測定結果は, レスポンシブリンクネットワークはネットワークが負荷に対してもリアルタイム性能を維持でき, ネットワーク拡張に対して柔軟性があることがわかった.

IEEE1394 ネットワークの通信レイテンシに関する評価は, 図 6.9 の C1 から C5 へ RL のデータリンクの 1 パケット分の最大ペイロードサイズに相当する 56 バイトのデータを 1msec の周期でアイソクロナス転送する際に生じる通信レイテンシを計測することで行った. 計測は, レスポンシブリンクネットワークでの計測と同様で, (1) ネットワークが無負荷状態, (2) C1 がアイソクロナス転送 (4040 バイト = 最大転送ペイロードサイズ: 4096 バイ

ト - 56 バイト) とアシンクロナス転送 (2048 バイト) を $125\mu\text{sec}$ 周期で外乱パケットを C5 へ送信している最悪のネットワーク状態 (最悪状態) の 2 通りの状態で各々 500 回の通信を行った。実験結果は図 6.12 であり、横軸は通信回数、縦軸は通信レイテンシを示す。

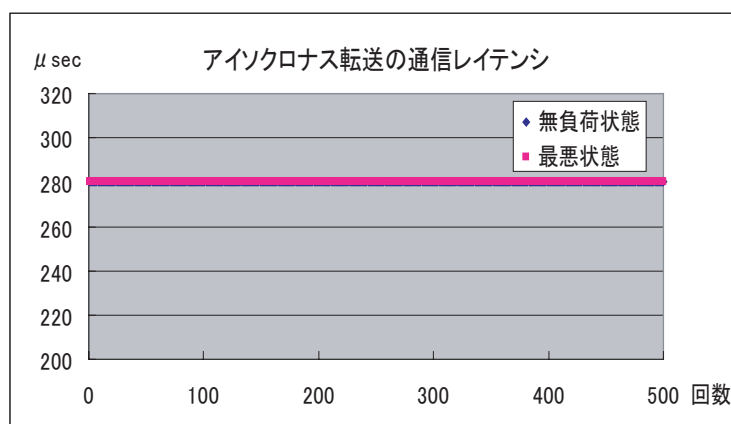


図 6.12: IEEE1394 ネットワークの実験結果

図 6.12 より、IEEE1394 ネットワークの通信レイテンシはレスポンスリンクネットワークと比較して大きいことがわかる。これは、OHCI の規格上、DMA 転送後に通信パケットがバスに書き込まれるまでに 2 サイクル ($125\mu\text{sec} \times 2 = 250\mu\text{sec}$) のレイテンシを必要とするためである。IEEE1394 ネットワークでは、通信レイテンシは大きいものの、外乱パケットによる影響をうけることなく、ほぼ一定のレイテンシを保証しており、レスポンスリンクネットワークと比較しても安定した通信レイテンシ保証を実現していると判断できる。

以上の結果より、最悪通信レイテンシの観点から検討すると、レスポンスリンクネットワークと IEEE1394 ネットワークは、ネットワークの負荷に関係せず最悪通信レイテンシを低レイテンシで保証でき、制御用途として十分利用できること、ネットワークの拡張に対しても柔軟性があることを検証できた。

通信ジッタに関する性能評価

本節では、通信ジッタに関する性能を示す。先の通信レイテンシの評価実験では、送信側は 1msec の周期でデータを送信していたので、受信側でデータ到着間隔 (ジッタ) を測定した。表 6.5 と表 6.6 は、レスポンスリンクネットワーク、IEEE1394 ネットワークの通信で生じた通信ジッタを標準偏差と peak-to-peak により評価した結果をそれぞれ示している。

レスポンスリンクネットワークでは、パケットに付与する優先度に応じて、追い越しをする特徴から、ネットワーク負荷に関係せず、低ジッタを実現している。 1msec の制御周期で考えると、受信側の揺れ幅は 0.7% 以内に収まっており、制御用途にも十分利用でき

表 6.5: 標準偏差による比較

	無負荷状態	最悪状態
レスポンスリンクネットワーク (イベントリンク)	1.21	1.32
レスポンスリンクネットワーク (データリンク)	1.25	1.30
IEEE1394 ネットワーク (アイソクロナス転送)	0.10	27.1

表 6.6: peak-to-peak による比較

	無負荷状態	最悪状態
レスポンスリンクネットワーク (イベントリンク)	6.3	6.5
レスポンスリンクネットワーク (データリンク)	6.7	6.6
IEEE1394 ネットワーク (アイソクロナス転送)	0.0	76.2

ることがわかる。また、図 6.24 のデータ転送とイベント転送は独立して低ジッタを保証している。

IEEE1394 ネットワークのアイソクロナス転送にジッタが生じている。つまり、アイソクロナス転送のスタートを表すサイクルスタートにジッタが生じていることになる。IEEE1394 では、通常、サイクルスタートは $125\mu\text{sec} \pm 12.5\text{nsec}$ に設定されており、ジッタは $\pm 12.5\text{nsec}$ になる。しかし、サイクルスタート送信時にデータ転送が行われている場合、サイクルスタートは延期され、アイソクロナスデータ転送にジッタが発生する場合が生じる。これは、 $125\mu\text{sec}$ のアイソクロナス間隔の終わり近くでアシンクロナス転送要求が発行され、サイクルマスタが次のアイソクロナストランザクションを $125\mu\text{sec}$ の周期で起動できなくなることで発生したと考えられる。サイクルマスタにジッタが生じた際、IEEE1394 では、次のサイクルでアイソクロナス転送に優先的にバンド幅を提供し、アシンクロナストランザクションの実行を抑制することで、サイクル時間を短くし、ジッタの影響を吸収する。以上より、アシンクロナス転送はもとより、アイソクロナス転送も厳密なリアルタイム性が保証されていないことがわかる。IEEE1394 ネットワークをモジュール間通信機構に利用する場合は、アシンクロナス転送の影響によりアイソクロナス転送にジッタが生じる恐れがあるので、アシンクロナス転送を利用しない、アシンクロナストランザクションのタイミングを作り込む、転送するパケットのサイズを小さくするなどの対応が必須となる。仮に、ロボットのモジュール間通信機構にアシンクロナス転送を利用しない場合、最大でも 80% のバンド幅しか利用できなくなるため非効率である。以上の結果から、IEEE1394 のアイソクロナス転送、アシンクロナス転送のように共有された同一の通信線を通して時分割に通信を行う方式では、通信時間が正確にバウンドできないので、リアルタイムシステ

表 6.7: 消費電力に関する評価

インタフェース	1 ノード	9 ノード
レスポンスリンク	0.2	1.8
IEEE1394	0.3	2.7

表 6.8: レスポンスリンクと IEEE1394 の比較結果

	リアルタイム通信	拡張性	消費電力・耐ノイズ性
レスポンスリンク			
IEEE1394		×	

ムは実現困難であると考えられる。

以上より，レスポンスリンクネットワークは，ネットワークの負荷の影響をほとんど受けることなく，低ジッタを保証でき，制御用途として十分利用できることを示せた。

消費電力・耐ノイズ性に関する検討

表 6.7 は，利用した評価環境のうち通信機器が必要とする消費電力を示す。

表 6.4 と表 6.7 は，IEEE1394 ネットワークの方がレスポンスリンクネットワークよりも低通信レイテンシや高バンド幅であるが，消費電力も増大してしまうというトレードオフを示している。また，IEEE1394 ネットワークのアイソクロナス転送はエラー訂正をしないため，ノイズ源の近くでは利用しにくい欠点がある。

以上より，レスポンスリンクは，低消費電力，ハードウェアでエラー訂正を行っている点で，IEEE1394 よりもロボットの間通信機構に適しているといえる。

まとめ

表 6.8 は，リアルタイム通信，拡張性，消費電力・耐ノイズ性の 3 点で比較した結果を示している。今までの検討の結果，レスポンスリンクネットワークの方が IEEE1394 ネットワークよりもモジュール間通信インタフェースに適していると判断することができた。従って，本研究は，これらのレスポンスリンクの特徴を利用したモジュール間通信機構を設計する。レスポンスリンクを利用するネットワークは，下記のユニークな機能によって，拡張性・柔軟性の高いリアルタイム通信を実現できる。レスポンスリンクの詳細は [山崎 04] を参照していただきたい。

- 通信パケットに優先度を付け、高い優先度の通信パケットが低い優先度の通信パケットを通信ノード毎に追い越し、
- イベント通信 (イベントリンク) とデータ通信 (データリンク) の分離、
- 同じネットワークアドレス (送信元アドレス及び送信先アドレス) を持つ通信パケットの経路を優先度別に設定することによって専用回線や迂回路を設けリアルタイム通信を制御、
- 通信パケットの優先度を通信ノード毎に付け替え可能にすることによってパケットの加減速を分散管理で制御、
- ハードウェアによるフレーム単位のエラー訂正、
- 通信速度を動的に変更可能、
- トポロジフリー、
- Plug-and-Play

機能別ロボットの電気・電子ハードウェアとソフトウェア

前節で、再構成可能なモジュール型ヒューマノイドロボットのモジュール間通信機構にはレスポンスリンクを利用することを決定した。レスポンスリンクは、現在、情報処理学会試行標準 WG6 において国内の標準化が行われ、ISO/IEC JTC1 SC25 WG4 において国際標準化作業を行っている。その結果、レスポンスリンクは今後広く用いられる可能性が高い通信規格である。しかし、レスポンスリンクは現在開発段階であり、レスポンスリンクをサポートするプロセッサがレスポンスプロセッサ [Yamasaki 01](図 6.13) と現在開発中のロボット制御用プロセッサである Responsive Multithreaded Processor [Yamasaki 05](図 6.14) の 2 種類だけである。従って、現段階では、各機能別ロボットのプロセッサにはレスポンスプロセッサを採用することとした。

レスポンスプロセッサは主に以下のような特徴がある。詳細は [Yamasaki 01] を参照していただきたい。

- SPARC lite プロセッサコア 100MHz
- レスポンスリンク (5ch)
- コンピュータ用周辺

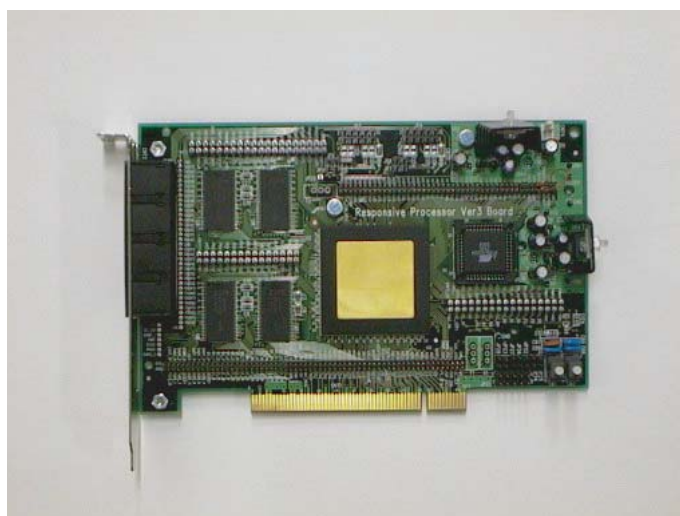


図 6.13: レスポンシブプロセッサ

- SDRAM I/F , DMAC , MMU , IRC , PCI I/F , USB , タイマ/カウンタ , SIO (同期 1ch , 非同期 2ch) , PIO(16 ビット) , RTC
- 制御用周辺
 - A/D コンバータ (8ch) , D/A コンバータ (2ch) , PWM(9ch) , パルスカウンタ (8ch)
- パワーマネージメントユニット
 - ソフトウェアでクロックを動的に変化することで , 消費電力を制御可能

レスポンシブプロセッサ上で動作し , 各機能別ロボットの制御ソフトウェアを管理するリアルタイム OS には , RT-Frontier を採用した (図 6.15) .

RT-Frontier は主に以下のような特徴がある . 詳細は [Kobayashi *et al.* 04] を参照していただきたい .

- 1msec を単位時間としたスケジューリング
- タスクの実行時間は $100\mu\text{sec}$ を単位として管理することが可能

通常 , ロボットの制御周期は固定である . そこで , 本システムアーキテクチャでは , 固定優先度スケジューリングで最適である Rate Monotonic(RM) スケジューリング [Lehiczky *et al.* 89] を RT-Frontier のスケジューラとして設計・実装した . RM スケジューリングは , 周期ス



図 6.14: Responsive Multithreaded Processor

レッドに対して、実行周期の短いスレッドに高い優先度を与え、優先的に処理を行うスケジューリング方法である。

図 6.16 に RM でスケジューリングされたスレッドの実行例を示す。図の矢印はスレッドの起動時刻を表している。それぞれのスレッドは周期的に起動する。例では Thread 0 の実行周期が最も短かく、Thread 3 の実行周期が最も長い。RM スケジューリングでは周期の短いスレッドに高い優先度を与えるため、Thread 0 に最も高い優先度が与えられ、Thread 3 に最も低い優先度が与えられる。

最初に全てのスレッドが起動し実行可能になるが、優先度に従ってスレッドを実行するため、最も高い優先度が与えられている Thread 0 が最初に実行される。Thread 0 の実行が完了すると、実行可能なスレッドの中で次に優先度の高い Thread 1、Thread 2 と順番に実行していく。

例では Thread 2 の実行中に Thread 0 が起動し実行可能になる。RM スケジューリングはスレッドのプリエンプション(横取り)を前提としている。プリエンプションでは、あるスレッドの実行中により優先度の高いスレッドが実行可能になると、優先度の高いスレッドの実行を優先するために実行するスレッドを切り替える。そのため、スケジューラは、Thread 2 の実行を中断し、Thread 0 に切り替える。そして Thread 0 の実行を開始する。Thread 0 の実行が完了すると、実行可能なスレッドの中で最も優先度の高い Thread 2 の実行を再開する。Thread 2 の実行中に Thread 1 が起動し実行可能になるため、Thread 2 の実行を再び中断し、実行するスレッドを Thread 1 に切り替えて、Thread 1 を先に実行する。Thread 1 の実行が完了すると Thread 2 の実行を再開する。Thread 2 の実行が完了すると、Thread 3 を実行する。

RM スケジューリングにおいて、スケジューリングが可能であるためのプロセッサ利用

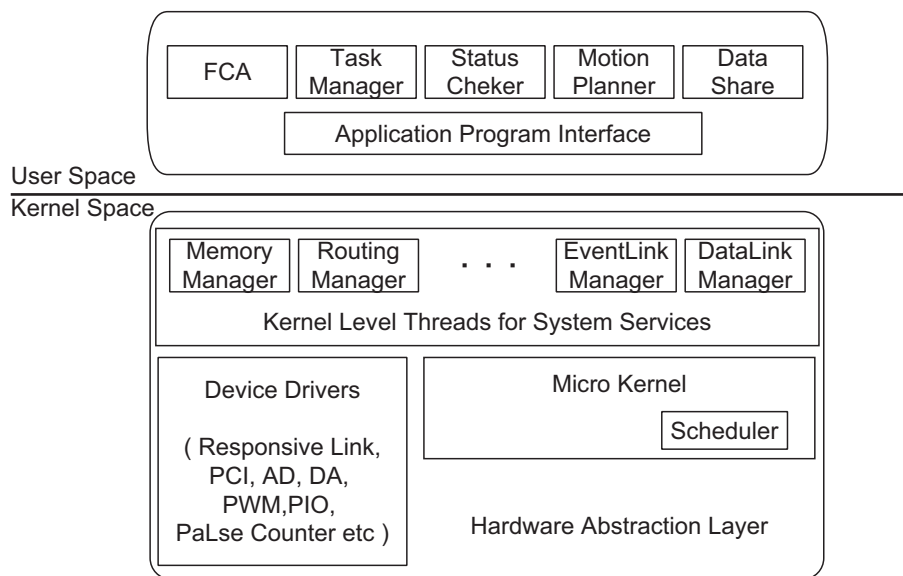


図 6.15: リアルタイム OS:RT-Frontier

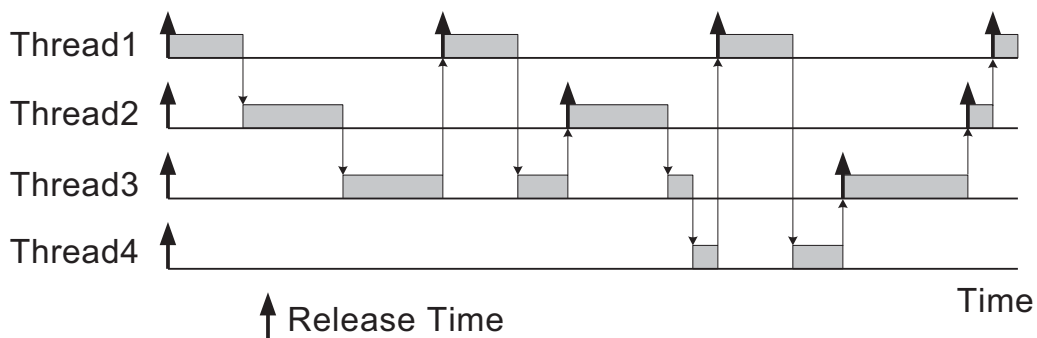


図 6.16: RM スケジューリングの例

率の最小の上限値 U_{lub} は、スレッド数 n に対して $U_{lub} = n(2^{\frac{1}{n}} - 1)$ である。 n が増加すると最終的に $U_{lub} = \ln 2 \cong 0.69$ に収束する。ただし、これは十分条件であるため、実際にスケジューリングが可能であるかどうか判定するためには、Response Time Analysis(RTA)を用いる。RTA では、まずスレッド τ_i の最悪反応時間 R_i を以下の式により、 $R_i^{(k)} = R_i^{(k-1)}$ となるまで反復して計算する。

$$R_i^{(0)} = C_i$$

$$R_i^{(k)} = C_i + \sum_{j:D_j < D_i} \left\lceil \frac{R_i^{(k-1)}}{T_i} \right\rceil C_j$$

求められた全てのスレッドの最悪反応時間がデッドラインよりも短い場合、つまり、

$$\forall \tau_i, R_i < D_i$$

を満たす場合にスケジューリング可能であると判断できる。

各機能別ロボットの制御ソフトウェアは、その特徴に応じた周期を設定し、RTA にてスケジューリング可能であるか判断することとする。

6.1.6 レスポンシブリンクを利用したモジュール間通信機構 RL-Net

本節では、ロボット制御アルゴリズムの並列・分散処理化に必要なリアルタイム性、拡張性、低消費電力・耐ノイズ性の3点を備えたモジュール間通信機構 RL-Net の設計について述べる。

図 6.17 に示す RL-Net は、図 6.5 に示す統合層と機能別ロボット層を結合するネットワーク層に相当する。ネットワーク層である RL-Net は、機能別ロボットと脳ロボット間の協調動作、データの共有を実現する上で重要な層である。

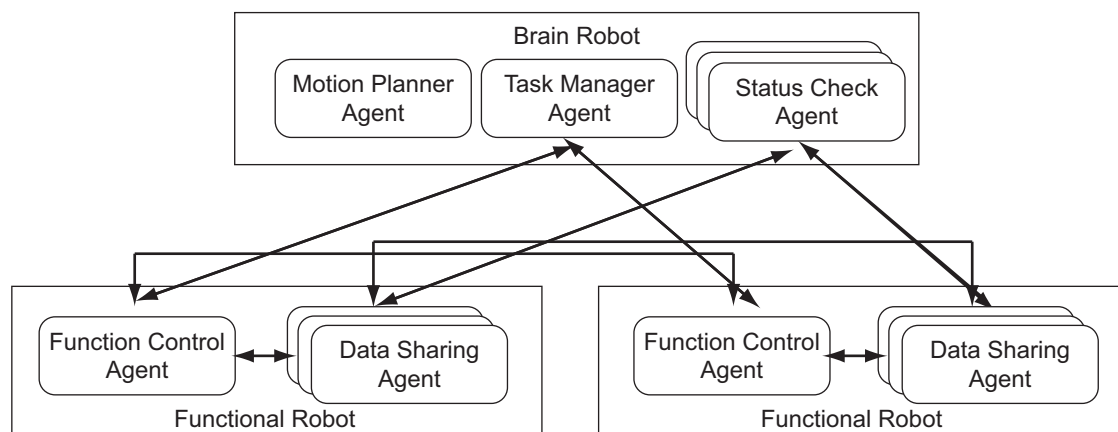


図 6.17: モジュール間通信機構:RL-Net

RL-Net は、統合層に実装するロボット全体の行動計画を行う Motion Planner Agent(以下、MPA と呼ぶ)、各機能別ロボットへタスク指示を行う Task Manager Agent(以下、TMA と呼ぶ)、各機能別ロボットの状態を管理する Status Check Agent(以下、SCA と呼ぶ)と、各機能別ロボットの最上位である機能別ロボット層に実装する機能別ロボットを制御する Function Control Agent(以下、FCA と呼ぶ)、機能別ロボットが取得・生成するセンサデータ、

制御パラメータを他のロボットへ公開/他のロボットから取得する Data Sharing Agent(以下, DSA と呼ぶ) を定義する。これらのエージェント群は非同期で動作する周期タスクである。

統合層に実装する MPA と TMA と SCA は、それぞれ図 6.4 の脳ロボットの行動計画部、機能別ロボットへのタスク指示部、機能別ロボットの状態管理部に相当する。また、機能別ロボット層に実装する FCA と DSA は、それぞれ図 6.4 の機能別ロボットの動作計画部、環境モデリング部に相当する。図 6.17 に示す SCA と DSA が複数個あるのは、機能別ロボットが公開する制御情報、センサ情報が様々な周期で更新されるからである。

モジュール間通信機構 RL-Net は、脳ロボットと機能別ロボットが Global Control, Co-operative Control を実現し、行動計画部の並列・分散処理化を確立するために、エージェント間の通信を提供する。具体的には以下のような通信を行う。

- MPA が行動計画を実行
- MPA の計画に基づいて TMA が各機能別ロボットへタスク要求
- FCA がタスクを受け取り、制御開始
- 制御情報、センサ情報を DSA が他のロボットへ公開/他のロボットから取得
- SCA-DSA 間の通信を基に MPA が Global Control を、DSA-DSA 間の通信を基に FCA-FCA 間で Cooperative Control を実行

以降では、エージェント間の通信方法の詳細設計について述べる。

イベント通信とデータ通信の分離

モジュール型ヒューマノイドロボットを構成する機能別ロボット間で通信するデータは、タスク要求や同期信号等の分散制御に重要なイベントを通知するためのデータ(以下、イベントと呼ぶ)と、センサデータ、制御パラメータ等の実際の制御に必要なデータ(以下、データと呼ぶ)の大きく2つに分類される。機能別ロボット間のイベント伝達は、通信データのサイズは比較的小さく、通信レイテンシをできるだけ小さくしたい性質がある。一方、機能別ロボット間のデータ伝達は、通信データのサイズは比較的大きく、バンド幅を確保したい性質がある。そこで、RL-Net では、レスポンスリンクの特徴であるイベントリンク、データリンクを利用することで、イベント・データ通信のリアルタイム性を保証する。

RL-Net では、図 6.5 に示したソフトウェアアーキテクチャを実現するために、機能別ロボット間で 1) タスク要求、2) データ共有、3) 緊急命令、4) 機能別ロボット間の同期、の大きく4種類の通信を確立する必要がある。1) の通信は、脳ロボットから他の機能別ロボット

へのタスク要求，または機能別ロボット間のタスク要求である．具体的には，TMA-FCA 間，FCA-FCA 間で行われる，エンドエフェクタの任意の位置への移動，任意の位置へ時間 t での移動，任意の対象物 X の三次元位置の測定等の抽象化された命令のことである．2) の通信は，行動計画やモジュール間の協調に必要な制御データ，センサデータの共有を行うものである．具体的には，SCA-DSA 間，DSA-DSA 間で行われる，1msec でセンサデータを，10msec で腕の現在位置を，33msec で画像処理結果を送受信する通信のことである．3) の通信データは，TMA-FCA 間，FCA-FCA 間で行われる，突発的な障害物回避等のリアクティブ行動，モジュール間通信のネットワークトラブル回避を行うために必要な緊急命令のことである．4) の通信データは，タスク要求への返答，機能別ロボット間の動作の同期，機能別ロボットが動作しているか確認するためのハートビートである．具体的には，TMA-FCA 間，FCA-FCA 間で行われる，タスク発行に対する ACK や，そのタスクの実行結果(タスク完了，未完了)，時刻合わせに関する通信のことである．

本 RL-Net では，通信の性質より，(1)，(3)，(4) をイベント，(2) をデータに分類する．

イベントパケットとデータパケット

本節では，RL-Net の通信パケットについて述べる．

図 6.18 は，RL-Net 上で通信されるレスポンスリンクのイベントリンク用のイベントパケット，データリンク用のデータパケットを示している．

RL-Net では，パケットヘッダ部を図 6.19 で示すように送信側及び受信側のアドレスから構成し，それぞれのアドレスを機能別ロボット ID(7 ビット) とエージェント ID(8 ビット) から構成することで，ネットワーク上の通信相手を一意に決定する．

次に，トレイラのフォーマットを図 6.20 に示す．各行は 1 バイトに相当する．これらのうち，Full ビットは有効なデータがペイロードを満たしていることを示す．一方，送信データが 56 バイトに満たない場合には，Data Length にデータ長を格納することが可能である．誤りが生じた場合には，Dirty0 ~ 15 ビットが設定される．Fatal ビットはハードウェアによりエラー訂正できない誤りが発生したことを示し，Correct はハードウェアにより訂正可能であったことを示す．一方，Start ビット及び End ビットはメッセージの先頭及び末尾を示すためのビットである．また，Int. ビットが設定されている場合，受信側において割り込みを生じる．RL-Net は，受信時に割り込みがかかるように Int. ビットを設定することとする．Serial Number フィールドはメッセージのうち何番目のパケットであることを示す．

ハードウェアによるパケットの自動送受信プロトコル

本節では，RL-Net の通信プロトコルについて述べる．

0	UD	Full	Data Length					
1	Dirty0	Dirty1	Dirty2	Dirty3	Dirty4	Dirty5	Dirty6	Dirty7
2	Dirty8	Dirty0	Dirty10	Dirty11	Dirty12	Dirty13	Dirty14	Dirty15
3	Start	End	Int.	Fatal	Correct	Serial Number (Cnt.)		

