

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

○於久健太郎[†] 清水健二[†] 松坂要佐[†] 小林哲則[†]

大規模統合システムであるヒューマノイドロボットの開発において、リアルタイム性を失わない拡張性と、モジュール間での円滑な情報共有を実現するためのシステムアーキテクチャについて検討する。各モジュールの処理をそれぞれ専用プロセッサに割り当てることにより並列処理してそれらを情報共有バスで接続するマルチプロセッサアーキテクチャを提案する。マルチモーダル対話ロボットの実装を通して、高密度に統合されたシステムが実現できることを確認する。

Multiprocessor Architecture for Multi-modal Humanoid Robots

○Kentaro OKU[†], Kenji SHIMIZU[†], Yosuke MATSUSAKA[†], Tetsunori KOBAYASHI[†]

Abstract — We discuss about a system architecture of humanoid robot which is a large scale integrated system, to realize extensibility without real-time performance and data sharing between functional modules. We propose a multiprocessor architecture, which processes every functional module at its exclusive processor in parallel, and connects them by data shared bus. We evaluate the efficiency of the proposed architecture through the experience in implementing a multi-modal conversation robot.

Keywords: integrated system, data shared bus, parallel processing, motor control, image processing, sound processing

1. はじめに

ヒューマノイドロボットに代表される多機能ロボットにおいては、音声・画像・対話・移動・制御などの多様な情報処理能力と、各機能間の協調動作が必要となる。例えば、当研究室のヒューマノイドロボット”ROBITA”・”ROBISUKE”における人とのグループ対話のタスクでは、発話内容を理解するための音声認識、対話者の顔画像認識、目と首のモータ制御などを実現するモジュール間の協調動作により、システム全体として一つのタスクが実行されている[1]。

このような様々な機能を実装されたロボットは、全体として膨大な機能をもった統合システムであり、様々な分野の専門家による大規模なプロジェクトとなる。したがって、これまでロボット工学の別々の分野で研究・開発されてきたさまざまな技術を、一つのシステムとして統合する必要がある。

優れたヒューマノイドロボットの実現を目指すには、ロボットが常に一つのタスクのみしか実行するのではなく、使用する目的に応じてさまざまなタスクを実行できると便利である。しかし、膨大なシステムであるヒューマノイドロボットのプログラムを、毎回タスクごとに全て書き直すのは非効率である。

ロボットが一つのタスクを行うには、各モジュール内であらかじめ処理して単純化された小容量の情

報を、モジュール間で共有することにより実現できる。例えば、ロボットの眼で人の顔の追跡をさせたい場合、画像処理モジュールがカメラからの画像をもとに肌色検出を行い、人の顔の位置情報を算出する。そして、モータ制御モジュールの眼と首のモータの現在角度情報を基に目標角度情報の指定を行うことで実現できる。このように、各モジュール間で共有している情報は、肌色位置情報、現在角度情報、目標角度情報の小容量の情報である。

また、モジュール間の情報共有を組み合わせることにより、さまざまなタスクの切り替えが実現できる。例えば、人の顔を追跡する代わりに、音が聞こえる方向にロボットを向かせるタスクを実現することを考える。この場合、音声処理モジュールがステレオマイクにより音源位置情報を算出する。そして同様に、モータ制御モジュールに目標角度情報の指定をする。このとき、画像処理モジュールの情報は無視する。

我々は、このように必要な様々なタスクの変換・再利用や機能の追加に対応するには、モジュール間でデータ入出力の依存関係や動作を妨げない情報共有を行い、様々なタスクに応じてモジュールの組み合わせを切り替えることにより、比較的容易にロボットシステムが開発できると考えている。このようなシステム開発においては、各モジュールが、全体のシステムの中から必要な情報を自由にアクセスできるような情報共有の枠組みが必要である。

シングルプロセッサ構成のシステムに、モジュールを追加していくと、プロセッサの負荷が増大し、モジュールのリアルタイム性が低下