図 6.20: RL-Net のパケットトレイラ

2. レスポンシブリンク制御レジスタの設定
3. DMAC の設定と起動
4. DMAC がプロセッサバスを獲得し，SDRAM から送信データをレスポンシブリンクに対して送信バッファ長だけバースト転送
5. 送信バッファ長のバースト転送が終了後，プロセッサバスを開放
6. 送信バッファ分の送信が終了したら，レスポンシブリンクがDMAC にDMA リクエストを行い，次の送信データを自動転送(上記4，5の繰り返し)
7. 送信データを全て送信し終わったらプロセッサに対して割り込みを発生し，送信終了を伝達

また，レスポンシブリンクを利用したパケットの受信は，受信データを保存する領域をSDRAMに確保し，その先頭アドレスとサイズをDMACおよびレスポンシブリンク制御レジスタに設定すれば，送信時と同様に，ハードウェアで自動的に行うことができる．受信は以下の手順で処理が行われる．

1. 受信データ用領域をSDRAM上に確保
2. レスポンシブリンク制御レジスタの設定
3. DMAC の設定と起動
4. レスポンシブリンクが受信バッファに設定したサイズのデータを受信したらDMACに対してDMAリクエストを発行
5. DMAC がプロセッサバスを獲得し，レスポンシブリンクの受信バッファから受信データをSDRAMにバースト転送
6. 転送長のバースト転送が終了後，プロセッサバスを開放(上記4，5の繰り返し)

7. 受信用 SDRAM 領域に指定したサイズだけ受信するか、割り込み要求ビットが設定されている受信パケットを受信するとプロセッサに対して割り込みを発生

レスポンスプロセッサは、イベント通信のレイテンシをできるだけ低く抑えるため、レスポンスプロセッサコアの内部バスであるハーバードバスとイベントリンクを直接接続している。つまり、イベントリンクは、プロセッサからはオーバーヘッドがまったく存在しない専用線として利用でき、非常に高速なデータ転送を行うことができる。

RL-Net では、レスポンスリンクのイベントリンクとデータリンクを利用した送受信プロトコルをそれぞれ管理するイベントリンクマネージャとデータリンクマネージャを RT-Frontier のカーネルスレッドとして設計・実装し、カーネルレベルでパケットの送受信処理を行う。具体的には、イベントリンクマネージャとデータリンクマネージャは、上述のパケットの準備とレジスタ設定 ((1) ~ (3)) と割り込み処理 (7) を行う。

一度の DMA 転送により送信するデータ量と割り込み応答時間の間にはトレードオフが存在する。割り込み応答時間が長い場合、システムへ、(1) 割り込みに対する処理が実行を開始できない (2) 優先度の高いスレッドが実行を横取りできない、などの影響を与えてしまう。ロボット制御に必要なデータ転送は、関連研究から考えても、56 バイトもあれば十分行うことができる [佐藤秀雄 他 01][石綿陽一 他 04]。そこで、割り込み応答時間と通信レイテンシを最小にするために、イベント/データリンクで 1 度に通信するパケットサイズを、それぞれ 1 パケット (ペイロードサイズ:16/56 バイト) に制限することとした。パケットサイズを 1 パケットに制限ことは、上述の手順 (6) について省くことができ、より高速な通信を行うことができる。

ユーザ空間で動作する各種エージェントは、下記の手順でパケットを送信する。

1. パケットのヘッダとトレイラを準備
2. ペイロードを生成し、ヘッダ、トレイラと合わせてパケットを生成
3. 図 6.21 に示す API をコールしてイベント/データリンクマネージャへパケットを渡す。図 6.21 中の header, payload は図 6.18 と図 6.19 に相当
4. 各マネージャが、パケットをレスポンスリンクの送信用 SDRAM にコピー

同様に、ユーザ空間で動作する各種エージェントは、下記の手順でパケットを受信する。

1. パケットを受信すると受信割り込み発生
2. 受信割り込みを受けた各マネージャは、受信データをユーザ空間で動作するスレッドがそれぞれ備える受信バッファへコピー



図 6.21: RL-Net が提供する通信用 API

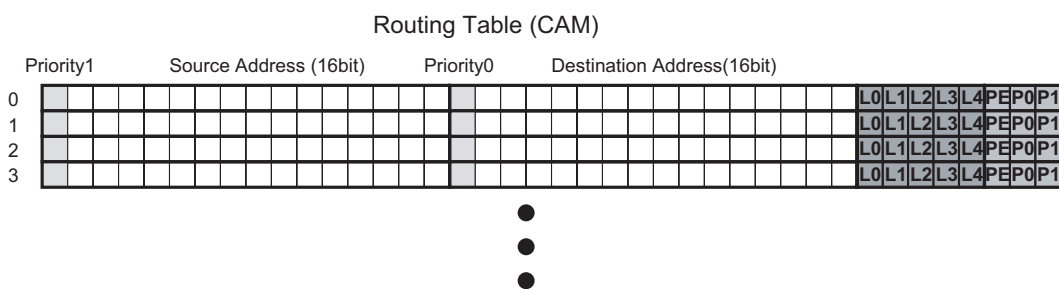


図 6.22: ルーティングテーブル

- ユーザ空間で動作するスレッドは、受信バッファをアクセスすることで、他の機能別ロボットからのデータを取得

RL-Net の機能別ロボット間パケットルーティング

本節では、RL-Net のネットワーク管理手法について述べる。

レスポンスリンクで送受信されるパケットは、図 6.19 に示した Source/Destination アドレス、パケットタイプ(イベント/データ)、そしてパケットに付加された優先度から成る合計 32 ビットヘッダを図 6.22 に示すルーティングテーブルに書き込まれた各エントリの Address Part と比較し、一致したエントリの Link Part で指定されている番号(0 から 4)のリンクに出力される。ここで、リンクの 0 番はプロセッサ自身と接続されているので、リンク 0 番へ出力すると、そのパケットはプロセッサに取り込まれることになる。Link Part の各属性ビットの値による動作を表 6.9 に示す。

表 6.9: ルーティングテーブルの属性ビット

EE	イベントテーブル有効ビット： イベントリンクに対してこのエントリが有効である
DE	データテーブル有効ビット： データリンクに対してこのエントリが有効である
PE	優先度付け替え有効ビット：優先度の付け替えをする
P1, P0	PE ビットが立っているときの付け替え後の優先度を指定
L0	CPU へ出力
L1, L2, L3, L4	リンクの 0~3 に対応．各リンクへ出力

表 6.10: ルーティングアルゴリズム

	静的ルーティング	動的ルーティング
ルーティングの設定	オフライン	オンライン
通信オーバーヘッド	小	大
ネットワークの負荷	不均等	均等

レスポンスリンクのパケットルーティングは、ルーティングに必要となるルーティングテーブルの設定はソフトウェアで行ない、パケットのルーティング自身はそのテーブルに基づいてハードウェアが行なうことになる。レスポンスリンクネットワークでは、レスポンスリンクの特徴より、同じ Source/Destination アドレスを持つパケットでも、通信ライン毎に、またパケットに付加された優先度 (2 ビット) 毎に異なった経路を指定できる。パケットの優先度は、出力する際に新たな優先度へ付け替えることも可能であり、レスポンスリンクネットワークは、これらの機能によって非常に柔軟な経路設定を可能にしている。更に、レスポンスリンクネットワークは、パケットの優先度に従い、より高い優先度のパケットが低い優先度のパケットをノード上においてに追い越すことができるという、パケットの追い越し機能を有している。

RL-Net では、これらの特徴を生かしたパケットルーティングを行うために、ルーティングマネージャを RT-Frontier のカーネルスレッドとして設計・実装する (図 6.15)。

以降では、ルーティングマネージャがサポートする機能別ロボット間のパケットルーティング手法について述べる。一般的に、複数のプロセッサを結ぶ通信路の設計手法は、静的ルーティングと動的ルーティングの 2 つに分類できる (表 6.10)。

静的ルーティングでは、オフライン時に通信経路を決定できるため、通信時に無駄なオーバーヘッドがない。ホップ数が等しい経路が複数ある場合には、負荷を分散させることが

できるが、各リンクの負荷が不均等な状況に陥る場合もある。

一方、動的ルーティングでは、システム動作中に経路設定を許容する方法であり、通信路や機能別ロボットの故障等の障害時にオンラインで経路変更することができ、静的ルーティングに比べロバスト性がある。しかしながら、動的ルーティング手法は、ネットワークトラブル等で通信路の再構築が始まると、通信路選択の計算、ネットワーク情報の全機能別ロボットへの再配布が必要となり、そのオーバーヘッドは、ロボットの制御に多大の影響を与える恐れがある。IEEE1394 のバスリセットがその例である。

一般的に、プロセッサ間通信機構では、故障等に対するロバスト性を要求される。情報処理分野の並列分散処理機構では、動的ルーティングによりロバスト性を保証しているものもあるが、自動車や飛行機等のクリティカルなシステムでは、通信インタフェースを冗長化した静的ルーティング手法を用いることでロバスト性を保証している。この手法は、パケット廃棄に備えるため、通常使用する通信経路とは別にバックアップ用の通信経路を用意し、それぞれに同一のパケットを送信することで、故障時の通信路の再設定にかかる処理時間を可能な限り削減することが可能とする。以上より、RL-Net は、自動車や飛行機と同様に通信インタフェースを冗長化した静的ルーティング手法を採用することとした。

図 6.23 は、脳ロボットの TMA から左腕ロボットの FCA へレスポンスリンクのリンクポート 2 から優先度 2 のイベントを送信するルーティング、左腕ロボットの FCA から TMA への優先度 1 のイベントをレスポンスリンクのリンクポート 1 で受信するルーティングの設定を示している。図 6.23 中の Header0、Header1 は図 6.21 の header に相当する。

```
Header0 = RL_HDR(Brain, TMA, L-Arm, FCA);  
set_rtable_entry(Header0, RLPRIORITY_2, RL_EVENT, RLT_LINK2);  
  
Header1 = RL_HDR(L-Arm, FCA, Brain, TMA);  
set_rtable_entry(Header1, RLPRIORITY_1, RL_EVENT, RLT_LINK1);
```

図 6.23: ルーティングテーブル例

優先度に基づいたパケットスケジューリング

RL-Net では、レスポンスリンクのパケットへの優先度付与に基づいた追い越し機能により、機能別ロボット間のリアルタイム通信を実現する。RL-Net では、レスポンスリンクの特徴より、各機能別ロボットにおいて、パケットが到着した時に既に送信が始まっているパケットがある時と、同優先度のパケットがバッファに並んでいる時以外、優先度

の高いパケットが待つことはなく、優先度の高いパケットは優先度の低いパケットより確実に先に送り出されることでリアルタイム通信を保証している (図 6.10, 図 6.11)。本節では、パケットヘモジュール型ヒューマノイドロボットシステムに最適な優先度を付与する手法について提案する。

動的優先度スケジューリングである EDF(Earliest Deadline First) や固定優先度スケジューリングである RM(Rate Monotonic) を代表とする、ほぼ全てのリアルタイムスケジューリングアルゴリズムは、優先度に従ってプリエンブションを行いながら、タスクを実行する。レスポンスリンクの優先度によるパケットの追い越し機能は、リアルタイムスケジューラのプリエンブションに相当する。そこで、RL-Net のパケットへの優先度の付与方法はリアルタイムスケジューリング理論を基に考えることとする。

RL-Net は、一部のイベント通信を除くとイベント・データ通信は一定周期で行われること、静的ルーティングを用いることより、固定優先度スケジューリング理論の方が、RL-Net には適していると考えられる。固定優先度によるリアルタイムスケジューリングとして RM, DM(Deadline Monotonic) 法がある。RM 法では、各タスクの周期 T とデッドライン D は同一であるという前提条件があるが、実際のリアルタイム処理では、デッドラインが周期長より短くなる場合はある。DM 法は、RM 法を $T > D$ の場合に拡張したアルゴリズムであり、優先度は、デッドラインの早いものが高くなる。DM 法は、 $T > D$ の条件下で最適なアルゴリズムであることが知られている [Audsley 90]。そこで RL-Net では、DM 法と同様に、デッドラインの早いパケットに、高優先度を割り当てることとした。

機能別ロボット間では、先にも述べたように、1) タスク要求、2) データ共有、3) 緊急命令、4) 機能別ロボット間の同期の 4 種類の通信データが存在する。1) の通信データは、機能別ロボット間の同期、低レイテンシを必要とする性質がある。また、システム全体のコンシステンスを保つためにも、脳ロボットからのタスク要求のほうが、機能別ロボット間のタスク要求よりも重要である。2) の通信データは、比較的サイズの必要な制御パラメータやセンサデータを共有する必要があるため、低レイテンシとスループットの両方とも重要である。3)、4) の通信データは、システムへの悪影響や問題を極力小さく押させるために、低レイテンシを必要とする性質がある。4) の通信データは、これらのデータサイズも小さいが、遅延を抑える必要がある。上述した通信データの性質から、RL-Net では、4 段階の優先度決定を表 6.11 のように固定して利用することとした。

RL-Net の性能を明らかにするために、レスポンスリンクネットワークのホップ数と通信レイテンシに関する性能を評価した。評価は、イベントリンク、データリンクともに、図 6.8 の C1 から任意のホップ数分離れた機能別ロボットへの通信レイテンシを、(1) ネットワークが無負荷状態、(2) C1 と通信対象のコントローラ以外の全コントローラが、通信対象のコントローラへ向けて低優先度のパケットをベストエフォートで送信している最悪

表 6.11: 通信パケットの優先度

優先度	イベントパケット	データパケット
0	ハートビート	-
1	機能別ロボット間の命令発行, フロー制御	-
2	脳ロボットから機能別ロボットへのタスク指示, グローバル同期	機能別ロボット間のデータ共有
3	緊急停止命令	脳-機能別ロボット間のデータ共有

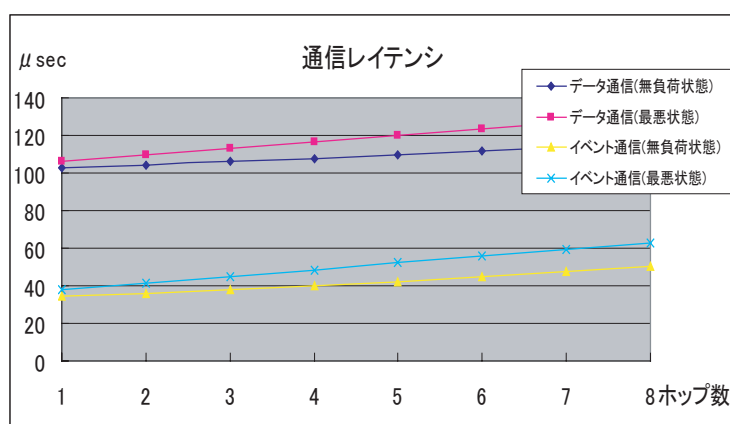


図 6.24: RL のホップ数と通信遅延に関する評価

のネットワーク状態の2通りで測定することで行った。図 6.24 の横軸はホップ数を、縦軸は1パケットのイベント、データを送受信にかかる通信レイテンシを示す。

図 6.24 は、RL-Net ではの高優先度を持つパケットが、ネットワークスイッチ毎で優先度追い越しをすることによって、ネットワークの負荷に影響を受けずに伝送され、最悪状態と無負荷状態間の値に必ずバウンドされることを示している。図より、イベントリンクを用いて1パケット通信した場合の無負荷時と最悪時の遅延時間はそれぞれ次式で表すことができる。

$$\text{無負荷時} : 32.1 + 2.4 \times \text{ホップ数} (\mu\text{sec}) \quad (6.2)$$

$$\text{最悪時} : 34.2 + 3.6 \times \text{ホップ数} (\mu\text{sec}) \quad (6.3)$$

またデータリンクを用いて1パケット通信した場合の無負荷時と最悪時の遅延時間は次式で表すことができる。

$$\text{無負荷時} : 100.5 + 1.9 \times \text{ホップ数} (\mu\text{sec}) \quad (6.4)$$

$$\text{最悪時: } 102.4 + 3.5 \times \text{ホップ数} (\mu\text{sec}) \quad (6.5)$$

モジュール型ヒューマノイドロボットでは、脚、腕、頭の粒度でシステムを分割するため、システム内のホップ数は、最大でも8ホップである。このことから、1パケットの通信にかかる最悪時の通信遅延時間は、イベントリンクなら $63\mu\text{sec}$ 、データリンクなら $130\mu\text{sec}$ となる。以上のように、モジュール間通信機構である RL-Net では、通信遅延時間 $T_{communication}$ をバウンド可能なリアルタイム通信を保証できるので、システム化技術の開発者は、再構成可能なモジュールロボットを構成する機能別ロボットが分散制御可能かどうかを判定することができ、トップダウンに制御システムを設計することが可能である。

優先度付与に関する問題

モジュール間通信機構 RL-Net は、表 6.11 のようにパケットの優先度を決定することで、図 6.24 に示す優先度に応じたリアルタイム通信を実現できる。しかし、優先度の高いパケットの通信量が設計上多くなった場合、RL-Net は優先度の高いパケットが全て送信されてしまうまで、優先度の低いパケットが送り出されなくなる問題を引き起こしたり、通信そのものが確立できなくなる恐れもある。

この問題の回避策の1つとして、機能別ロボット間をパケットの衝突が全く生じない専用回線で接続する手法がある。例えばヒューマノイドロボットを開発した際に、当初は頭モジュール、肩モジュール、肘モジュール、指モジュールと接続し、それらの経路をホップして通信を行っていたとする。設計後にどうしても頭モジュールと指モジュール間の通信レイテンシが間に合わないと判明した場合、RL-Net では、後付で頭モジュールと指モジュール間に通信経路を増設することにより対処することができる。しかしながら、ヒューマノイドロボットの体内 LAN などの大規模なネットワークを構築する際には、配線長、搭載スペースの関係上、全ての通信路に専用回線を用いることは不可能に近い。つまり、RL-Net を利用してロボットの並列分散処理ソフトウェアを実現するためには、体内 LAN 全体の通信レイテンシ、バンド幅を保証できるように、優先度に基づいてホップして通信するのか、専用回線を用いて通信するのかを判断し、ネットワークのトポロジを設計する必要がある。

しかしながら、RL-Net では、トポロジフリーであるレスポンスリンクの特徴より、ロボット内の同一機能別ロボット間には複数個の通信経路の候補が存在し、システムが大規模になればなるほど、その候補数も増えてしまう。その結果、通信レイテンシとバンド幅の要求を満たす通信経路を設計するのが困難になる。

そこで、RL-Net は、ロボットの制御アルゴリズムを実際に設計する前に、各機能別ロボットが要求する通信レイテンシとバンド幅を保証できるかどうかシミュレーションできる RL-NetSim 機構を備えることとする。

F_ID	A_ID	Delay	Band	E/D	PRI	L1F	L1A	L2F	L2A	L3F	L3A	L4F	L4A
0	0	200	10	E	0	2	0						
0	1	100	10	D	0	2	1						
1	0	100	10	D	0			3	0				
2	0	100	10	E	0	0	0						
2	1	200	10	D	0	0	1						
3	0	100	10	D	0			1	0				

F_ID:機能別ロボットID
A_ID:エージェントID
Delay:遅延時間(μ sec)
Band:バンド幅(Mbps)
E/D:イベントパケット, データパケット
PRI:優先度
LxF:レスポンスリンクのx番目のポートを介して接続している相手先機能別ロボットID
LxA:レスポンスリンクのx番目のポートを介して接続している相手先エージェントID

図 6.25: RL-NetSim

RL-NetSim の動作手順は以下である .

1. RL-Net 上で通信を行うエージェント同士を図 6.25 のように記述 . 記述内容は , エージェント名 , RL ポート , イベント通信かデータ通信か , 通信遅延時間 , バンド幅である .
2. 1 の定義に対して通信遅延 , バンド幅の 2 項目に関する評価テストを実行

通信遅延に関する評価テストは以下のように行う . RL は 4 段階の優先度を保持しており , スイッチ部分において優先度の高いパケットが優先度の低いパケットを追い越し可能であるという特徴を持つ (図 6.24) . つまり , スイッチ部分で優先度の高いパケットとの衝突が生じた際には , 優先度の高いパケットが先に送られる . 従って , ルーティングマネージャは , 遅延保証に関しては , 優先度の高いパケットの影響を考えなければならない . ここで , 機能別ロボットの各リンクにおいて , 新たな通信接続の確立要求と比較して高い優先度を持つパケット数を Pkt_{num} と定義した場合 , それらの影響によって生じる最大遅延時間 Del_{higher} は以下の式で表すことができる .

$$Del_{higher} = Pkt_{num} \times Del_{cut_through} \quad (6.6)$$

式で求めた Del_{higher} を利用して , 要求する遅延時間 Del に関する評価テストは式のように表すことができる . 式で示される条件式を満たす場合には , 遅延に時間制約を保証可能で

ある．

$$\text{Del} - \text{Del}_{\text{higher}} > 0 \quad (6.7)$$

バンド幅に関する評価テストは以下のように行う．機能別ロボット間が通信する周期，データサイズから計算できるバンド幅の合計が RL が規定するバンド幅以下に収まっている必要がある．RL は 4 つの外部インタフェースを保持している．従って，ルーティングマネージャは，各リンクにおけるバンド幅状況を管理し，余剰バンド幅を B_{avail} として保持する．要求するバンド幅 B に関する評価テストは式のように表すことができる．式で示される条件式を満たす場合には，要求バンド幅を保証可能である．

$$B_{\text{avail}} - B > 0 \quad (6.8)$$

RL-NetSim は，解析結果を，ルーティングマネージャが利用可能な図 6.23 のフォーマットとして出力し，制御ソフトウェアを記述する際にそのまま利用可能である．通信路の一部でも評価テストで許可されない場合，ロボットは安定した動作を実現することができないため，RL-NetSim が機能別ロボット間の遅延時間，通信量に関するエラーを示す．

6.2 プロトタイプロボット P1 を利用した予備実験

本節では，ヒューマノイドロボットの再構成機構を設計する前に，ここまで設計した行動計画の並列・分散化，階層化ソフトウェア，モジュール間通信機構 RL-Net の有効性を検証する．検証は，図 6.26 に示す脳ロボット，移動ロボット，頭ロボットから構成するプロトタイプロボット P1 [平 他 03] [Taira *et al.* 04] を利用した．ただし，ここまでの設計を評価するために開発したプロトタイプロボット P1 は，再構成機構を備えていないため，機能別ロボット毎に分離することは不可能である．

6.2.1 プロトタイプロボット P1 のシステム構成

プロトタイプロボット P1 を構成する脳ロボット，移動ロボット，頭ロボットの特徴は以下である．各機能別ロボットは，制御用プロセッサにはレスポンスプロセッサを，制御用 OS には RT-Frontier を利用してゐる．また，各機能別ロボット間は RL-Net で接続する．

- 脳ロボット

- システム全体の管理

- － ソフトウェア構成

- * SCA(1msec 周期)

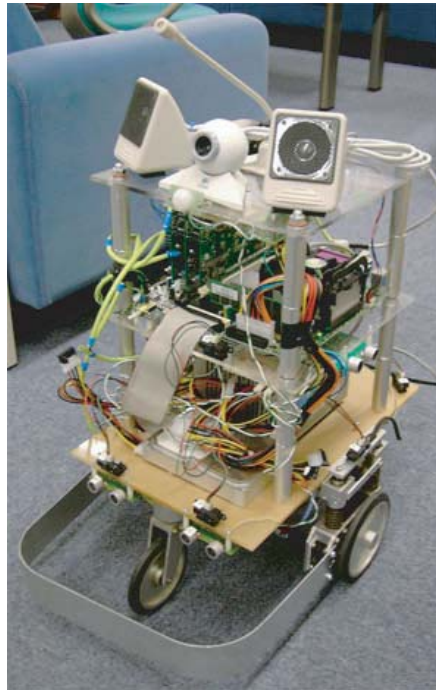


図 6.26: プロトタイプロボット P1

- * TMA(10msec 周期)
- * MPA(100msec 周期)

- 車輪移動ロボット

自身が備えるセンサ情報に基づいたパスプランニング, ナビゲーション, 障害物回避, 緊急停止機能を提供

- 機械構造

左右の各車輪 1 自由度の計 2 自由度

- 電気・電子ハードウェア

内界センサとしてジャイロセンサ (Fiber Optical Gyro), エンコーダを, 外界センサとして超音波センサ, 赤外線センサ, 接触センサを装備

- ソフトウェア部

- * 内界センサの値を参照した速度制御 (1msec 周期)
- * 内界センサと外界センサの値を参照してカルマンフィルタを用いて自己位置推定 (10msec 周期)
- * パスプランニング (100msec 周期)



図 6.27: プロトタイプロボット P1 の情報処理システム

- 頭ロボット

視覚処理機能と聴覚処理機能を提供

- 機械構造

自由度なし

- 電気・電子ハードウェア

2.2.2 節のヒューマノイドロボットに必要な情報処理能力で述べた、 320×240 の画像から 11×11 の小領域を対象として全画面の距離画像を 10msec 周期で生成する性能を確保するためには、レスポンスプロセッサは先にも述べたように処理能力が十分でない。そこで、[Uchimura *et al.* 04] のように、2.8GHz の Pentium 4 プロセッサ (Memory:512M バイト, OS:ART-Linux) をレスポンスプロセッサのサブコントローラとして使用することとした。レスポンスプロセッサとこのサブコントローラは PCI 経由で接続する。前述したようにレスポンスプロセッサの後継版である Responsive Multithreaded Processor へ移行することが前提であるため、今回の実装では、このような構成は問題なしとして扱う

- ソフトウェア部

- * 対象物のトラッキングや肌色検出, 顔認識, 2次元距離計測機能

- * HMM を元に音声認識, 音声出力, 音源定位機能

図 6.27 は、脳ロボット、車輪移動ロボット、頭ロボット用のレスポンスプロセッサと、それらを接続するレスポンスリンクを含む、プロトタイプロボット P1 が搭載する情報処理システムを示している。

図 6.28 はプロトタイプロボット P1 の制御ソフトウェアを示している。脳ロボットの TMA, SCA は、各機能別ロボットの FCA, DSA とそれぞれ RL-Net 上で接続し、各機能別ロボット

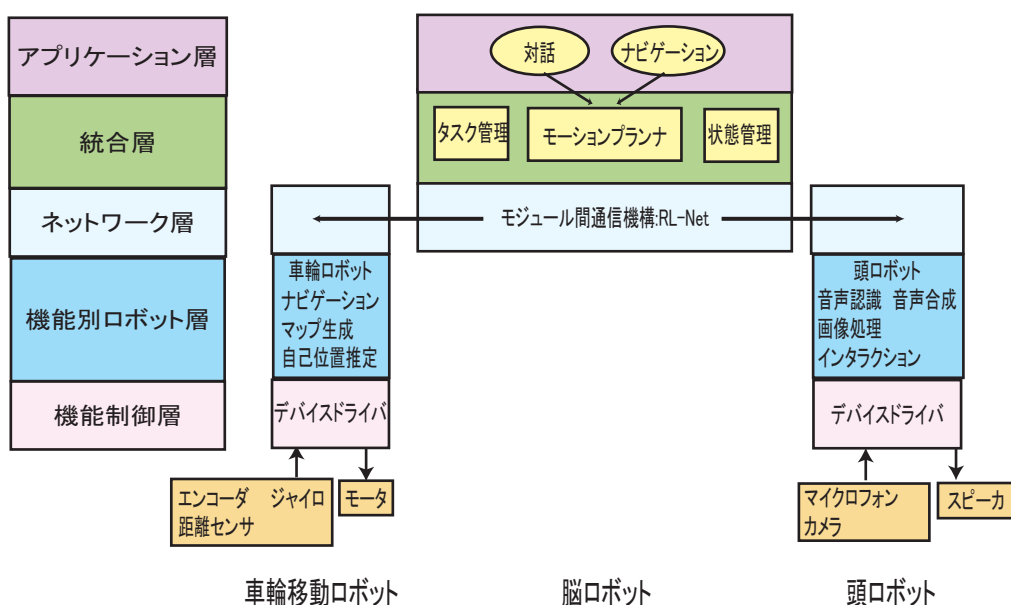


図 6.28: プロトタイプロボット P1 の制御ソフトウェア構成図

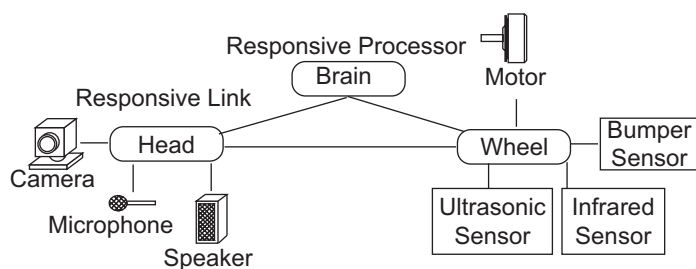


図 6.29: プロトタイプロボット P1 の情報処理システムの構成図

が提供する機能を利用して Global Control を実現する．各機能別ロボットの Local Control に関するにて提供する機能は上述通りである．プロトタイプロボット P1 の情報処理システムは，車輪移動ロボットと頭ロボットは，RL-Net 上で接続する Cooperative Control により障害物回避機能を提供する．この Cooperative Control は Global Control とは別の制御ループで形成され，Global Control とは並列で動作する．

6.2.2 プロトタイプロボット P1 を利用した予備実験

本節では，システムアーキテクチャの有効性を示すためにプロトタイプロボット P1 を利用して行った実験例を示す．図 6.30，図 6.31，図 6.32 は，プロトタイプロボット P1 のアプリケーション例を示している．

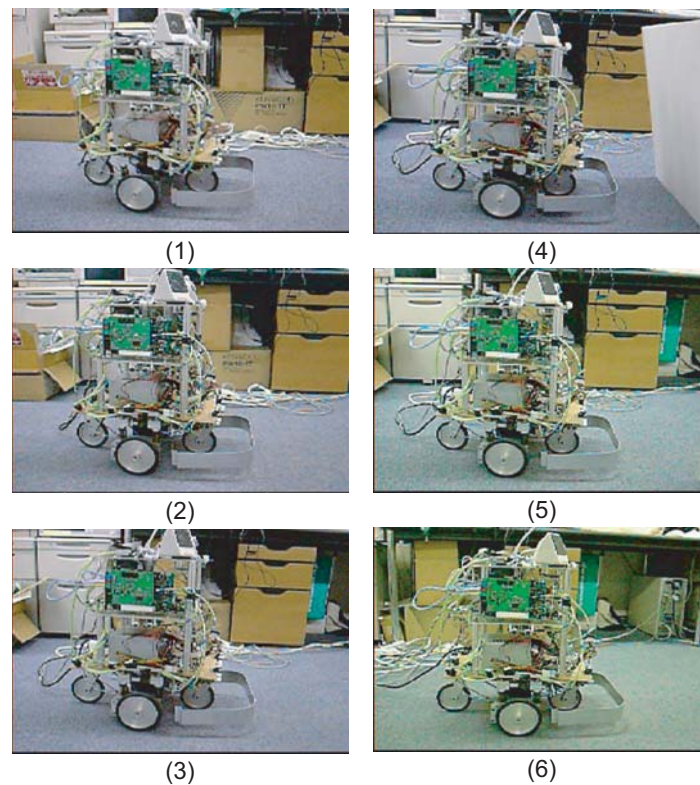


図 6.30: アプリケーション 1:障害物回避行動

1つ目の動作例は、ロボットが障害物を回避して進む動作である。図6.30は、脳ロボットのタスクに基づいて車輪移動ロボットがナビゲーションタスクを実行するアプリケーション例である。図6.30の番号は、下記の番号と対応している。

1. ユーザが音声入力でもうロボットに対してタスクを要求
2. 脳ロボットが車輪移動ロボットへ目的地までナビゲーションするようにタスクを要求
3. タスク要求を受けた車輪移動ロボットはナビゲーションタスクを実行
4. 車輪移動ロボットは動的な障害物を発見すると停止動作を実行
5. 車輪移動ロボットは障害物が動的な障害物が移動するまで停止動作を実行
6. 車輪移動ロボットは障害物が移動することを認識すると、ナビゲーションタスクを再開

2つ目の動作例は、ユーザが端末で指示した場所までロボットが自律移動する動作である。図6.31は、ユーザが端末で指示した場所までロボットが自律移動を行っている様子である。図6.31の番号は、下記の番号と対応している。

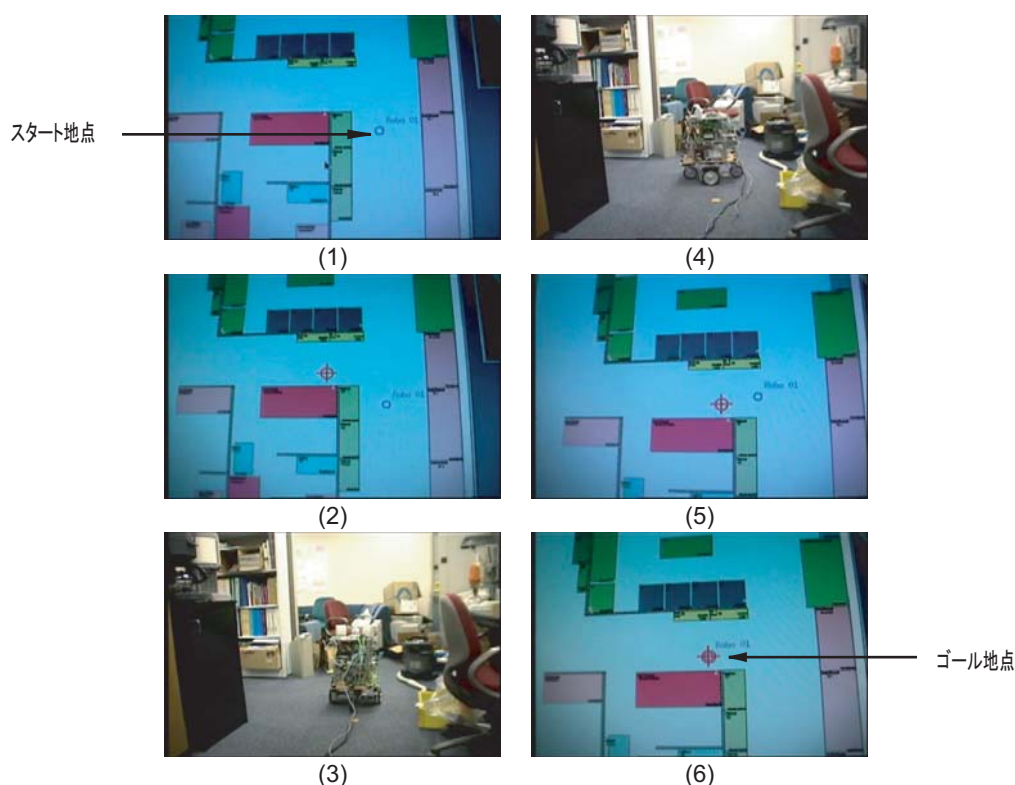


図 6.31: アプリケーション 2: 自律移動

1. ユーザが遠隔にあるユーザ端末 (コンピュータ) から脳ロボットへ無線ネットワークを介してスタート地点からゴール地点へ移動するようにタスクを要求
2. 脳ロボットがそのタスク要求を実現するための行動計画を実行
3. 脳ロボットの行動計画の結果, 車輪移動ロボットへゴール地点へ移動するナビゲーションタスクを要求
4. 脳ロボットの指示を受けた車輪移動ロボットは, あらかじめ保持する環境内のマップに基づいてナビゲーションタスクを実行
5. 障害物を認識したら, その障害物を回避しながらゴール地点へナビゲーション
6. 移動後, 脳ロボットはユーザへタスク終了の合図を送信

3つ目の動作例は, ロボットと人間とのインタラクションである. 図 6.32 は, 人間とのインタラクションしている様子を示している. 図 6.32 の番号は, 下記の番号と対応している.

1. ユーザが遠隔にあるユーザ端末 (コンピュータ) から脳ロボットへ無線ネットワークを介してユーザのいる位置へ移動するようにタスク要求

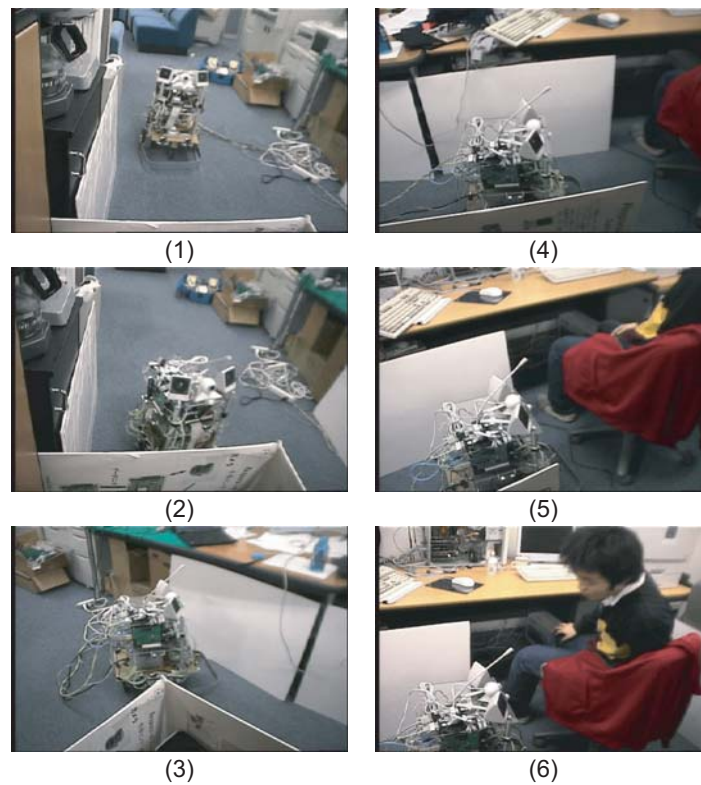


図 6.32: アプリケーション 3:人間とのインタラクション

2. 脳ロボットがそのタスク要求を実現するための行動計画を実行
3. 行動計画の結果, 車輪移動ロボットへB地点へ移動するようにタスクを要求する. ただし, 車輪移動ロボットは環境内のマップを保持している. また同時に, 頭ロボットへユーザの顔検出を行うようにタスク要求
4. 脳ロボットのタスク要求を受けた車輪移動ロボットは障害物を回避しながらユーザのいる地点へ移動. 同時に, 頭ロボットはユーザの顔検出を実行
5. 頭ロボットがユーザを発見後, 脳ロボットは, 頭ロボットへは人間とのコミュニケーションを, 車輪移動ロボットには移動停止動作のタスク要求
6. 頭ロボットは人間とコミュニケーションを行い, 車輪移動ロボットは移動停止動作を実行

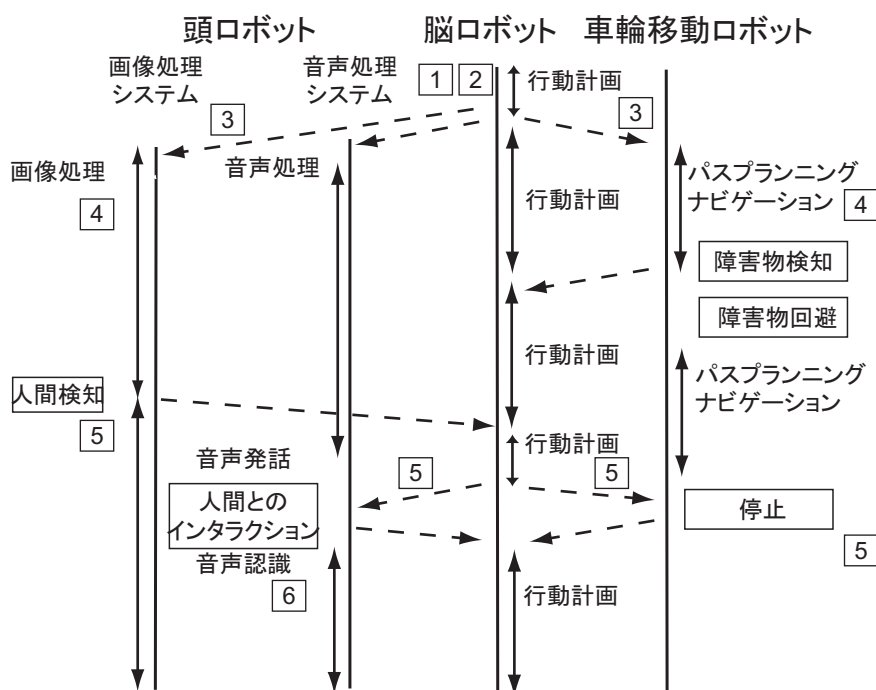


図 6.33: アプリケーション 3 の情報処理システムの制御フロー

6.2.3 プロトタイプロボット P1 を利用した予備実験の評価・考察

図 6.33 は、図 6.32 に示したアプリケーション 3 を処理している際の情報処理システムの動作を示している。図 6.33 中に示す番号は、先に述べたアプリケーションの動作を示した番号と図 6.32 の番号と対応している。

この図 6.33 から、プロトタイプロボット P1 が搭載する情報処理システムは、各機能別ロボットが複数の行動計画を並列処理していることが明らかである。図 6.34 は、本システムアーキテクチャを比較検討するために、車輪移動ロボットに接続していたアクチュエータ、センサなどの電気・電子ハードウェアと制御ソフトウェアを脳ロボットへ移植したプロトタイプロボット P1 の集中制御方式の情報処理システムを示している。図 6.34 に示した情報処理システムは、関連研究で述べた HRP(図 3.7) や ASIMO(図 3.10) と、1つのプロセッサ上でロボット全体の行動計画を処理している点で類似している。図 6.35 は、図 6.34 に示した集中制御化したプロトタイプロボット P1 の情報処理システムが図 6.32 に示したアプリケーションを実行した時の情報処理システムの制御フローを示している。本システムアーキテクチャに基づいた制御フロー(図 6.33)と従来の制御手法に基づいた制御フロー(図 6.35)を比較すると、脳ロボットの行動計画部の処理量が大きく異なっている。具体的には、従来の制御手法に基づいた制御フローは、図 2.7 に示した「知覚処理 環境モデリング 行動計画 行動実行」という処理を順番に実行しているだけである。行動計画部の

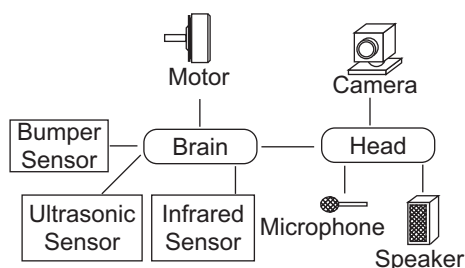


図 6.34: 集中制御化したプロトタイプロボット P1 の情報処理システム

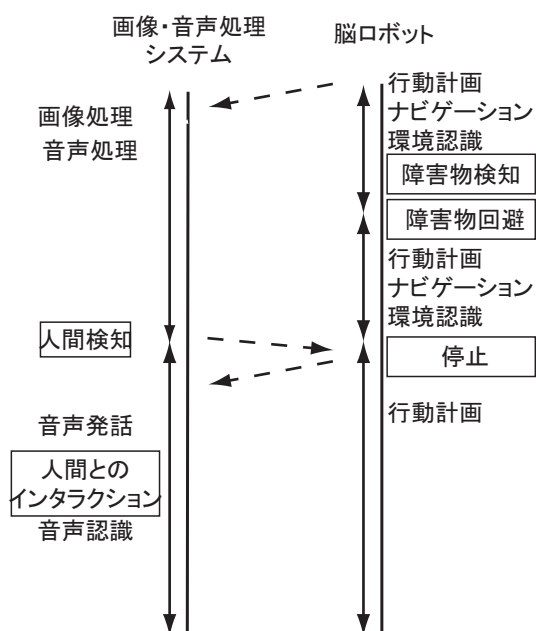


図 6.35: 従来の制御手法での制御フロー

並列性に着目した本アーキテクチャは，この一連の処理を図 6.33 に示したように並列処理可能であることを実証できた．

次に，制御フローを用いた定性的な評価から明らかになった本システムアーキテクチャの有効性を定量的に解析する．表 6.12 は，図 6.33 に示す制御フロー内の主な処理の制御周期と実行時間を示している．ただし，機能別ロボット間のデータ共有，タスク要求に関しては，最悪通信遅延時間を示している．

一方，表 6.13 は，図 6.35 に示す制御フロー内の主な処理の制御周期と実行時間を示している．ただし，機能別ロボット間のデータ共有，タスク要求に関しては，最悪通信遅延時間を示している．

表 6.12 と表 6.13 を比較すると，表 6.13 に示す処理の方が，複数の処理を並列処理できる

表 6.12: プロトタイプロボット P1 の制御

機能別ロボット	タスク	制御周期	実行時間
脳ロボット	SCA	1msec	100 μ sec
脳ロボット	TMA	100msec	2msec
脳ロボット	MPA	100msec	10msec
車輪移動ロボット	速度制御	1msec	100 μ sec
車輪移動ロボット	自己位置推定	10msec	2msec
車輪移動ロボット	パスプランニング	100msec	10msec
頭ロボット	顔認識	-	300msec
頭ロボット	2次元距離測定	33msec	10msec
頭ロボット	音声認識	-	1sec
頭ロボット	音声合成	-	1sec
頭ロボット	音源定位	-	1sec
脳ロボットのSCAと車輪移動ロボットのDSA	データ共有	1msec	105 μ sec
脳ロボットのTMAと車輪移動ロボットのFCA	タスク要求	10msec	37 μ sec
脳ロボットのSCAと頭ロボットのDSA	データ共有	33msec	105 μ sec
脳ロボットのTMAと頭ロボットのFCA	タスク要求	100msec	37 μ sec
車輪移動ロボットのDSAと頭ロボットのDSA	データ共有	33msec	105 μ sec
車輪移動ロボットのFCAと頭ロボットのFCA	タスク要求	100msec	37 μ sec

表 6.13: プロトタイプロボット P1 の制御 (集中制御時)

機能別ロボット	タスク	制御周期	実行時間
脳ロボット	MPA	120msec	40msec
脳ロボット	速度制御	1msec	100 μ sec
脳ロボット	自己位置推定	40msec	2msec
頭ロボット	顔認識	-	300msec
頭ロボット	2次元距離測定	33msec	10msec
頭ロボット	音声認識	-	1sec
頭ロボット	音声合成	-	1sec
頭ロボット	音源定位	-	1sec
脳ロボットと頭ロボット	タスク・データ	120msec	37 μ sec

ので、各実行時間を小さく抑えることができ、その結果として各制御周期を小さく設定することができる。図 6.29 に示す情報処理システムは複数個のプロセッサを利用しているため、図 6.34 に示す情報処理システムよりも消費電力を必要とする。しかし、図 6.29 に示す

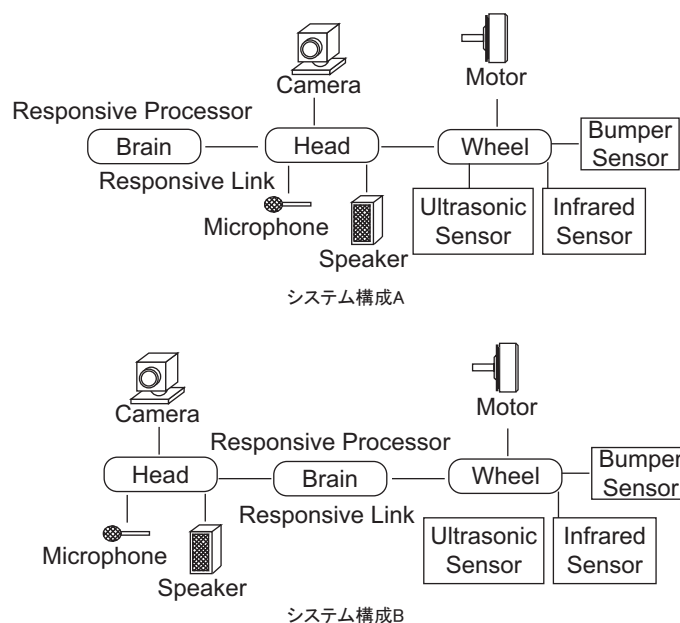


図 6.36: プロトタイプロボット P1 の情報処理システムの構成図 2

情報処理システムが処理する各々の情報量は，図 6.34 に示す情報処理システムが処理する情報量よりも小さいため，同じ精度の処理を行うのであれば，図 6.29 に示す情報処理システムは，より低消費電力のプロセッサを利用して構成することができる．一方，図 6.34 に示す情報処理システムが，図 6.29 に示す情報処理システムと同じ精度の処理を行うためには，現在使用しているプロセッサよりも高速なものを利用する必要がある．その結果，図 6.29 に示す情報処理システムの方が，消費電力を小さく抑えることができると考えられる．

ロボットの再構成機構を実現するためには，各機能別ロボット間のトポロジを変更する度に制御アルゴリズムがその影響を受けるようでは，再構成機構の利点が薄れてしまう．そこで本システムアーキテクチャでは，この問題点を解決するためにモジュール間通信機構 RL-Net を提唱している．プロトタイプロボット P1 を利用して RL-Net の評価を実施する．図 6.36 は，プロトタイプロボット P1 の情報処理システム間のネットワークトポロジを図 6.29 から変更した構成を示している．表 6.14 は，図 6.36 に示した情報処理システムで図 6.30，図 6.31，図 6.32 に示すアプリケーションを実行した際の制御周期と通信遅延時間を示している．表 6.14 と表 6.15 を比較すると，ネットワークのホップ数に応じて通信遅延時間が大きくなっている．従来の分散制御システムでは，実際に動作させるまでこの通信遅延時間を見積もることは難しいことが欠点として挙げられているが，本システムアーキテクチャでは，RL-NetSim を利用することであらかじめ通信遅延時間を見積もることができる．

このことから，プロトタイプロボット P1 の備える情報処理システムは，RL-Net により，

表 6.14: ネットワークトポロジに関する評価

構成	機能別ロボット	タスク	周期	通信時間
A	脳ロボットのSCAと車輪移動ロボットのDSA	データ共有	10msec	110 μ sec
A	脳ロボットのTMAと車輪移動ロボットのFCA	タスク要求	100msec	40 μ sec
A	脳ロボットのSCAと頭ロボットのDSA	データ共有	33msec	105 μ sec
A	脳ロボットのTMAと頭ロボットのFCA	タスク要求	100msec	35 μ sec
A	車輪移動ロボットのDSAと頭ロボットのDSA	データ共有	33msec	105 μ sec
A	車輪移動ロボットのFCAと頭ロボットのFCA	タスク要求	100msec	35 μ sec
B	脳ロボットのSCAと車輪移動ロボットのDSA	データ共有	10msec	105 μ sec
B	脳ロボットのTMAと車輪移動ロボットのFCA	タスク要求	100msec	35 μ sec
B	脳ロボットのSCAと頭ロボットのDSA	データ共有	33msec	105 μ sec
B	脳ロボットのTMAと頭ロボットのFCA	タスク要求	100msec	35 μ sec
B	車輪移動ロボットのDSAと頭ロボットのDSA	データ共有	33msec	110 μ sec
B	車輪移動ロボットのFCAと頭ロボットのFCA	タスク要求	100msec	40 μ sec

トポロジの変化に対しても柔軟に対応できることがわかった。

以上の3つの機能別ロボットから構成するプロトタイプロボットP1を利用した実験を通して、ロボットの行動計画の並列性を抽出することに着目して設計した本システムアーキテクチャは、ロボットの情報処理能力向上に対して有効であると判断できた。また、各機能別ロボットの制御ソフトウェアは、変更することなくロボットへ導入することも可能であることがわかった。さらには、機能別ロボット間を接続するトポロジに依存することなく、同じソフトウェアで制御可能であることも検証することができた。

以降では、ここまでの設計にシステム再構成機構を追加し、再構成可能なモジュール型ヒューマノイドロボットを実現する。

6.3 システム再構成機構

本節では、ヒューマノイドロボットを構成する機械構造、電気・電子ハードウェア、ソフトウェアの再構成を可能とする機構を設計する。

6.3.1 モジュール接続ユニットの設計

再構成に伴う機能別ロボット間の物理的な着脱を可能とするモジュール接続用コネクタは、再構成可能なモジュール型ヒューマノイドロボットを構成する全機能別ロボットが備

えることになる。本システムアーキテクチャでは、このモジュール接続用コネクタのことを「モジュール接続ユニット」として定義する。本システムアーキテクチャの位置付けは、人間がオフライン時に手でロボットの構成を変更する静的再構成可能なヒューマノイドロボットであるため、モジュール接続ユニットは、静的再構成をサポートすればよいこととなる。このようなモジュール接続ユニットは、再構成時に機能別ロボット間の機械構造、通信インタフェース(レスポンスリンク)、電源線を手で接続しやすい機構を備える必要がある。また、表 3.1 で明らかにしたように、再構成機構を持つモジュールロボットの問題点は、接続部分の構造が他のロボットと比べて冗長であり、ロボット全体の重量が増加してしまう点である。従って、冗長部分をどれだけ少なく抑えられるかが重要である。

M-TRAN は磁石とバネから構成するモジュール接続ユニットを利用している。M-TRAN のモジュール接続ユニットは、道具を利用しなくても再構成できる利点がある。しかしそのユニットは、利用している磁石が他の電気・電子ハードウェアへ影響を与える恐れがある、磁石とバネの動作をセンサで監視する(接続の安定性を保証する)のが困難である、重量のあるモジュールを磁石で接続するには磁石のサイズが大きくなってしまふなど欠点がある。また、M-TRAN を構成する 1 つのモジュールは 6 面体であり、全ての面で他のモジュールと接続可能である。そのため M-TRAN は拡張性・柔軟性が高いが、その分冗長な部分が多くなり、ロボット全体の重量が増加してしまっている。

AIBO は、頻繁な再構成を対象としておらず、モジュール接続ユニットを専用の工具を利用して接続する。AIBO の再構成機構は、まず本体の電源を切り、バッテリーとメモリースティックを外し、その後、専用の工具を利用してモジュールを接続することで実現する。HERMES も AIBO と同様である。AIBO のモジュール接続ユニットは、電気・電子ハードウェアへ影響を与えない、安定性を保証できる利点がある。しかしそのユニットは、専用の工具を利用しないと接続が不可能な欠点がある。AIBO は、M-TRAN と異なり、モジュール同士が接続可能な場所が限定されている。このため AIBO は M-TRAN と比較して拡張性・柔軟性は低い、冗長な部分を極力押さえられるため重量も微増である。

以上のことから、モジュール接続ユニットは下記の特徴を備えることとする。

- 機械構造への要件
 - － ユーザがモジュールの着脱を簡単に行える単純な機構
 - － ヒューマノイドロボットの激しい動きに対しても安定して動作できることを保証する機構
 - － 機能別ロボット単体で画像処理システムやマニピュレータとして利用・開発する際には、モジュール接続ユニットが支障になる恐れがあるので、モジュール接続ユニット自身もボディから簡単に着脱可能な機構

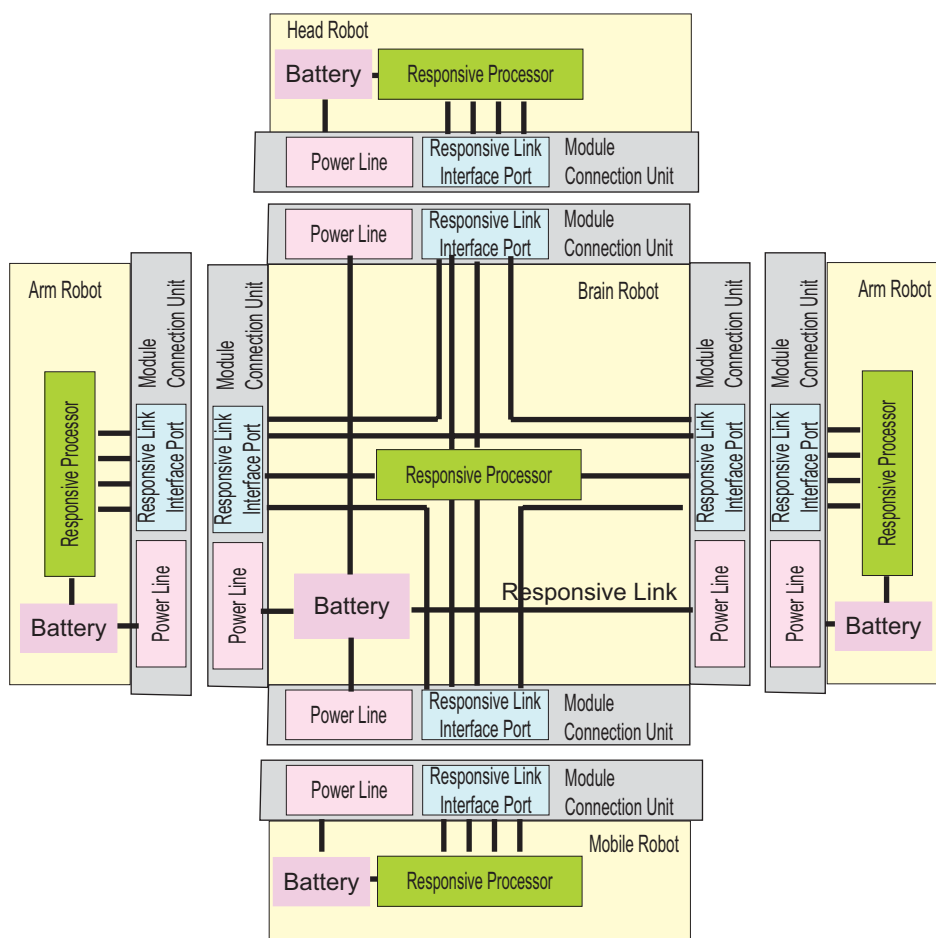


図 6.37: モジュール接続ユニットの設計概念図

- 重量増加を回避するために冗長な部分を極力押さえた機構
- 電気・電子ハードウェアへの要件
 - 他の電気・電子ハードウェアに影響を与えない機構
 - モジュール接続ユニットを接続すると、モジュール間通信インタフェース(レスポンスリンク)と電源線も同時に接続することができる機構

図 6.37 は、モジュール接続ユニットの設計概念図を示している。

図 6.37 は、「一般的なロボットに必要なもの」に「統合化に必要なもの」であるモジュール接続ユニットを追加した図を示している。モジュール接続ユニットは、機械構造、通信インタフェース、電源線を備えており、モジュール接続ユニット同士が接続することで、機能別ロボットの通信インタフェース、電源線が接続可能である。脳ロボット以外の機能別ロボットのモジュール接続ユニットが脳ロボットのモジュール接続ユニットよりも大きく

なっているのは、モジュール接続ユニットの機械構造に、簡単なはめ込み式を採用するからである。

6.3.2 再構成可能とする制御ソフトウェアの設計

再構成可能なモジュール型ヒューマノイドロボットは、用途に応じて機能別ロボットを組み換え可能な点、機能別ロボットだけで利用・開発ができる点が特徴である。新たに開発した機能別ロボットや今まで再構成可能なモジュール型ヒューマノイドロボットの一部として利用したことがない機能別ロボットを、再構成可能なモジュール型ヒューマノイドロボットに組み込むだけでは、そのモジュール型ヒューマノイドロボットは動作することは不可能である。

例を挙げて考える。新たなデバイスをコンピュータへ導入する際には、そのデバイスを制御できるデバイスドライバが必要である。デバイスドライバが既に組み込まれている場合は、そのデバイスをすぐに使用することが可能である。デバイスドライバがない場合は、現在導入しているコンピュータや OS に適応したデバイスドライバをインストールする必要がある。

モジュール型ヒューマノイドロボットもこの例と同様である。ユーザが、あるタスク X を達成するために、(脳ロボット、機能別ロボット A、機能別ロボット B) の構成で動作するモジュール型ヒューマノイドロボットのソフトウェアを利用していたとする。ユーザが異なるタスク Y を達成するためには、機能別ロボット B の代わりに自由度構成の異なる機能別ロボット C を備えるヒューマノイドロボットを利用する必要性が生じた。そこで、(脳ロボット、機能別ロボット A、機能別ロボット C) で構成するモジュール型ヒューマノイドロボットが動作するには、機能別ロボット B と機能別ロボット C の差を吸収して、新たに制御ソフトウェアを開発する必要がある。制御ソフトウェア開発が一旦終了すると、(脳ロボット、機能別ロボット A、機能別ロボット C) の構成でヒューマノイドロボットは動作可能となり、タスク Y を達成することができる。ユーザが、場面に応じて (脳ロボット、機能別ロボット A、機能別ロボット B)、(脳ロボット、機能別ロボット A、機能別ロボット C) のどちらかの構成のヒューマノイドロボットを使い分けることを可能にするためには、システムの再構成後に、再構成後のロボット自身が、どのような機能別ロボットから構成するヒューマノイドロボットであるのかを把握し、制御ソフトウェアを使い分ける必要がある。そのためには、ヒューマノイドロボットの制御ソフトウェアは、(1) 自身を構成する各機能別ロボットの種類や性能の判別し、(2) 再構成後のロボットに相応しい制御ソフトウェアへ切り替え、機能別ロボット同士で情報の共有化ができる設計が必要である。以降では、これら (1)、(2) の設計について述べる。

先にも例を挙げた OS のデバイス管理機能を再度例に挙げて、(1)、(2) の設計方針を考える。OS が、実際に何らかの作業をする際には、コンピュータに接続された周辺機器にアクセスする必要があり、周辺機器はその開発元などが書いたデバイスドライバを通して制御される。例えば、ユーザーが何かを画面に表示したいなら、文字やピクセルを表示させるためにカーネルを通してモニターのドライバ (VGA とか VESA) を使用する。デバイス管理は最初に様々なバス (PCI や USB) 上をスキャンして実装されたデバイスを検出し、対応するドライバを探す。デバイス管理は各 OS 固有の部分であり、カーネルの設計によってドライバの扱いは異なるが、一般にカーネルはドライバが物理的にデバイスにアクセスするための入出力ポートやメモリ空間を用意する必要がある。つまり、OS のデバイス管理機能では、アプリケーションが各種のデバイスを統一的に取り扱うためのインタフェース機能、および各種のデバイスに対応したデバイスドライバの登録/管理機能、デバイスドライバとのインタフェース機能を提供している。

以上から、(1)、(2) に対する本システムアーキテクチャの設計は、図 6.38 に示すように OS のデバイス管理機能を模倣する構成とする。

(1) に対する本システムアーキテクチャの設計は、各機能別ロボットが、機能別ロボットの種類を区別可能とする固有情報を備えることとする [Taira *et al.* 06]。本システムアーキテクチャでは、この各機能別ロボットが持つべき固有情報を「モジュール情報」として定義する。モジュール型ヒューマノイドロボットに必要なモジュール情報の項目を次のようにまとめる。

1. モジュール情報 1: 機能別ロボットの識別に関する情報
種類, 規格
2. モジュール情報 2: 機能別ロボットの制御量に関する情報
質量, 重心, 自由度, 可動範囲, トルク
3. モジュール情報 3: 機能別ロボットが提供する機能や能力に関する情報
提供できる機能, DSA が公開する制御データとその更新周期

モジュール情報 1 は、開発する機能別ロボットを識別・管理する情報である。本提案では、OS のデバイス管理と同様に、各機能別ロボットに固定のメジャー番号とマイナー番号を割り当てることで機能別ロボットの識別を実現する。メジャー番号は接続される機能別ロボットの種類を、マイナー番号は機能別ロボットの規格を表すことにする。本システムアーキテクチャでは、脳部, 頭部, 右腕部, 左腕部, 右手部, 左手部, 移動部の順番にメジャー番号を 0 番から割り振ることとした。

モジュール情報 2 は、機能別ロボットの制御に使用するパラメータ群である。

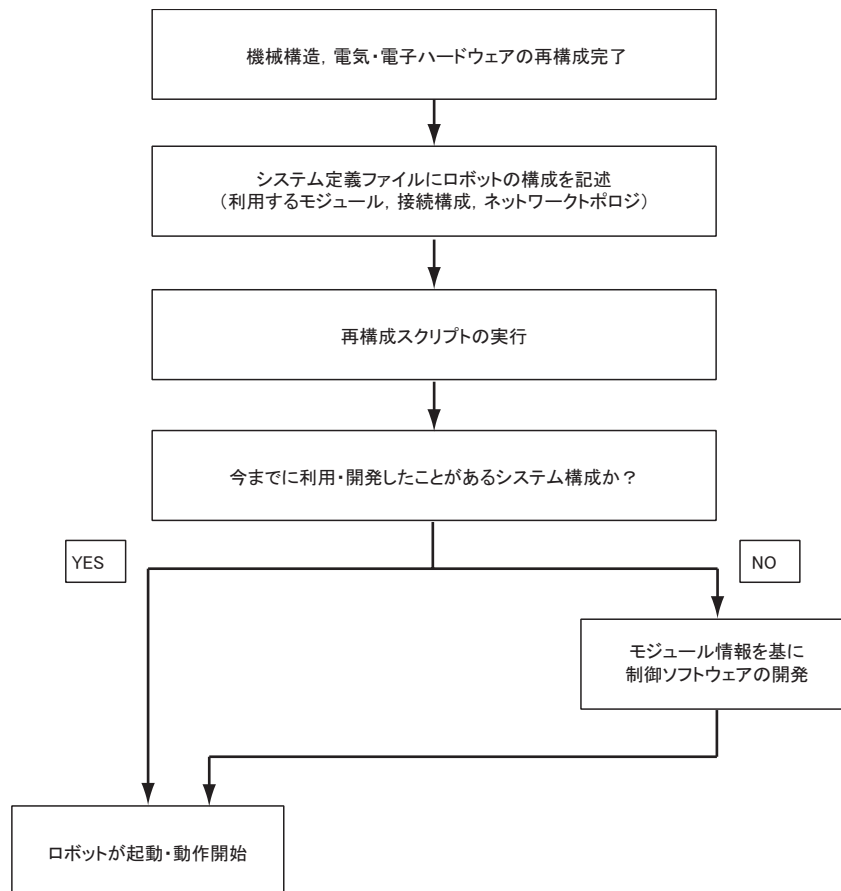


図 6.38: モジュール型ヒューマノイドロボットの制御ソフトウェアの再構成

モジュール情報 3 は、機能別ロボットが提供できる機能や他の機能別ロボットに公開できる制御データを示す情報である。機能は API の形で提供する。公開できる制御データは、公開できるデータの種類（センサデータ、カメラからの 3 次元座標、腕の現在位置等）と、データの更新周期（1msec, 10msec 等）、公開する DSA の ID の組み合わせとして表記する。

これらのモジュール情報を基に、脳ロボットは Global Control を、機能別ロボットは Cooperative Control を行う。

(2) に対する本システムアーキテクチャの設計は、以下の手順で各機能別ロボットの制御ソフトウェアを再構成後のロボットに相応しい制御ソフトウェアに切り替える構成を採用する。

1. モジュール型ヒューマノイドロボットシステム構成を定義するファイル「robot.conf」を用意。そのファイル内にモジュール情報を利用して、モジュール型ヒューマノイドロボットの構成図を定義。具体的には、モジュール情報 1, 物理的な接続位置（脳ロボットから見た位置, レスポンシブリンクのポート番号）を定義

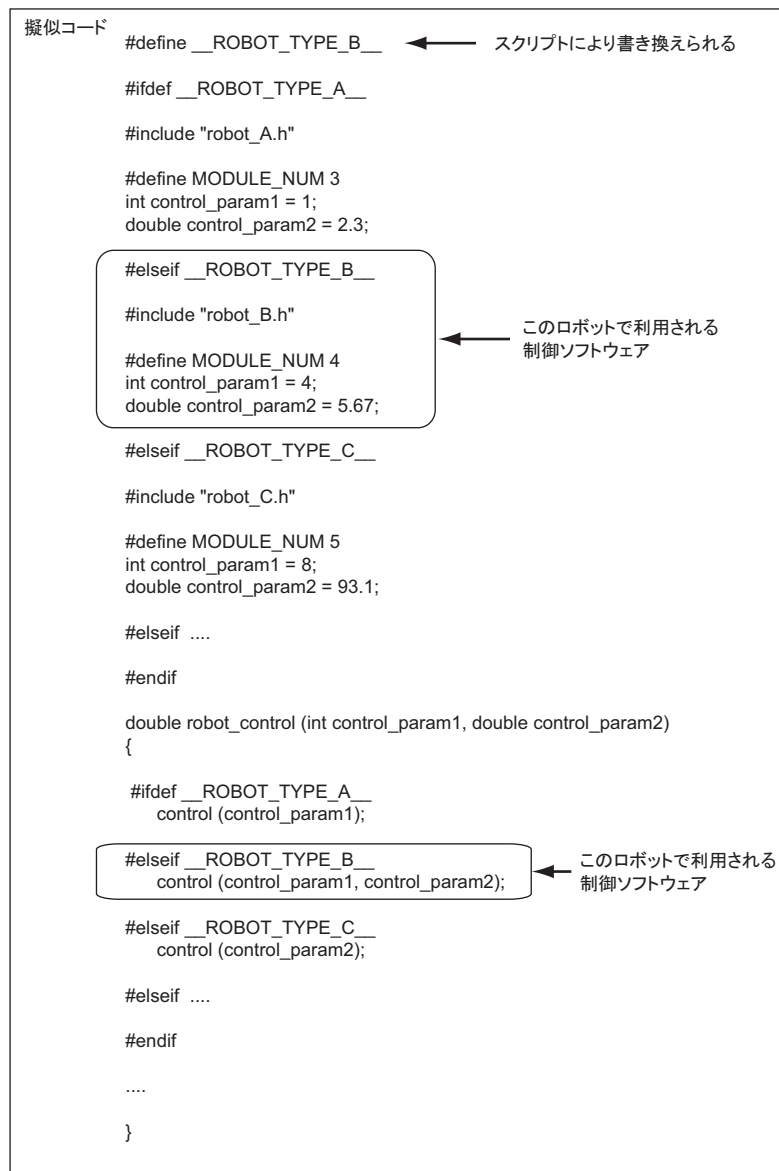


図 6.39: モジュール型ヒューマノイドロボットの制御ソフトウェアの切り替え

2. robot.conf を読み込み，その構成が今まで利用したことがあるかないかを判断するスクリプトを実行
3. 初めての構成と判断した場合は，ヒューマノイドロボットは起動することがなく，ソフトウェアの開発を行う．もしも既に起動したことがある構成なら，制御ソフトウェアを図 6.39 のように切り替える．

6.4 本章のまとめ

本章は、ロボット機能の並列性に着目し、行動計画の並列・分散化を実現する制御アルゴリズムの提案・設計し、その有効性を示すために、プロトタイプロボット P1 の実装・評価について述べた。その結果、プロトタイプロボット P1 が搭載する情報処理システムは、従来のロボットが搭載する情報処理システムより、動作の並列性を抽出した処理を行うことが可能であった。

続いて本節では、プロトタイプロボット P1 で示した成果へ再構成機構を追加するために、モジュール接続ユニット、ソフトウェア機構を提案し、再構成可能なモジュール型ヒューマノイドロボットを実現可能とするシステムアーキテクチャの設計を行った。

次章は、設計したシステムアーキテクチャに基づいた再構成可能なモジュール型ヒューマノイドロボットを開発し、システムアーキテクチャの有効性について検証する。

第7章

プロトタイプロボットの実装と評価実験

本章では，前章で述べた設計に従い，再構成可能なモジュール型ヒューマノイドロボットのプロトタイプロボットの実装を行う．また，(1) ヒューマノイドロボットが搭載する情報処理システムの情報処理能力不足，(2) 拡張性・柔軟性のあるシステム統合化技術の欠如というヒューマノイドロボットが抱えている2つの問題点に対して，本システムアーキテクチャが提案する(1) 行動計画部の並列・分散処理，(2) システム再構成機構が有効であるかどうかプロトタイプロボットを利用した実験を通して検証する．

7.1 再構成可能なモジュール型ヒューマノイドロボット R1

7.1.1 プロトタイプロボット R1 の全体概要

図 7.1 は，システムアーキテクチャの有効性の検証を行うために開発した再構成可能なモジュール型ヒューマノイドロボットのプロトタイプ R1 [Taira *et al.* 05] を示す．プロトタイプロボット R1 は，下記に示す計 5 種類 7 機種のプロトタイプ開発した機能別ロボットから構成する．

- 脳ロボット 1 台
- 移動ロボット 2 台
- 腕ロボット 4 台
- 頭ロボット 2 台

プロトタイプロボット R1 は，機能別ロボットの組み合わせに応じて多少異なるが，およそ高さ 1250mm，幅 800mm，奥行 600mm，重量 30kg，自由度 20 である．ユーザは，プロトタイプロボット R1 を，図 7.1 が示す全ての機能別ロボットから構成するヒューマノイドロボット，図 7.2 が示す脳ロボットと頭ロボットと車輪移動ロボットから構成するヒューマノイドロボット，脳ロボット，頭ロボット，右腕ロボット，左腕ロボットから構成する上

7. プロトタイプロボットの実装と評価実験7.1. 再構成可能なモジュール型ヒューマノイドロボット R1

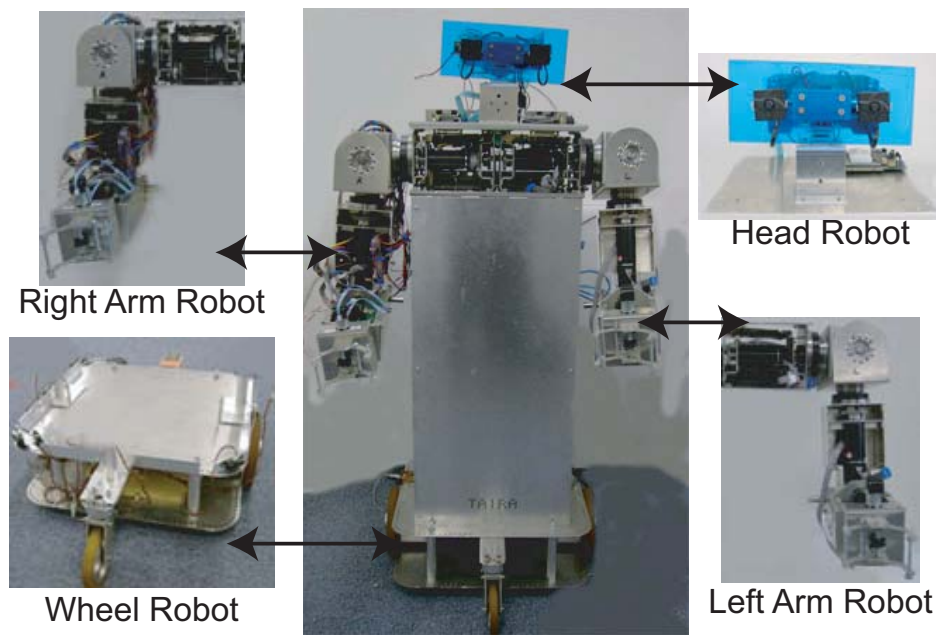


図 7.1: 再構成可能なモジュール型ヒューマノイドロボットのプロトタイプ R1

半身ヒューマノイドロボット，腕ロボット，頭ロボットなどの単体のロボット等，使用用途や使用環境下に応じた機能や形態を備えたロボットとして利用可能である．

7.1.2 プロトタイプロボット R1 のシステム構成

再構成可能なモジュール型ヒューマノイドロボットのシステムアーキテクチャに基づいて開発した各機能別ロボットは，ヒューマノイドロボットの一部の機能を備えるが，ヒューマノイドロボットの一部であることを意識することなく単体の自律ロボットとして利用・開発が可能である．プロトタイプロボット R1 を構成する機能別ロボットは，第 6 章で述べたプロトタイプ P1 と同様に，各機能別ロボットは，制御用プロセッサにはレスポンスプロセッサを，制御用 OS には RT-Frontier を利用している．また，各機能別ロボット間は RL-Net で接続する．以降ではプロトタイプ開発した機能別ロボットについて述べる．

脳ロボット

図 7.3 は，開発した脳ロボットを示す．脳ロボットはシステム全体の行動計画，機能別ロボットの管理を行うための機能別ロボットである．脳ロボットの主な特徴は以下である．

- ソフトウェア構成

- SCA(1msec 周期)

7. プロトタイプロボットの実装と評価実験 7.1. 再構成可能なモジュール型ヒューマノイドロボット R1

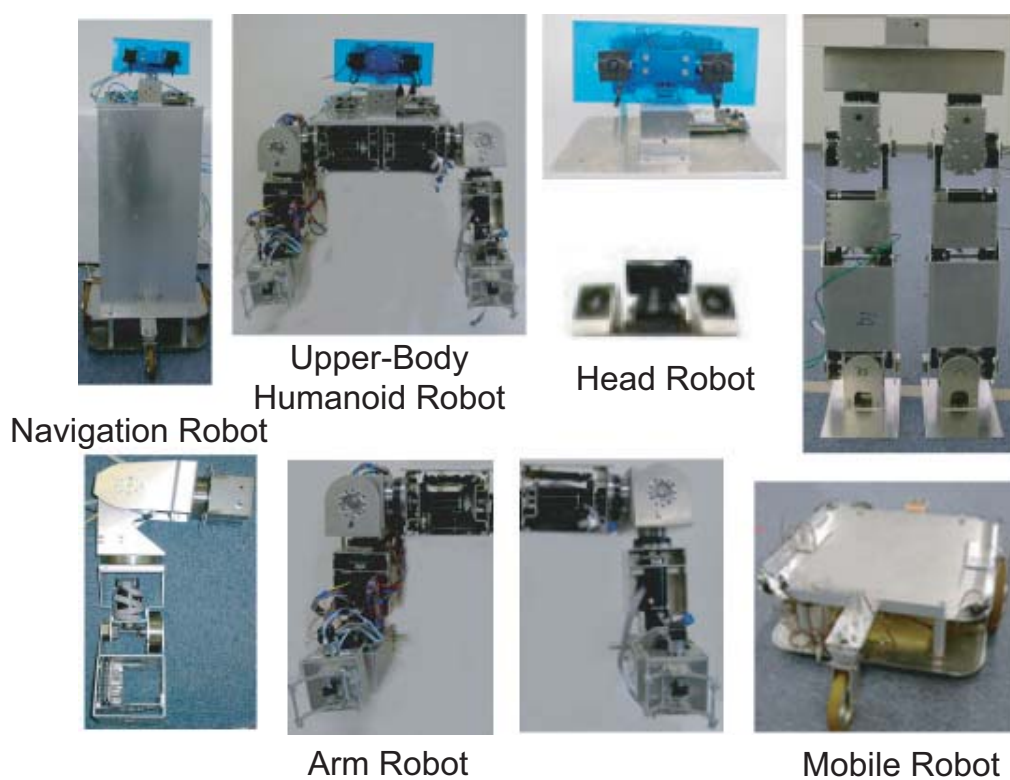


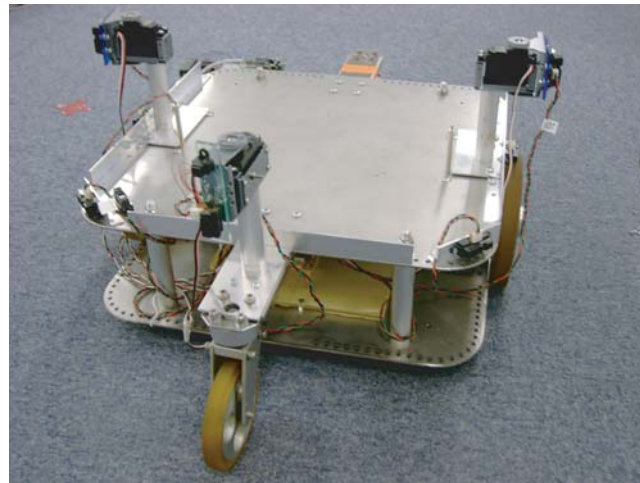
図 7.2: 再構成可能なモジュール型ヒューマノイドロボット



図 7.3: 脳ロボット

- TMA(10msec 周期)
- MPA(100msec 周期)

7. プロトタイプロボットの実装と評価実験7.1. 再構成可能なモジュール型ヒューマノイドロボット R1



Responsive Processor
I/O peripherals
Control Circuit

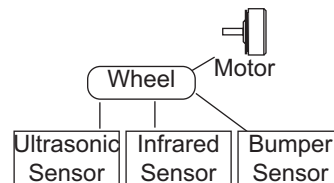
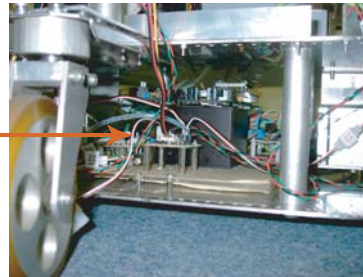


図 7.4: 車輪移動ロボット

車輪移動ロボット

図 7.4 は開発した車輪移動ロボットを示す。車輪移動ロボットは、自身が備えるセンサ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別ロボットである。車輪移動ロボットの主な特徴は以下である。

- 機械構造
左右の各車輪 1 自由度の計 2 自由度
- 電気・電子ハードウェア
内界センサとしてジャイロセンサ (Fiber Optical Gyro)，エンコーダを，外界センサとして超音波センサ，赤外線センサ，接触センサを装備
- ソフトウェア部

7. プロトタイプロボットの実装と評価実験7.1. 再構成可能なモジュール型ヒューマノイドロボット R1

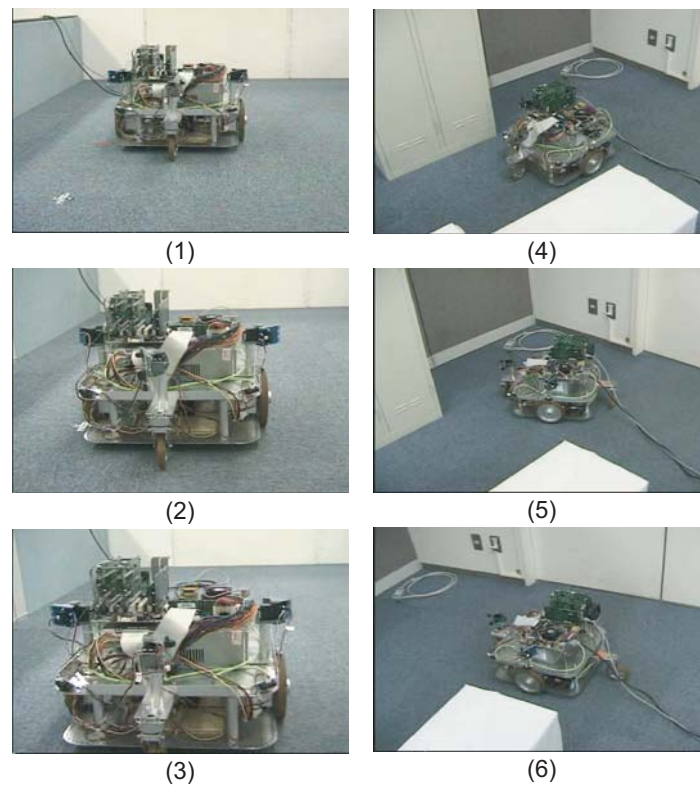


図 7.5: 機能別ロボット単体の行動例：車輪移動ロボット

- 内界センサの値を参照した速度制御 (1msec 周期)
- 内界センサと外界センサの値を参照してカルマンフィルタを用いて自己位置推定 (10msec 周期)
- パスプランニング (100msec 周期)

図 7.5 は車輪移動ロボットを利用したマップ生成，動作計画の実験の様子を示している [山浦 他 05b] [山浦 他 05a] [鎌田 他 04] .

歩行移動ロボット

図 7.6 は開発した歩行移動ロボットを示す．歩行移動ロボットは，歩行機能を提供する機能別ロボットである．歩行移動ロボットの主な特徴は以下である．

- 機械構造
左右の各脚 6 自由度の計 12 自由度
- 電気・電子ハードウェア

7. プロトタイプロボットの実装と評価実験7.1. 再構成可能なモジュール型ヒューマノイドロボット R1

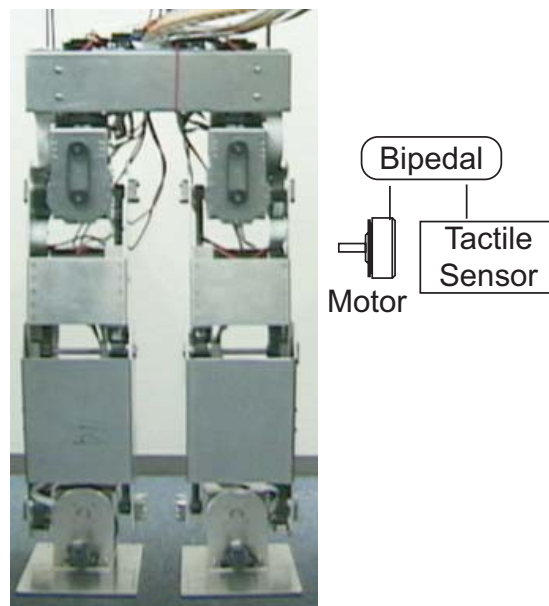


図 7.6: 歩行移動ロボット

内界センサとしてエンコーダを，外界センサとして足の着地面を検出するために足裏にタクトイルセンサを装備

- ソフトウェア部
 - － 内界センサの値を参照した位置制御 (1msec 周期)
 - － 順運動学，逆運動学 (300msec 周期)
 - － パスプランニング (1000msec 周期)

図 7.7 は歩行移動ロボットが歩行アルゴリズムの実験を行っている様子である [鎌田 他 06] [山浦 他 06] .

右腕ロボット

図 7.8 は開発した右腕ロボットを示す．右腕ロボットはロボットのマニピュレータ部分を担う機能別ロボットである．右腕ロボットの主な特徴は以下である．

- 機械構造
 - 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
- 電気・電子ハードウェア
 - 内界センサとしてエンコーダを，外界センサとしてハンド部にタクトイルセンサを装備

7. プロトタイプロボットの実装と評価実験 7.1. 再構成可能なモジュール型ヒューマノイドロボット R1

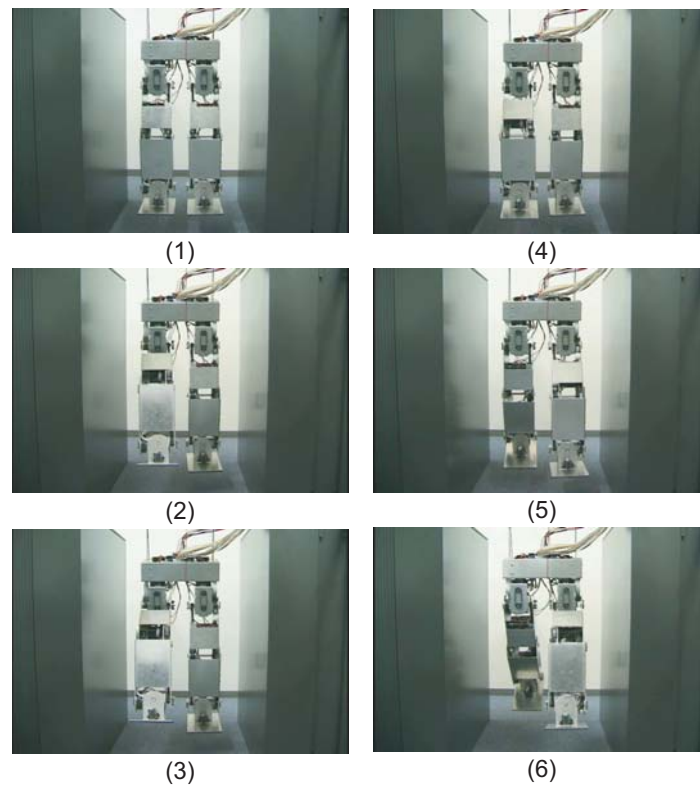


図 7.7: 機能別ロボット単体の行動例：歩行移動ロボット

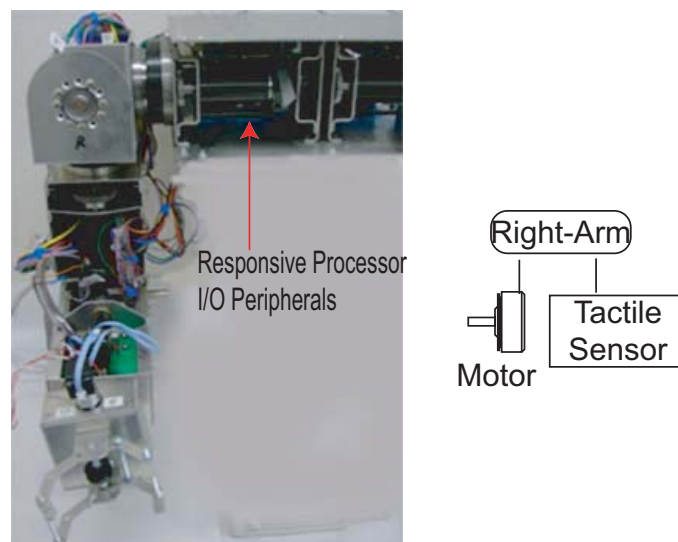


図 7.8: 右腕ロボット

- ソフトウェア部

- 内界センサの値を参照した速度制御 (1msec 周期)

7. プロトタイプロボットの実装と評価実験7.1. 再構成可能なモジュール型ヒューマノイドロボット R1

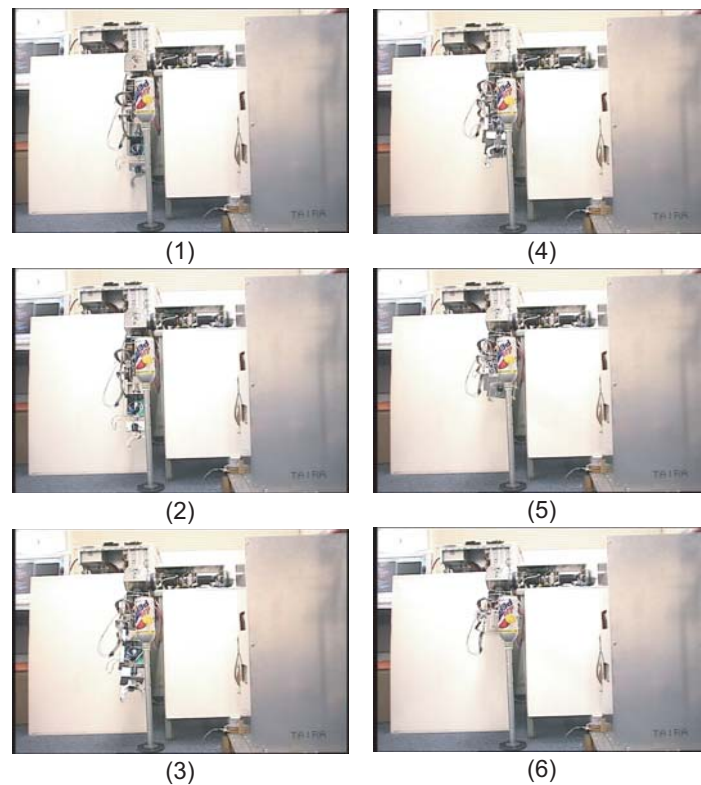


図 7.9: 機能別ロボット単体の行動例：右腕ロボット

- 内界センサと外界センサの値を参照して自己位置推定 (10msec 周期)
- コリジョンフリーなリーチング戦略 (100msec 周期)

図 7.9 は右腕ロボットを用いたリーチング戦略の行動計画実験を行っている様子である [上山 他 06] .

左腕ロボット

右腕ロボットと同様である .

頭ロボット A

図 7.10 は開発した頭ロボット A を示す . 頭ロボット A は , 視覚処理部と聴覚処理部の 2 系統から構成するロボットである . 頭ロボット A の主な特徴は以下である .

- 機械構造
自由度なし

7. プロトタイプロボットの实装と評価実験7.1. 再構成可能なモジュール型ヒューマノイドロボット R1

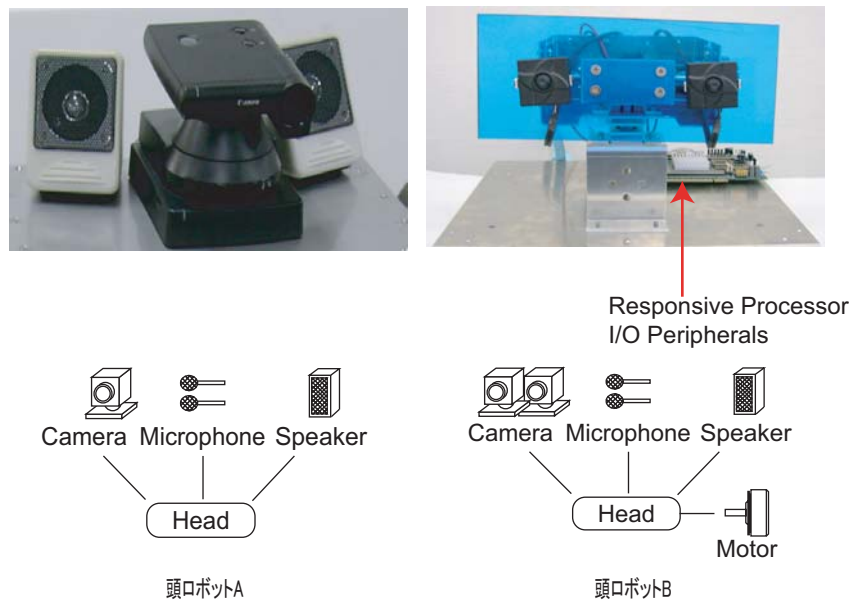


図 7.10: 頭ロボット

- 電気・電子ハードウェア
 - プロトタイプロボット P1 と同様
 - CMOS カメラを 1 つ搭載
- ソフトウェア部
 - プロトタイプロボット P1 と同様

頭ロボット B

頭ロボット B は、頭ロボット A と同様に視覚処理部と聴覚処理部の 2 系統から構成するロボットである。頭ロボット B の主な特徴は以下である。

- 機械構造
 - 首に 2 自由度，目 (カメラ) のパン・チルト動作の計 6 自由度
- 電気・電子ハードウェア
 - プロトタイプロボット P1 と同様
 - CMOS カメラを 2 つ搭載
- ソフトウェア部

7. プロトタイプロボットの実装と評価実験 7.1. 再構成可能なモジュール型ヒューマノイドロボット R1

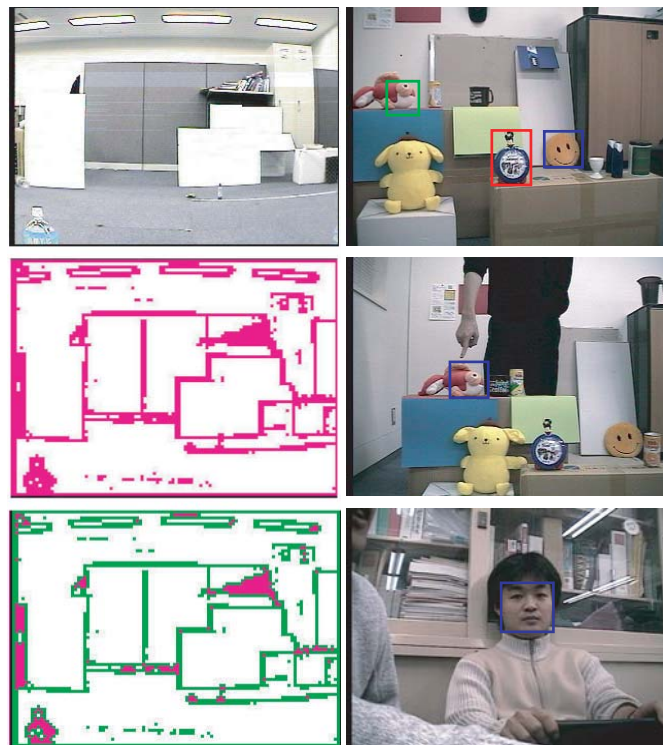


図 7.11: 機能別ロボット単体の行動例：頭ロボット B

- プロトタイプロボット P1 と同様
- 3次元距離測定機能

図 7.11 は頭ロボット B を用いた環境認識システムに関する研究を行っている様子を示している [鎌田 他 05] .

7.1.3 プロトタイプロボット R1 の制御ソフトウェア

図 7.12 は、プロトタイプロボット R1 の階層化した制御ソフトウェアを示す。脳ロボットの TMA, SCA は、各機能別ロボットの FCA, DSA とそれぞれ RL-Net 上で接続し、各機能別ロボットが提供する機能を利用して Global Control を実現する。各機能別ロボットが Local Control にて提供する機能は 7.1.2 節で述べた通りである。プロトタイプロボット R1 の情報処理システムは、機能別ロボット間が RL-Net 上で接続する Cooperative Control により、下記に示す機能も提供する。これらの Cooperative Control は Global Control とは別の制御ループで形成され、Global Control とは並列で動作する。

- 車輪移動ロボットの FCA, DSA と頭ロボットの FCA, DSA による障害物回避機能

7. プロトタイプロボットの実装と評価実験 7.1. 再構成可能なモジュール型ヒューマノイドロボット R1

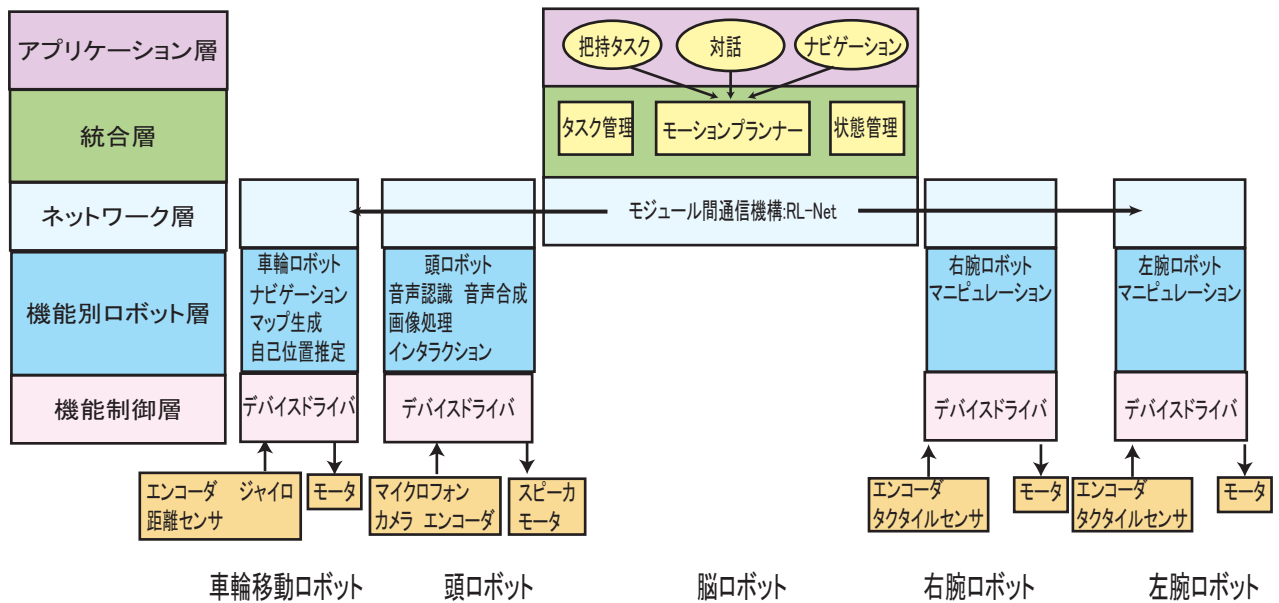


図 7.12: プロトタイプロボット R1 のソフトウェア構成

- 右(左)腕ロボットの FCA, DSA と頭ロボットの FCA, DSA によるビジュアルフィードバック機能
- 右腕ロボットの FCA, DSA と左腕ロボットの FCA, DSA によるコリジョンフリーな行動計画

7.1.4 モジュール接続ユニット

図 7.13 は、プロトタイプ開発したモジュール接続ユニットを示す。各モジュール接続ユニットは、設計に従って、機械構造、通信インターフェース(レスポンスリンク)、電源線から構成している。図 7.13 のモジュール接続ユニット A は、図 7.14 に示すように各機能別ロボットの上に取り付け、モジュール接続ユニット B は各機能別ロボットの下部に取り付ける。

図 7.14 は、ユーザがモジュール接続ユニットを利用して機能別ロボットを再構成している様子を示している。機能別ロボット間の再構成に必要な手順は以下である。機能別ロボットを分離する際は逆の手順である。図 7.14 中の番号は、以下の番頭と対応している。

1. 機能別ロボットとコネクタを接続 (ネジ 10 個で接続可能。モジュール接続ユニットを機能別ロボットから簡単に着脱することが可能)

7. プロトタイプロボットの实装と評価実験 7.2. プロトタイプロボット R1 を利用した実験による評価・考察

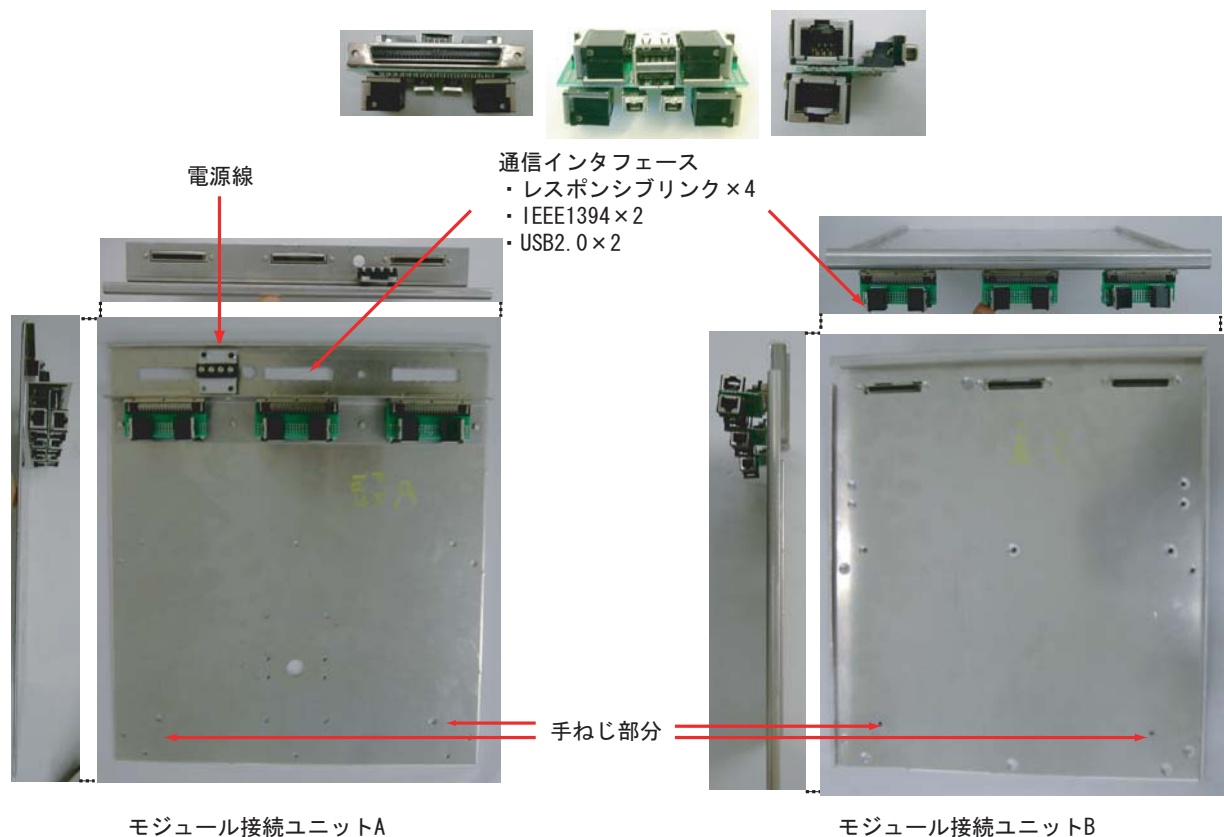


図 7.13: モジュール接続ユニット

2. モジュールの持つコネクタ B をシステムのコネクタ A に重ね、奥まではめ込む。この時に機械的にモジュール接続ユニット A と B は固定される
3. 手ネジを 2 箇所接続
4. 通信インターフェース (レスポンスリンク)、電源線を接続

7.2 プロトタイプロボット R1 を利用した実験による評価・考察

本節では、まず始めにプロトタイプロボット R1 を利用した実験について述べる。続いて、その実験の結果に基づいて、(1) ヒューマノイドロボットが搭載する情報処理システムの情報処理能力不足、(2) 拡張性・柔軟性のあるシステム統合化技術の欠如というヒューマノイドロボットが抱えている 2 つの問題点に対して、本システムアーキテクチャが提案する (1) 行動計画部の並列・分散処理、(2) システム再構成機構が有効であるかどうかを検証する。

7. プロトタイプロボットの实装と評価実験 7.2. プロトタイプロボット R1 を利用した実験による評価・考察

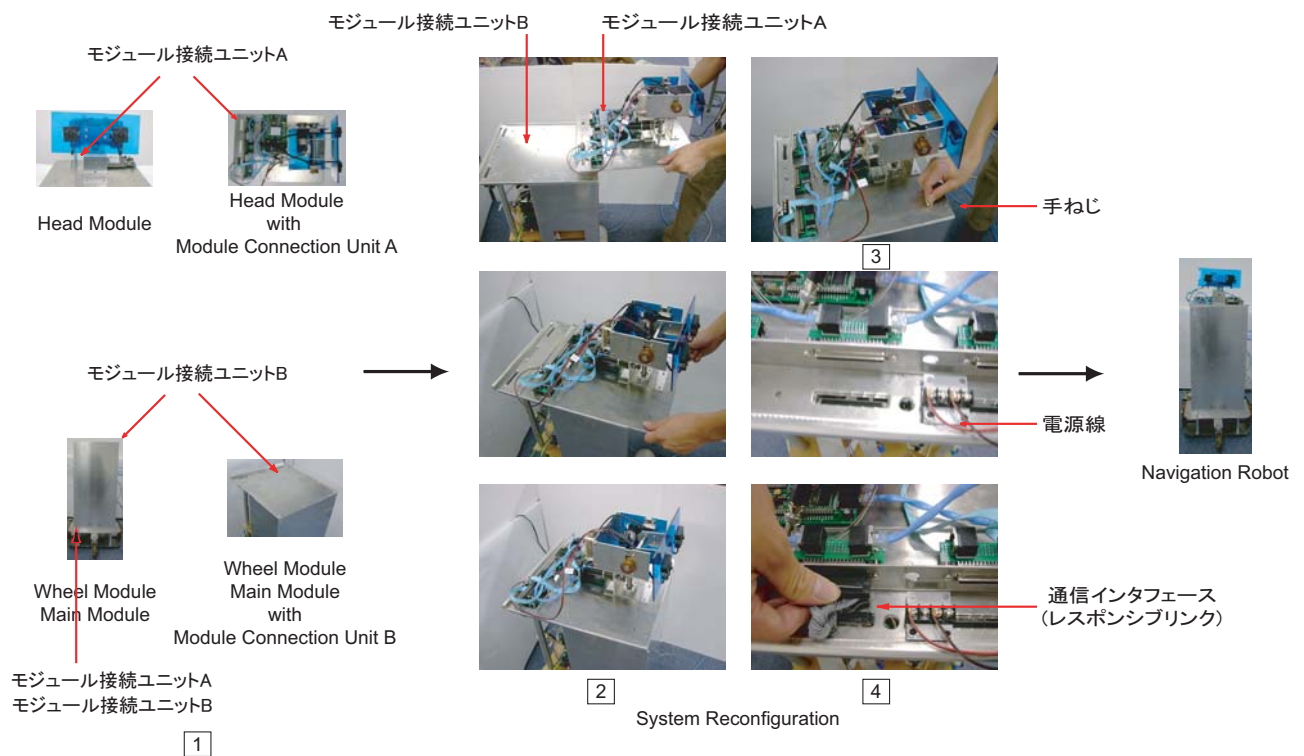


図 7.14: モジュール接続ユニットを用いた機能別ロボットの再構成

7.2.1 プロトタイプロボット R1 を利用した実験

モジュール型ヒューマノイドロボットの3つの動作例を挙げる。

1つ目の動作例は、人間とのインタラクションの例である握手行動である。図 7.15 は、この動作を行う際に R1 が搭載する情報処理システムを示している。

図 7.16 は、モジュール型ヒューマノイドロボットが握手動作を行っている様子を示している。図 7.16 中の番号は、下記の番号と対応している。

1. ユーザが頭ロボットへ握手したいとタスクを音声入力
2. 脳ロボットは頭ロボットの音声認識結果から行動計画を実行し、車輪移動ロボットへ頭ロボットが音源定位により取得したユーザがいる向きへ回転するようにタスク要求。また同時に、頭ロボットへユーザの顔検出を行うようにタスク要求
3. タスク要求を受けた車輪移動ロボットは、回転運動を開始。同時に、頭ロボットはユーザの顔検出を実行
4. 頭ロボットがユーザを発見後、脳ロボットは、頭ロボットへは人間とのコミュニケーションを、車輪移動ロボットには移動停止動作のタスク要求

7. プロトタイプロボットの実装と評価実験 7.2. プロトタイプロボット R1 を利用した実験による評価・考察

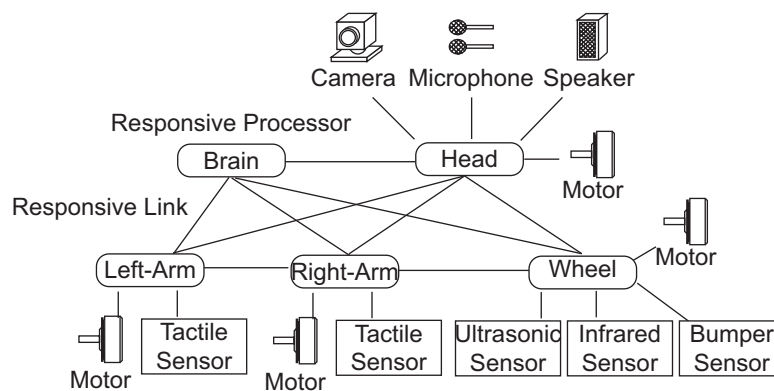
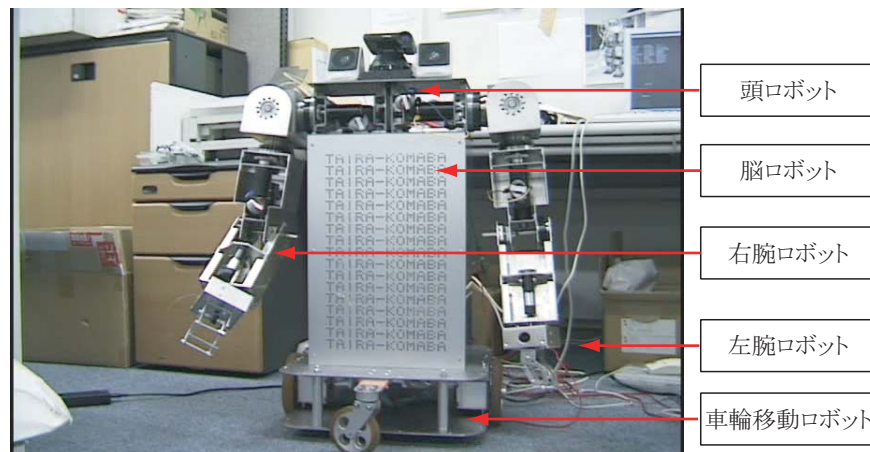


図 7.15: モジュール型ヒューマノイドロボットの情報処理システム

5. 脳ロボットは腕ロボットへ握手行動をタスク要求

6. 腕ロボットは握手行動を実行

2つ目の動作例は、人間とのインタラクションの例である物の受け渡し行動である。図 7.17 は、この動作を行う際に R1 が搭載する情報処理システムを示している。

図 7.18 は、モジュール型ヒューマノイドロボットが物の受け渡し行動を行っている様子を示している。図 7.18 中の番号は、下記の番号と対応している。

1. ユーザが頭ロボットへ対象物 (缶) の受け渡したいとタスクを音声入力
2. 脳ロボットは頭ロボットの音声認識結果から行動計画を実行し、車輪移動ロボットへ頭ロボットが音源定位により取得したユーザがいる向きへ回転するようにタスク要求。また同時に、ユーザが持つ缶の検出を行うようにタスク要求
3. 頭ロボットが缶を発見したら、脳ロボットは腕ロボットへユーザから缶を受け渡してもらえ位置まで移動するようにタスク要求

7. プロトタイプロボットの实装と評価実験 7.2. プロトタイプロボット R1 を利用した実験による評価・考察

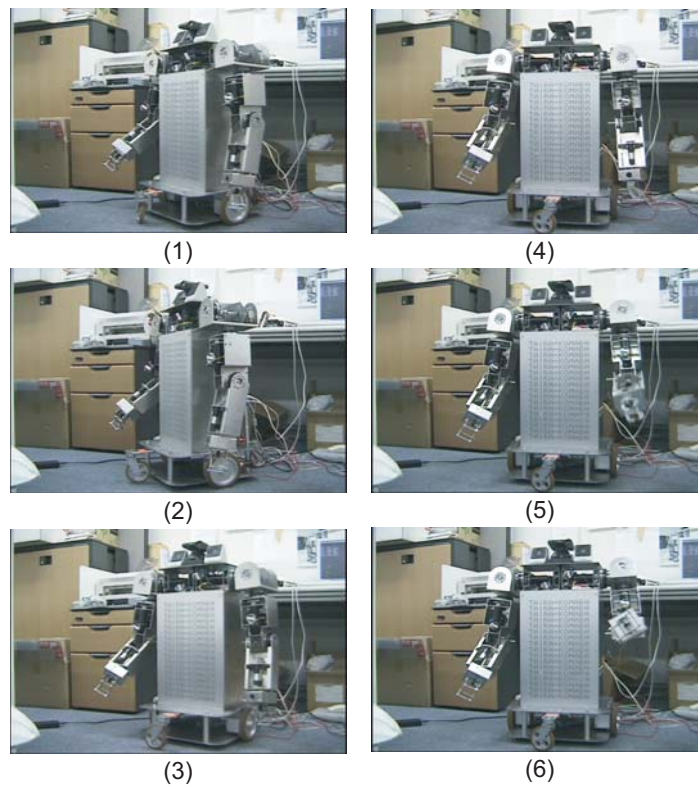


図 7.16: 握手動作

4. 腕ロボットが移動後，脳ロボットは頭ロボットへユーザに対して「缶を渡してください」と音声発話するタスクを要求
5. 腕ロボットはユーザが缶を渡してくれるのを認識し，缶の把持を実行
6. 把持後，缶を受け取る

3つ目の動作例は，自律して缶を把持する行動である．図 7.19 は，この動作を行う際に R1 が搭載する情報処理システムを示している．

図 7.20 は，モジュール型ヒューマノイドロボットが自律して缶を把持する動作を行っている様子を示している．図 7.20 中の番号は下記の番号と対応している．

1. ユーザが頭ロボットへ対象物 (缶) を探し，把持するタスクを音声入力
2. 脳ロボットは頭ロボットの音声認識結果から行動計画を実行し，頭ロボットへ脳ロボットが対象物を探し，3次元距離を測定するタスクを要求
3. 頭ロボットが対象物を発見し3次元距離を測定後，脳ロボットは，車輪移動ロボットへ対象物を把持可能な位置まで移動するようにタスク要求を，右腕ロボットへは，車輪移動ロボットが移動する位置から実際に対象物を把持できる腕の制御をタスク要求

7. プロトタイプロボットの実装と評価実験 7.2. プロトタイプロボット R1 を利用した実験による評価・考察

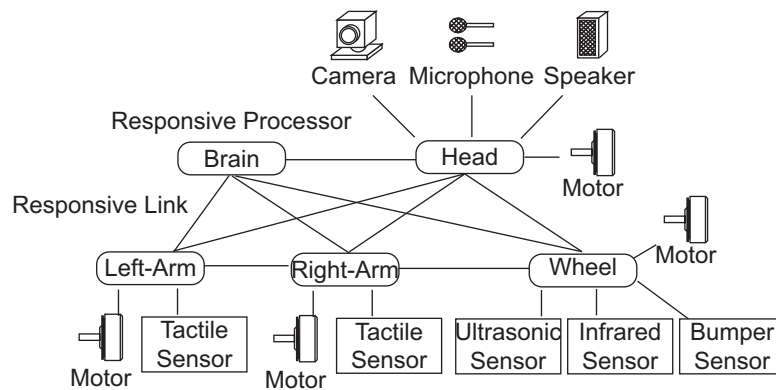
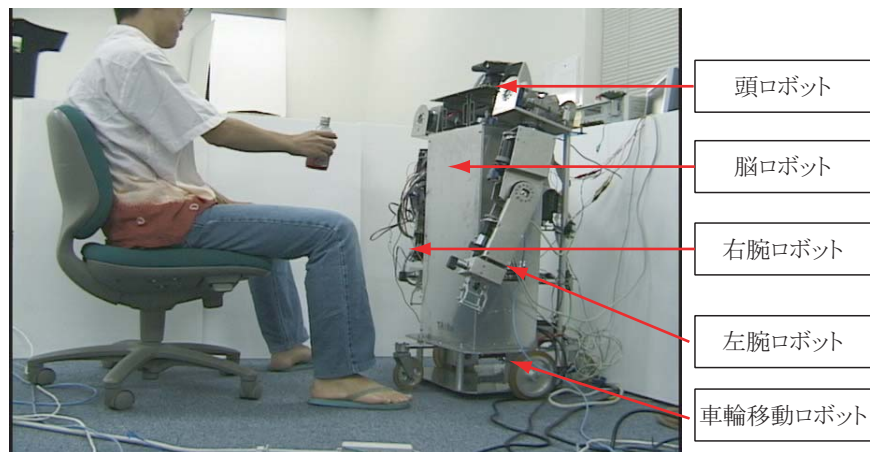


図 7.17: モジュール型ヒューマノイドロボットの情報処理システム

4. 車輪移動ロボットと右腕ロボットはタスク要求に応じて行動計画を行い，タスクを実行
5. 車輪移動ロボットが右腕ロボットが対象物を把持可能な位置まで移動して停止
6. 右腕ロボットが対象部とを頭ロボットと協調して把持タスクを実行

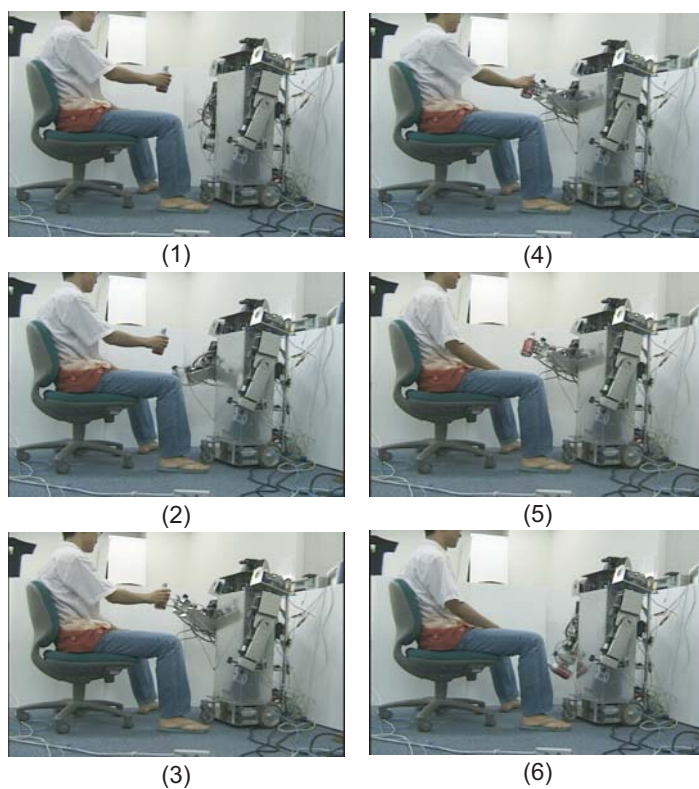


図 7.18: 物の受け渡し行動

7. プロトタイプロボットの实装と評価実験 7.2. プロトタイプロボット R1 を利用した実験による評価・考察

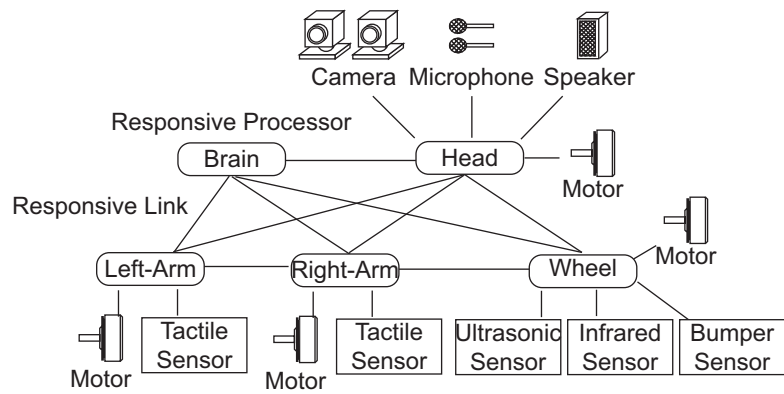
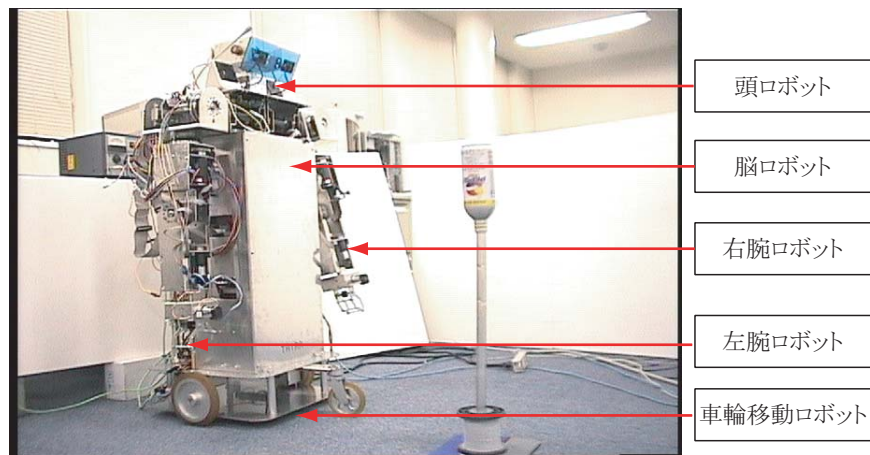


図 7.19: モジュール型ヒューマノイドロボットの情報処理システム

7. プロトタイプロボットの実装と評価実験 7.2. プロトタイプロボット R1 を利用した実験による評価・考察

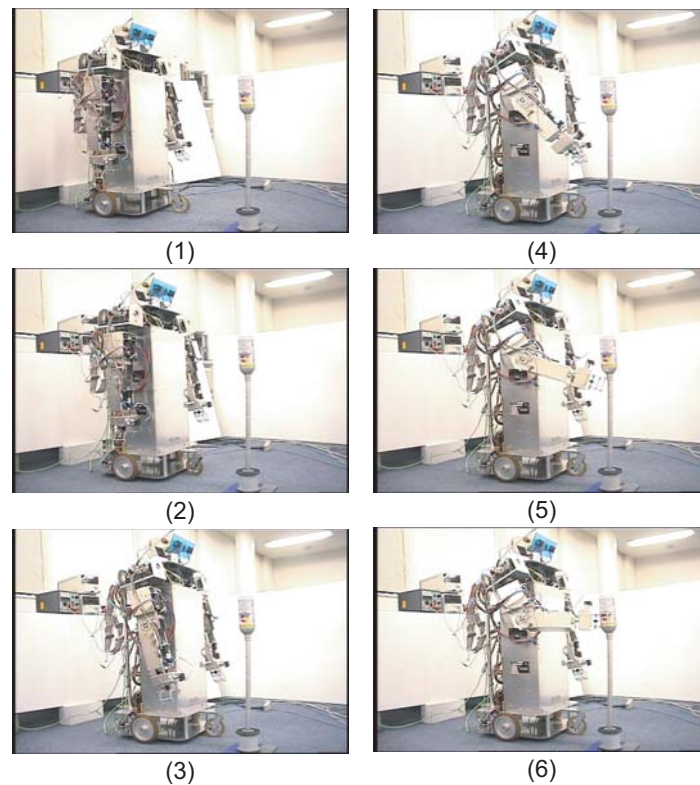


図 7.20: 自律行動

7.2.2 行動計画部の並列・分散処理化に関する評価

図 7.21 は、図 7.18 に示したタスクを処理している際の情報処理システムの動作を示している。図 7.21 中に示す番号は、先に述べたアプリケーションの動作を示した番号と図 7.19 中の番号に対応している。

この図 7.21 から、プロトタイプロボット R1 が搭載する情報処理システムは、各機能別ロボットが別々に動作することで、ロボット全体として複数の行動計画を同時並列して処理していることが明らかである。図 7.22 は、本システムアーキテクチャを比較検討するために、車輪移動ロボットに接続していたアクチュエータ、センサなどの電気・電子ハードウェアと制御ソフトウェアを脳ロボットへ移植した R1 の集中制御方式の情報処理システムを示している。

図 7.22 に示す情報処理システムは、関連研究で述べた HRP(図 3.7) や ASIMO(図 3.10) の情報処理システムと類似している。図 7.23 は、図 7.22 に示した集中制御化した R1 の情報処理システムが図 7.18 に示したアプリケーションを実行した時の情報処理システムの制御フローを示している。

本システムアーキテクチャに基づいた制御フロー(図 7.21) と従来の制御手法に基づいた

7. プロトタイプロボットの实装と評価実験 7.2. プロトタイプロボット R1 を利用した実験による評価・考察

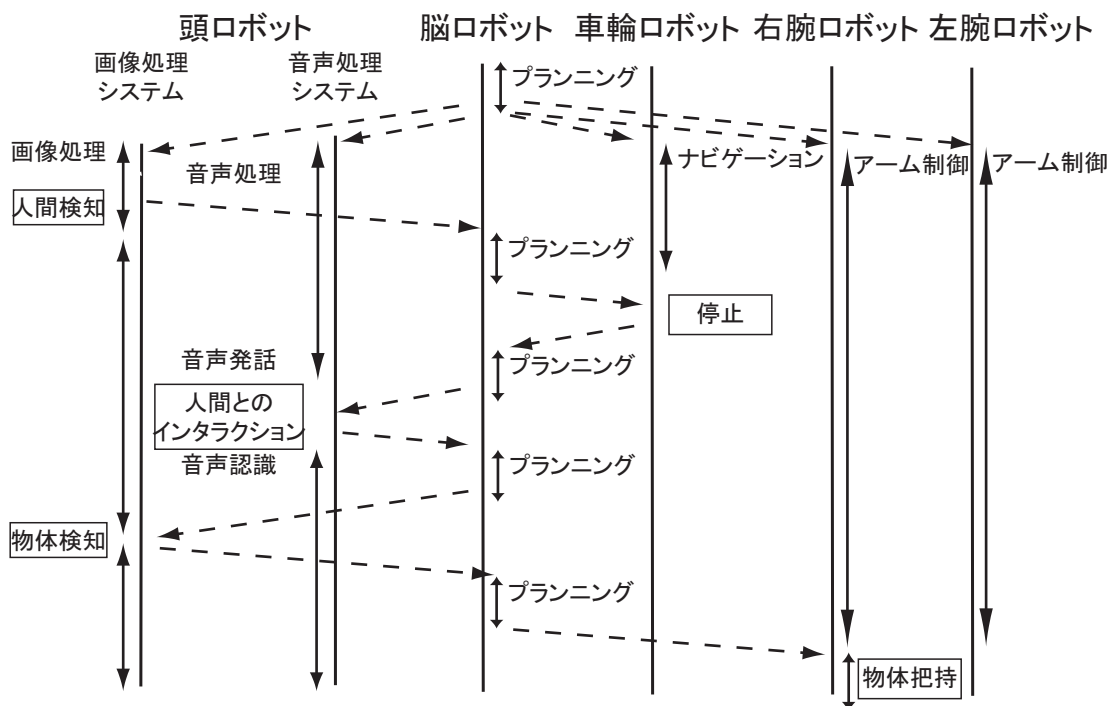


図 7.21: プロトタイプロボット R1 が搭載する情報処理システムの制御フロー

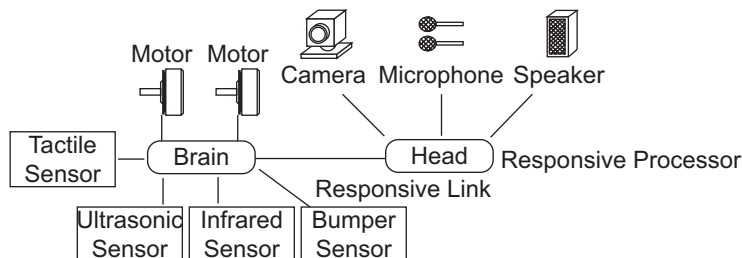


図 7.22: 集中制御化したプロトタイプロボット R1 の情報処理システム

制御フロー (図 7.23) を比較すると、脳ロボットの動作計画部の処理量が大きく異なっている。具体的には、従来の制御手法に基づいた制御フローは、図 2.7 に示した処理を順番に実行しているだけである。しかし実際には、図 7.21 に示すプロトタイプロボット R1 の制御フローが明らかにしたように、その動作計画部は並列処理可能であったことが検証できた。

次に、制御フローを用いた定性的な評価から明らかになった本システムアーキテクチャの有効性を定量的に解析する。表 7.1 は、図 7.22 時の主な処理の制御周期と実行時間を示している。ただし、機能別ロボット間のデータ共有、タスク要求に関しては、最悪通信遅延時間を示している。

表 7.2 は、図 7.23 に示す制御フロー内の主な処理の制御周期と実行時間を示している。

7. プロトタイプロボットの実装と評価実験 7.2. プロトタイプロボット R1 を利用した実験による評価・考察

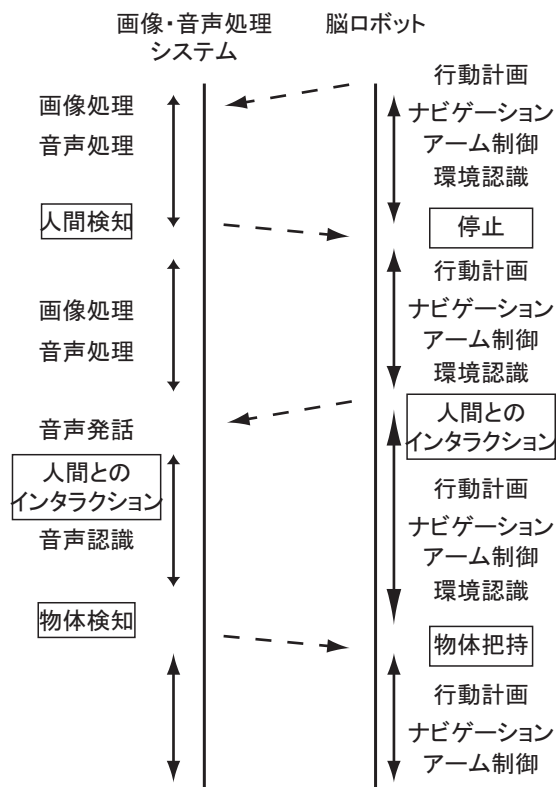


図 7.23: 従来の制御手法での制御フロー

ただし，音声処理と機能別ロボット間のデータ共有，タスク要求に関しては，最悪通信遅延時間を示している．

表 7.1 と表 7.2 を比較すると，表 7.1 に示すモジュール型ヒューマノイドロボットの処理の方が，複数の処理を並列処理できるので，各実行時間を小さく抑えることができ，その結果として各制御周期を小さく設定することができる．

7.2.3 拡張性・柔軟性に関する評価

図 7.24 はモジュール型ヒューマノイドロボットの利点を示している．本プロトタイプ開発では，ヒューマノイドロボットのアプリケーションのために，表 7.3 に示す特徴を持つ頭ロボット A と頭ロボット B の 2 種類の頭ロボットを開発した．

図 7.24 の上図は頭ロボット A を搭載するモジュール型ヒューマノイドロボットを，下図は頭ロボット B を搭載するモジュール型ヒューマノイドロボットを示している．頭ロボット A を搭載するモジュール型ヒューマノイドロボットは，図 7.24 中の (2)，(3) で示すように，ロボットはユーザが示す対象物の 3 次元距離を測定できないため，図 7.18 に示したアプリケーションと同様にユーザの近くまでしか手を伸ばすことができない．そのためユー

7. プロトタイプロボットの实装と評価実験 7.2. プロトタイプロボット R1 を利用した実験による評価・考察

表 7.1: プロトタイプロボット R1 の制御

機能別ロボット	タスク	制御周期	実行時間
脳ロボット	SCA	1msec	100 μ sec
脳ロボット	TMA	100msec	2msec
脳ロボット	MPA	100msec	10msec
車輪移動ロボット	速度制御	1msec	100 μ sec
車輪移動ロボット	自己位置推定	10msec	2msec
車輪移動ロボット	パスプランニング	100msec	10msec
右(左)腕ロボット	速度制御	1msec	100 μ sec
右(左)腕ロボット	自己位置推定	10msec	2msec
右(左)腕ロボット	パスプランニング	100msec	10msec
頭ロボット	顔認識	-	300msec
頭ロボット	3次元距離測定	33msec	10msec
頭ロボット	音声認識	-	1sec
頭ロボット	音声合成	-	1sec
頭ロボット	音源定位	-	1sec
脳ロボットのSCAと車輪移動ロボットのDSA	データ共有	10msec	105 μ sec
脳ロボットのTMAと車輪移動ロボットのFCA	タスク要求	100msec	37 μ sec
脳ロボットのSCAと右(左)腕ロボットのDSA	データ共有	10msec	105 μ sec
脳ロボットのTMAと右(左)腕ロボットのFCA	タスク要求	100msec	37 μ sec
脳ロボットのSCAと頭ロボットのDSA	データ共有	33msec	105 μ sec
脳ロボットのTMAと頭ロボットのFCA	タスク要求	100msec	37 μ sec
車輪移動ロボットのDSAと頭ロボットのDSA	データ共有	33msec	105 μ sec
車輪移動ロボットのFCAと頭ロボットのFCA	タスク要求	100msec	37 μ sec
右(左)腕ロボットのDSAと頭ロボットのDSA	データ共有	33msec	105 μ sec
右(左)腕ロボットのFCAと頭ロボットのFCA	タスク要求	100msec	37 μ sec
右腕ロボットのDSAと左腕ロボットのDSA	データ共有	10msec	105 μ sec
右腕ロボットのFCAと左腕ロボットのFCA	タスク要求	100msec	37 μ sec

ザから対象物を受け渡してもらうことになる一方、頭ロボット B を搭載するモジュール型ヒューマノイドロボットは、(2)、(3) で示すように、ロボットはユーザが示す対象物の 3 次元距離を測定できるため、図 7.20 に示したアプリケーションと同様にユーザの持つ対処物まで自律して手を伸ばすことができる。そのためユーザから対象物を受け取ることが可能となる。このように、人間から受け渡してもらうことを前提としたタスクには頭ロボット A を利用したモジュール型ヒューマノイドロボットが、自律性を求められるタスクには頭ロボット B を利用したモジュール型ヒューマノイドロボットが適している。

7. プロトタイプロボットの实装と評価実験 7.2. プロトタイプロボット R1 を利用した実験による評価・考察

表 7.2: プロトタイプロボット R1 の制御 (集中制御時)

機能別ロボット	タスク	制御周期	実行時間
脳ロボット	MPA	120msec	40msec
脳ロボット	車輪の速度制御	4msec	100 μ sec
脳ロボット	車輪の自己位置推定	40msec	2msec
脳ロボット	腕の速度制御	4msec	100 μ sec
脳ロボット	自己位置推定	40msec	2msec
頭ロボット	顔認識	-	300sec
頭ロボット	3次元距離測定	33msec	10msec
頭ロボット	音声認識	-	1sec
頭ロボット	音声合成	-	1sec
頭ロボット	音源定位	-	1sec
脳ロボットと頭ロボット	タスク・データ	120msec	37 μ sec

表 7.3: 頭ロボットの比較

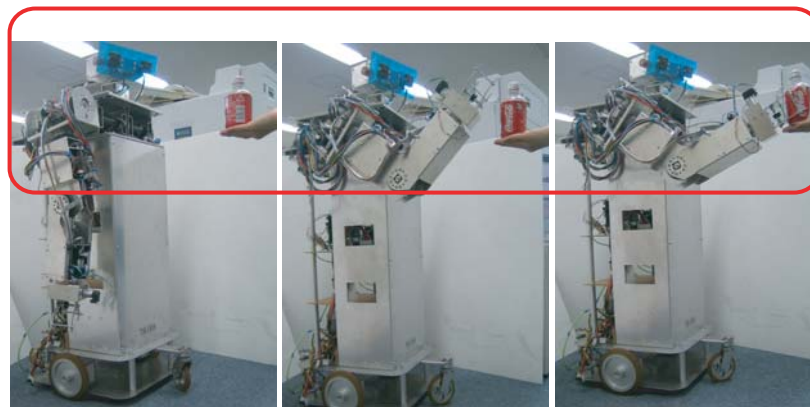
	頭ロボット A	頭ロボット B
DOFs	2	6
機能	2次元画像処理	3次元画像処理
自律性	低い	高い
計算量	少ない	多い
処理速度	速い	遅い
消費電力	少ない	大きい
搭載スペース	小さい	大きい
重量	軽い	重い

常に頭ロボット B を利用したモジュール型ヒューマノイドロボットがよいのではないかと思うが、頭ロボット B を利用したモジュール型ヒューマノイドロボットは、3次元距離測定や視野拡大が実現できるが、頭ロボット A と比較して計算量が多くなる、重量が重くなるなどの欠点もある。アプリケーションによっては3次元距離が必要になる場合は仕方がないが、3次元処理を必要としないアプリケーションであれば、消費電力が少なく、軽量である頭ロボット A を利用する方がよい。このようなトレードオフはロボットには常に生じることである。図 7.25 は移動方法に関するトレードオフを示している。車輪移動ロボットで乗り越えられる程度の段差がある空間では、移動速度、消費電力等を考慮して車輪移動ロボットの方が最適であるが、車輪移動ロボットでは、段差を乗り越えることが不可能で

7. プロトタイプロボットの実装と評価実験 7.2. プロトタイプロボット R1 を利用した実験による評価・考察



(1) (2) (3)
頭ロボットAを搭載するモジュール型ヒューマノイドロボット



(1) (2) (3)
頭ロボットBを搭載するモジュール型ヒューマノイドロボット

図 7.24: 再構成可能なモジュール型ヒューマノイドロボットの利点:頭ロボット

ある。一方、歩行移動ロボットでは、段差がある空間でも動作できるが、車輪移動ロボットと比較して、移動速度、消費電力、安定の面で劣っている。モジュール型ヒューマノイドロボットでは、このようなトレードオフに対しても、ユーザの使用用途や利用環境下に応じた機能別ロボットを利用することで問題を解決することができる。

従来のヒューマノイドロボットが、このようなトレードオフを解決するためには、頭部、移動部の再開発のみではなく、機械構造、電気・電子ハードウェア、ソフトウェアを含むシステム全体を開発し直さなければならない場合が多い。しかし本モジュール型ヒューマノイドロボットは、図 7.24 に示すように機能別ロボットを取り換えるのみでシステムの機能や形状を変更することができた。ヒューマノイドロボットの再構成機構を実現するために開発したモジュール接続ユニットは、図 7.13 に示した A, B とともに約 0.5Kg であった。図 7.1

7. プロトタイプロボットの実装と評価実験 7.2. プロトタイプロボット R1 を利用した実験による評価・考察

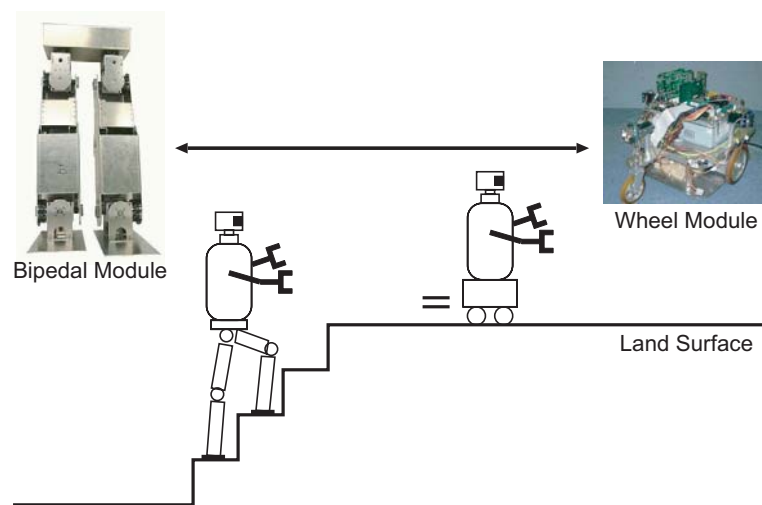


図 7.25: 再構成可能なモジュール型ヒューマノイドロボットの利点:移動ロボット

に示したプロトタイプロボット R1 は、機能別ロボットを接続するためにモジュール接続ユニット A, B それぞれ 3 個の合計 6 個利用することになる。つまり、モジュール接続ユニットの重量は 3Kg であり、プロトタイプロボット R1 全体の重量である 30Kg の 10% である。ロボットの重量が増加することは、ロボットの動作を遅くする、稼動範囲を狭くするなど、ロボットの動作に対して欠点を招いてしまう。今回のプロトタイプ実装には、材料としてアルミニウム (比重 $2.7g/cm^3$) を利用したが、マグネシウム (比重 $1.82g/cm^3$)、FRP (Fiber Reinforced Plastics) (比重 $1.8g/cm^3$)、炭素繊維 (比重 $1.6g/cm^3$) など、アルミニウムより軽くて剛性のある様々な存在する。これらを利用して実装するだけでも 3.0Kg を 2.0Kg に抑えることができる。今後はこのように最適な材料を利用すること、より軽量のモジュール接続ユニットを設計することが重要になるが、3.0Kg の増加により動作に制限がかかるのと引き換えに、ロボットが再構成機構を備えられることは、ロボットのユーザ・開発者にとって有益であると本実験の結果から判断できると考えられる。

ロボットの再構成機構を実現するためには、各機能別ロボット間のトポロジを変更する度に制御アルゴリズムがその影響を受けるようでは、再構成機構の利点が薄れてしまう。そこで本システムアーキテクチャでは、この問題点を解決するためにモジュール間通信機構 RL-Net を提唱している。プロトタイプロボット R1 を利用して RL-Net の評価を実施する。図 7.26 は、プロトタイプロボット R1 の情報処理システム間のネットワークトポロジを図 7.15 から変更した構成を示している。表 7.4 は、図 7.26 に示した情報処理システムで図 7.16、図 7.18、図 7.20 に示すアプリケーションを実行した際の制御周期と通信遅延時間を示している。

表 7.1 と表 7.4 を比較すると、ネットワークのホップ数に応じて通信遅延時間が式 6.2 か

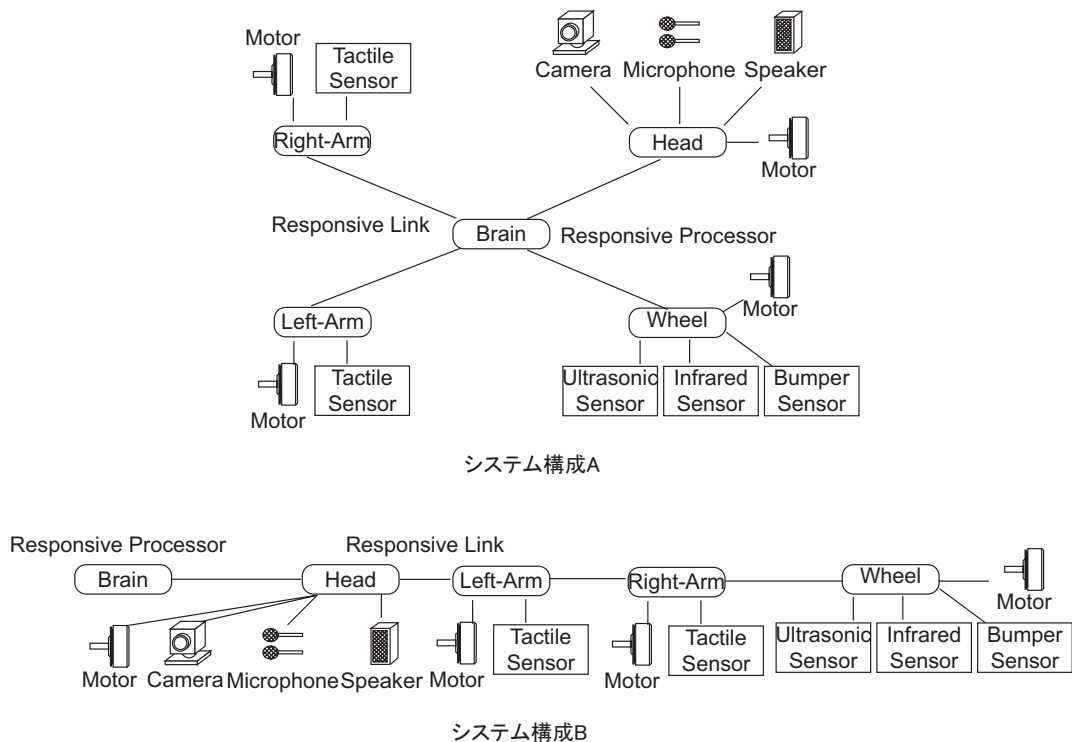


図 7.26: 情報処理システムの構成図

ら式 6.5 に従って増加している．従来の分散制御システムでは，実際に動作させるまでこの通信遅延時間を見積もることは難しいことが欠点として挙げられているが，本システムアーキテクチャでは，RL-NetSim を利用することであらかじめ通信遅延時間を見積もることができた．

7.3 本章のまとめ

本節では，本研究が提案する再構成可能なモジュール型ヒューマノイドロボットを実現するシステムアーキテクチャに従って開発したプロトタイプロボット R1 を利用して，(1) ヒューマノイドロボットが搭載する情報処理システムの情報処理能力不足，(2) 拡張性・柔軟性のあるシステム統合化技術の欠如というヒューマノイドロボットが抱えている 2 つの問題点に対して，本システムアーキテクチャが提案する (1) 行動計画部の並列・分散処理，(2) システム再構成機構が有効であるかどうかプロトタイプロボットを利用した実験を通して検証した．

問題点 (1) に対して，プロトタイプロボット R1 は，既存のヒューマノイドロボットと比

表 7.4: ネットワークトポロジに関する評価

構成	機能別ロボット	タスク	周期	実行時間
A	脳ロボットのSCAと車輪移動ロボットのDSA	データ共有	10msec	105 μ sec
A	脳ロボットのTMAと車輪移動ロボットのFCA	タスク要求	100msec	37 μ sec
A	脳ロボットのSCAと右(左)腕ロボットのDSA	データ共有	10msec	105 μ sec
A	脳ロボットのTMAと右(左)腕ロボットのFCA	タスク要求	100msec	37 μ sec
A	脳ロボットのSCAと頭ロボットのDSA	データ共有	33msec	105 μ sec
A	脳ロボットのTMAと頭ロボットのFCA	タスク要求	100msec	37 μ sec
A	車輪移動ロボットのDSAと頭ロボットのDSA	データ共有	33msec	109 μ sec
A	車輪移動ロボットのFCAと頭ロボットのFCA	タスク要求	100msec	41 μ sec
A	右(左)腕ロボットのDSAと頭ロボットのDSA	データ共有	33msec	109 μ sec
A	右(左)腕ロボットのFCAと頭ロボットのFCA	タスク要求	100msec	41 μ sec
A	右腕ロボットのDSAと左腕ロボットのDSA	データ共有	10msec	109 μ sec
A	右腕ロボットのFCAと左腕ロボットのFCA	タスク要求	100msec	41 μ sec
B	脳ロボットのSCAと車輪移動ロボットのDSA	データ共有	10msec	117 μ sec
B	脳ロボットのTMAと車輪移動ロボットのFCA	タスク要求	100msec	49 μ sec
B	脳ロボットのSCAと右腕ロボットのDSA	データ共有	10msec	113 μ sec
B	脳ロボットのTMAと右腕ロボットのFCA	タスク要求	100msec	45 μ sec
B	脳ロボットのSCAと左腕ロボットのDSA	データ共有	10msec	109 μ sec
B	脳ロボットのTMAと左腕ロボットのFCA	タスク要求	100msec	41 μ sec
B	脳ロボットのSCAと頭ロボットのDSA	データ共有	33msec	105 μ sec
B	脳ロボットのTMAと頭ロボットのFCA	タスク要求	100msec	37 μ sec
B	車輪移動ロボットのDSAと頭ロボットのDSA	データ共有	33msec	112 μ sec
B	車輪移動ロボットのFCAと頭ロボットのFCA	タスク要求	100msec	45 μ sec
B	右腕ロボットのDSAと頭ロボットのDSA	データ共有	33msec	109 μ sec
B	右腕ロボットのFCAと頭ロボットのFCA	タスク要求	100msec	41 μ sec
B	左腕ロボットのDSAと頭ロボットのDSA	データ共有	33msec	105 μ sec
B	左腕ロボットのFCAと頭ロボットのFCA	タスク要求	100msec	37 μ sec
B	右腕ロボットのDSAと左腕ロボットのDSA	データ共有	10msec	105 μ sec
B	右腕ロボットのFCAと左腕ロボットのFCA	タスク要求	100msec	37 μ sec

較して、行動計画部を並列・分散処理できることを示した。問題点(2)に対して、プロトタイプロボットR1は、ユーザに対して使用用途や使用環境に応じた機能や形態を提供することが可能であることを示した。さらには、プロトタイプロボットR1は、搭載する情報処理システムのネットワークトポロジを変更しても同じ動作を保証できることを示した。

以上より、本システムアーキテクチャが提案する(1)行動計画部の並列・分散処理、(2)シ

ステム再構成機構が，ヒューマノイドロボットが抱えている2つの問題点である(1)ヒューマノイドロボットが搭載する情報処理システムの情報処理能力不足，(2)拡張性・柔軟性のあるシステム統合化技術の欠如に対して，有効であると判断することができた．

第8章

まとめ

2006年現在，我が国は，少子高齢化・労働力人口の減少・国際競争の激化といった課題を抱えている．今後，これらの問題は，労働生産性の向上，女性の社会進出等，総合的な政治対策の中で解決されるべきであるが，ロボット産業国である我が国は競争力を発揮しやすい，製造業の生産工程又はサービス業の作業工程において一層のロボット化・自動化が不可欠であるという理由から，我が国が得意とするロボット技術は，これらの解決の糸口を与え，日本のさらなる成長の一翼を担っていくと考えられる．このように，ヒューマノイドロボットは，今後の製造業，サービス業の国際的な競争力や，少子高齢化・労働力の減少に対しての解決策として期待されている．

しかし，2006年現在のヒューマノイドロボットはその期待を応えられるレベルに達していない．レベルに達していない原因を追求したところ，その原因は従来のヒューマノイドロボットが抱える2つの問題点に起因することが明確化された．1つ目の問題点は，従来のヒューマノイドロボットが搭載する情報処理システムが，期待に応じた動作をするための処理能力を満たしていない点であった．2つ目の問題点は，従来のヒューマノイドロボットが，現在までに開発されてきた優れた電気・電子技術，センサ・アクチュエータ技術，システム・制御技術，計算機・情報技術，機械・製造技術などの電気・電子ハードウェアとソフトウェアを搭載でき，異分野の技術領域の融合による新たな技術を創出できるシステム統合化技術が提唱されていない点であった．人間支援のためのヒューマノイドロボットを実現するためには，高度な情報処理能力を備える情報処理システムと，要素技術をヒューマノイドロボットへ統合するためのシステム統合化技術の2つを実現できるシステムアーキテクチャの提案が待たれていることがわかった．

これら2つの問題点を解決できるシステムアーキテクチャを提案するにあたり，本研究では，問題点(1)であるヒューマノイドロボットが搭載する情報処理システムの情報処理能力不足に対して，ヒューマノイドロボットの動作の並列性に着目した．従来のヒューマノイドロボットは視覚処理，音声処理，行動計画を複数のプロセッサ上で並列処理している．特に視覚処理，音声処理に関しては，アルゴリズムをFPGAやASICなどの専用ハードウェア

8. まとめ

ア上へ実装し、高速な並列処理を実現している。しかし、従来のヒューマノイドロボットは、行動計画処理を1つのプロセッサ上で並行処理している。つまり、行動計画処理を並列化することがロボットの情報処理能力向上に繋がると判断することができる。ヒューマノイドロボットの動作に着目すると、ヒューマノイドロボットが搭載する情報処理システムは、従来のヒューマノイドロボットの情報処理システムのように視覚処理、音声処理、行動計画部を並列処理する構成よりも、頭部、システム全体の行動計画部、移動部、アーム部という粒度で並列処理する構成の方が適していると解釈することができた。従って、問題点(1)に対して、本システムアーキテクチャは行動計画部の並列・分散処理化を行うことで解決することとした。

続いて、問題点(2)である拡張性・柔軟性のあるシステム統合化技術の欠如に対してはモジュール型ロボット技術に着目した。モジュールロボットは、「モジュール」と呼ばれるサブシステムの結合から構成するロボットであり、周囲の環境に合わせてその機能や形態を変化させながら、その環境に応じた移動方法や作業方法を行うことができる。更に、モジュールロボットは、物理的にもモジュール毎に着脱を容易な構造にすることによって、モジュール単位での再利用や交換が可能になり、故障や仕様変更、システム拡張に対して柔軟に、低コストで対応できる。このようにシステム全体をモジュール化・部品化しているモジュールロボットは、新しい機能を持つモジュールを追加したり、取り外したりすることによってシステム全体として持つ機能の再構築が可能である点、使用環境に応じて任意の組み合わせで動作可能である点が利点であった。従って、問題点(2)に対して、システムアーキテクチャはシステム再構成機構を採用することで解決することとした。

以上より、本研究では、これら2つの解決法を「通常は、単体の要素技術ロボットとして動作し、ヒューマノイドロボットに統合化された際には、ヒューマノイドロボットの部品として動作する」という設計概念を持つ再構成可能なモジュール型ヒューマノイドロボットのシステムアーキテクチャを提案した。

本システムアーキテクチャは、再構成可能なモジュール型ヒューマノイドロボットを実現するために、機能別ロボットの提案、脳ロボットの提案、再構成可能なモジュール型ヒューマノイドロボットの制御方式、階層化制御ソフトウェアプラットフォーム、モジュール間通信機構 RL-Net、各機能別ロボットが要求する通信レイテンシとバンド幅をシミュレーションできる RL-NetSim 機構、モジュール接続ユニットの7項目から構成される。

本研究では、本システムアーキテクチャが提案する(1)行動計画部の並列・分散処理、(2)システム再構成機構が有効であるかどうかを、3つの機能別ロボットから構成する車輪移動ロボット型のプロトタイプロボット P1 と、5つの機能別ロボットから構成するヒューマノイドロボット型のプロトタイプロボット R1 を利用した実験を通して検証した。問題点(1)に対して、プロトタイプロボット P1 とプロトタイプロボット R1 は、従来のヒューマ

8. まとめ

ノイドロボットと比較して、行動計画部を並列・分散処理できることを示した。問題点(2)に対して、プロトタイプロボット R1 は、ユーザに対して使用用途や使用環境に応じた機能や形態を提供することが可能であることを示した。さらには、プロトタイプロボット P1 とプロトタイプロボット R1 は、構成する情報処理システムのネットワークポロジの変更に対しても同じ動作を安定して動作可能であることを示した。

以上より、本システムアーキテクチャが提案する(1)行動計画部の並列・分散処理、(2)システム再構成機構が、ヒューマノイドロボットが抱えている2つの問題点である(1)ヒューマノイドロボットが搭載する情報処理システムの情報処理能力不足、(2)拡張性・柔軟性のあるシステム統合化技術の欠如に対して、有効であると判断することができた。

謝辞

本研究を進めるにあたり，適切なご指導を頂きました慶應義塾大学助教授 山崎 信行先生に厚く御礼申し上げます．先生には学部時代から研究，論文に至るまで大変お世話になりました．論文執筆にあたり，有益なアドバイスを頂いた慶應義塾大学教授 天野 英晴先生，慶應義塾大学教授 今井 倫太先生，慶應義塾大学教授 大西 公平先生には大変お世話になりました．先生方にはお忙しい中，論文を見て頂き，大変感謝しております．

本研究を行う機会を頂いた慶應義塾大学教授 安西 祐一郎先生に心より感謝を申し上げます．また，ロボットの製作に協力して下さった，三和金属工業株式会社 野田 正弘さんに深く感謝いたします．山崎研究室，安西・今井研究室で共に過ごしたメンバーには色々と助けて頂き，感謝しております．最後に，私の進む道を暖かく見守っていただき，また日頃から支えてくれた家族に深く感謝の意を表します．

2007年2月

平 哲也

参考文献

- [Akachi *et al.* 05] Kazuhiko Akachi, Kenji Kaneko, Noriyuki Kanehira, Shigehiko Ota, Go Miyamori, Masaru Hirata, Shuuji Kajita, and Fumio Kanehiro. Development of humanoid robot HRP-3P. In *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, pp. 50–55, 2005.
- [ARC] *ARCNET*.
- [Arkin 98] R. C. Arkin. *Behavior-based Robotics*. MIT press, 1998.
- [Association] 1394 Trade Association. *1394 Standards and Specifications Summary*.
<http://www.1394ta.org/index.html>.
- [Audsley 90] N. C. Audsley. *Deadline Monotonic Scheduling*, 1990.
- [Breazeal 99] Breazeal. A Context-Dependent Attention System for a Social Robot. In *Proceedings of the IEEE-Joint Conference on Artificial Intelligent*, pp. 1146–1151, 1999.
- [Castano *et al.* 00] Andres Castano, Wei-Min Shen, and Peter Will. CONRO: Towards Deployable Robots with Inter-Robot Metamorphic Capabilities. *Autonomous Robots Journal*, Vol. 8, No. 3, pp. 309–324, 2000.
- [Chen *et al.* 02] Jiming Chen, Zhi Wang, and Youxian Sun. Real-Time Capability Analysis For Switch Industrial Ethernet Traffic Priority-Based. In *Proceedings of IEEE International Conference on Control Applications*, pp. 525–529, 2002.
- [Chestnut *et al.* 03] Joel Chestnut, James J. Kuffner, Koichi, Nishiwaki, and Satoshi Kagami. Planning Biped Navigation Strategies in Complex Environments. In *Proceedings of International Conference on Humanoid Robots*, 2003.
- [FlexRay Consortium] FlexRay Consortium,
<http://www.flexray.com/>. *FlexRay*.

- [Fujita 00] Masahiro Fujita. Digital Creatures for Future Entertainment Robotics . In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 801–806, 2000.
- [Fujita *et al.* 99] Masahiro Fujita, Hiroaki Kitano, and Toshitada Doi. Syntactic-Semantic Analysis of Reconfigurable Robot. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1567–1572, 1999.
- [Group] PCI Industrial Computer Manufacturers Group. *CompactPCI Specification*.
<http://www.picmg.org/compactpci.stm>.
- [Hirai 97] Kazuo Hirai. Current and Future Perspective of Honda Humanoid Robot. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 500–508, 1997.
- [Hoang *et al.* 02] Hoai Hoang, Magnus Jonsson, Ulrik Hagstron, and Anders Kallerdahl. Switched Real-Time Ethernet with Earliest Deadline First Scheduling - Protocols and Traffic Handling. In *Proceedings of IEEE International Parallel and Distributed Processing Symposium*, pp. 94–99, Apr 2002.
- [HONDA] HONDA. *HONDA Humanoid Robot Site*. <http://www.honda.co.jp/robot/>.
- [Hosoda *et al.* 06] Yuji Hosoda, Saku Egawa, Junichi Tamamoto, Kenjiro Yamamoto, Ryousuke Nakamura, and Masahito Togami. Basic Design of Human-Symbiotic Robot EMIEW. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5079–5084, 2006.
- [Inaba 97] Masayuki Inaba. Remote-Brained Humanoid Project. *Advanced Robotics*, Vol. 11, No. 6, pp. 605–620, 1997.
- [IONA] IONA,
<http://www.iona.co.jp/>. *CORBA orbit/E*.
- [ISA94] *Plug and Play ISA Specification v.1.0a*.
<http://www.microsoft.com>, 1994.
- [Ishida 04] Tatsuzo Ishida. Development of a small biped entertainment robot qrio. In *Proceedings of International Symposium on Micro-Nanomechatronics and Human Science*, pp. 23–28, 2004.

- [Ishiguro *et al.* 99] Hiroshi Ishiguro, Toshiyuki Kanda, Katsumi Kimoto, and Toru Ishida. A Robot Architecture Based on Situated Modules. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1617–1623, 1999.
- [Kagami *et al.* 02] 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 *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2557–2562, 2002.
- [Kaiser *et al.* 99] 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*, 1999.
- [Kamimura *et al.* 01] Akiya Kamimura, Satoshi Murata, Eiichi Yoshida, Haruhisa Kurokawa, Kohji Tomita, and Shigeru Kokaji. Self-Reconfigurable Modular Robot - Experiments on Reconfiguration and Locomotion. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 606–612, 2001.
- [Kanehiro *et al.* 02] Funiio Kanehiro, Kiyoshi Fujiwara, Shuuji Kajita, Kazuhito Yokoi, Kenji Kaneko, Hirohisa Hirukawa, Yoshihiko Nakamura, and Katsu Yamane. Open Architecture Humanoid Robotics Platform. In *Proceedings of IEEE International Conference on Robotics and Automation* , pp. 24–30, 2002.
- [Kaneko *et al.* 02] Kenji Kaneko, Fumio Kanehiro, Shuuji Kajita, Kazuhiko Yokoyama, Kazuhiko Akachi, Toshikazu Kawasaki, Shigehiko Ota, and Takakatsu Isozumi. Open Architecture Humanoid Robotics Platform. In *Proceedings of IEEE/RSJ International Conference on Robotics and Automation*, pp. 24–30, 2002.
- [Kawauchi *et al.* 94] Yoshio Kawauchi, Makoto Inaba, and Toshio Fukuda. Dynamically Reconfigurable Intelligent System of Cellular Robotic System(CEBOT) with Entropy Min/Max Hybrid Algorithm. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 464–469, 1994.
- [Kobayashi *et al.* 04] Hidenori Kobayashi, and Nobuyuki Yamasaki. RT-Frontier: A Real-Time Operating System for Practical Imprecise Computation. In *Proceedings of IEEE Real-Time and Embedded Technology and Applications Symposium*, pp. 255–264, 2004.

- [Komuro *et al.* 03] Takashi Komuro, Idaku Ishii, Masatoshi Ishikawa, and Atsushi Yoshida. A Digital Vision Chip Specialized for High-speed Target Tracking. *IEEE transaction on Electron Devices*, Vol. 50, No. 1, pp. 191–199, 2003.
- [Lehiczky *et al.* 89] J. P. Lehiczky, L. Sha, and Y. Ding. The Rate Monotonic Scheduling Algorithm: Exact Characterization and Average Case Behavior. In *Proceedings of IEEE 10th Real-Time Systems Symposium*, pp. 166–171, 1989.
- [Matsumoto *et al.* 02] Kohsei Matsumoto, Kenichi Shimada, Huiying Chen, Jun Ota, and Tamio Arai. Interface Design and Implementation for Mono-Functional Modular Robots. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3018–3023, 2002.
- [Miyashita *et al.* 04] Takahiro Miyashita, and Hiroshi Ishiguro. Human-like Natural Behavior Generation based on Involuntary Motions for Humanoid Robots. *Journal of Robotics and Autonomous Systems*, Vol. 48, No. 4, pp. 203–212, 2004.
- [Moravec 00] Hans Moravec. ロボットは人間を超えるか. Vol. 30, No. 1, pp. 104–111, 2000.
- [Nakadai *et al.* 02] Kazuhiro Nakadai, Hiroshi Okuno, and Hiroaki Kitano. Auditory Fovea Based Speech Separation and Its Application to Dialog System. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1314–1319, 2002.
- [Peleg *et al.* 97] Alex Peleg, Sam Wilkie, and Uri Weiser. Intel MMX for multimedia PCs. *Communications of the ACM*, Vol. 40, No. 1, pp. 24–38, 1997.
- [Raman *et al.* 00] Srinivas K. Raman, Vladimir Pentkovski, and Jagannath Keshave. Implementing Streaming SIMD Extensions on the PentiumIII Processor. *IEEE Micro*, Vol. 20, No. 4, pp. 47–57, 2000.
- [R.Bischoff *et al.* 99] R.Bischoff, and V.Graefe. Integrating, Vision, Touch and Natural Language in the Control of a Situation-Oriented Behavior-Based Humanoid Robot. In *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pp. 999–1004, 1999.
- [Sakagami *et al.* 02] Yoshiaki Sakagami, Ryuji Watanabe, Chiaki Aoyama, Shinichi Matsunaga, Nobuo Higaki, and Kikuo Fujimura. The Intelligent ASIMO: System Overview

- and Integration. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2478–2483, 2002.
- [Taira *et al.* 04] Tetsuya Taira, and Nobuyuki Yamasaki. Functionally Distributed Control Architecture for Robot Systems. In *Proceedings of International Conference on Computers and Their Applications*, pp. 270–273, 2004.
- [Taira *et al.* 05] Tetsuya Taira, Nobuhide Kamata, and Nobuyuki Yamasaki. Design and Implementation of Reconfigurable Modular Humanoid Robot Architecture. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1071–1076, 2005.
- [Taira *et al.* 06] Tetsuya Taira, and Nobuyuki Yamasaki. Design and Implementation of the Reconfiguration Mechanism for a Modular Humanoid Robot. *Journal of Robotics and Mechatronics*, Vol. 13, No. 3, pp. 286–298, 2006.
- [Uchimura *et al.* 04] Yutaka Uchimura, Nobuyuki Yamasaki, and Kohei Ohnishi. Prioritized Data Transfer for a Bilateral Robot Control via Real-Time Network System. *Transactions of the Society of Instrument and Control Engineers*, Vol. 40, No. 2, pp. 199–206, 2004.
- [USB] *Universal Serial Bus Specification Revision 2.0*.
<http://www.usb.org>. 2000.
- [Wind River Systems] Wind River Systems,
<http://www.windriver.com/>. *VxWorks*.
- [Yamasaki 01] Nobuyuki Yamasaki. Design and Implementation of Responsive Processor for Parallel/Distributed Control and Its Development Environments. *Journal of Robotics and Mechatronics*, pp. 125–133, 2001.
- [Yamasaki 05] Nobuyuki Yamasaki. Responsive Multithreaded Processor for Distributed Real-Time Systems. *Journal of Robotics and Mechatronics*, Vol. 17, No. 2, pp. 130–141, 2005.
- [Yim *et al.* 00] Mark Yim, David G. Duff, and Kimon D. Roufas. PolyBot: a Modular Reconfigurable Robot. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 514–520, 2000.

[Yodaiken *et al.*] Victor Yodaiken, and Michael Barabanov. *RT-Linux*.

<http://www.fsmlabs.com/>.

[トヨタ自動車株式会社] トヨタ自動車株式会社,

<http://www.toyota.co.jp/>. トヨタパートナーロボット.

[於久 他 02] 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.

[加賀美聡 他 02] 加賀美聡, James Kuffner, 西脇光一, 稲葉雅幸, 井上博允. ヒューマノイド H7 の足接地場所の 2 次元高速計画法. 第 20 回ロボット学会学術講演会予稿集, p. 1C24, 2002.

[株式会社ムービングアイ] 株式会社ムービングアイ,

<http://www.movingeye.co.jp/>. *ART-Linux*.

[株式会社安川電機] 株式会社安川電機,

<http://www.yaskawa.co.jp/>. *SmartPal*.

[鎌田 他 04] 鎌田展秀, 平哲也, 山崎信行. 環境認識のための超音波センサシステムの設計と実装. ロボティクス・メカトロニクス講演会 2004, pp. 1A1–S–036, 2004.

[鎌田 他 05] 鎌田展秀, 岡本穰, 山越隆央, 平哲也, 山崎信行. 環境認識のためのカメラシステムの設計と実装. ロボティクス・メカトロニクス講演会 2005, pp. 1A1–S–036, 2005.

[鎌田 他 06] 鎌田展秀, 平哲也, 伊藤務, 山浦康史, 上山真生, 山崎信行. 二足歩行を実現するための分散制御システムの構築. ロボティクス・メカトロニクス講演会 2006, pp. 2P2–C22, 2006.

[鏡 他 03] 鏡慎吾, 小室孝, 石井抱, 石川正俊. ビジョンチップのためのマルチターゲットトラッキングとその応用. 電子情報通信学会論文誌, Vol. J86-D-II, No. 10, pp. 1411–1419, 2003.

[黒滝 他 04] 黒滝俊輔, 鈴木紀章, 安生健一郎, 本村真人, 若林一敏, 天野英晴. FFT の DRP によるアクセラレーション手法. In *FPGA/PLD Conference and Exhibition*, 2004.

[佐藤秀雄 他 01] 佐藤秀雄, 矢向高弘. 時間制約の厳しいマルチメディアのためのリアルタイム通信機構. Vol. 42, No. 2, pp. 189–196, 2001.

- [山浦 他 05a] 山浦康史, 鎌田展秀, 平哲也, 山崎信行. 可動型赤外線センサを用いた自律移動ロボットの地図生成システムの設計と実装. ロボティクス・メカトロニクス講演会 2005, pp. 1P2-S-020, 2005.
- [山浦 他 05b] 山浦康史, 鎌田展秀, 平哲也, 山崎信行. 可動型赤外線センサを用いた動的スキャンによる地図生成システムの設計と実装. 第23回日本ロボット学会学術講演会, pp. 2A1-H-34, 2005.
- [山浦 他 06] 山浦康史, 鎌田展秀, 平哲也, 山崎信行. モジュール型ヒューマノイドロボットのための二足歩行モジュールの開発. ロボティクス・メカトロニクス講演会 2006, pp. 2P1-B10, 2006.
- [山崎 04] 山崎信行. 分散制御用リアルタイム通信 Responsive Link の設計および実装. 情報処理学会論文誌コンピューティングシステム, Vol. 45, No. SIG 3, pp. 50-63, 2004.
- [小屋迫光太郎 他 96] 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ッサモジュールの開発. 第14回ロボット学会学術講演会予稿集, pp. 355-356, 1996.
- [小林 他 03] 小林啓吾, 仲谷篤人, 高橋秀行, 潮俊光. 時間付きペトリネットとモジュラステートネットを用いたヒューマノイドロボットの行動計画. システム制御情報学会論文, Vol. 16, No. 10, pp. 493-499, 2003.
- [松井 他 04] 松井俊浩, 比留川博久, 石川裕, 山崎信行, 加賀美聡, 堀俊夫, 金広文男, 斎藤元, 稲邑哲也. ヒューマノイド・ロボットのための実時間分散情報処理. 電子情報通信学会, 2004.
- [上山 他 06] 上山真生, 平哲也, 鎌田展秀, 伊藤務, 山崎信行. モジュール型ヒューマノイドロボットのアームモジュールの設計と実装. ロボティクス・メカトロニクス講演会 2006, pp. 2P2-C23, 2006.
- [神田 他 02] 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, Vol. 20, No. 3, pp. 315-323, 2002.
- [神田 他 06] 神田真司, 村瀬有一, 岡林桂樹, 麻田務. サービスロボット:enon. 富士通技報, pp. 307-313, 2006.
- [清水正晴 他 01] 清水正晴, 奥山悠, 田原哲雄, 古田貴之, 北野宏明, 富山健. 小型・多関節ヒューマノイドにおける全身運動制御用インテリジェントネットワークデバイスの開発. 第19回日本ロボット学会学術講演会予稿集, pp. 791-792, 2001.

- [石綿 他 98] 石綿陽一, 松井俊浩, 國吉康夫. 高度な実時間処理機能を持つLinuxの開発. 第16回日本ロボット学会学術講演会予稿集, pp. 355–356, 1998.
- [石綿陽一 他 04] 石綿陽一, 金広文男, 比留川博久, 齋藤元. Ethernetを用いた実時間通信方式の開発. 第22回日本ロボット学会学術講演会予稿集, p. 1C25, 2004.
- [川内 他 03] 川内直人, 古結義浩, 長島是, 大西献, 日浦亮太. ホームユースロボット“wakamaru”. Vol. 40, No. 5, 2003.
- [前野 他 98] 前野隆司, 竹村研治郎, 小島信行. 縦振動と横振動の縮退に基づく多自由度超音波モータの開発. 日本ロボット学会誌, Vol. 16, No. 8, pp. 1115–1122, 1998.
- [藤田雅博 他 01] 藤田雅博, 佐部浩太郎. ロボットエンターテインメントの提案. 第19回ロボット学会学術講演会, pp. 1089–1090, 2001.
- [平 他 03] 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, pp. 1P1–1F–D3, 2003.

論文目録

主論文に関する公刊論文

- Tetsuya Taira , and Nobuyuki Yamasaki , “Design and Implementation of the Reconfiguration Mechanism for a Modular Humanoid Robot” , Journal of Robotics and Mechatronics Vol. 18, No. 3, pp. 286-298, 2006.
- Tetsuya Taira , and Nobuyuki Yamasaki , “Development of Modular Humanoid Robot based on Functionally Distributed Modular Robot Architecture” , Journal of Robotics and Mechatronics Vol. 17, No. 3, pp. 236-247, 2005.
- Tetsuya Taira , and Nobuyuki Yamasaki , “Functionally Distributed Control Architecture for Autonomous Mobile Robots” , Journal of Robotics and Mechatronics Vol. 16, No. 2, pp. 217-224, 2004.

国際会議発表

- Tetsuya Taira, Nobuhide Kamata, and Nobuyuki Yamasaki, “Design and Implementation of Reconfigurable Modular Humanoid Robot Architecture” , IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1071-1076, 2005.
- Tetsuya Taira, and Nobuyuki Yamasaki, “Functionally Distributed Control Architecture for Robot Systems” , International Conference on Computers and Their Applications, pp. 270-273, 2004.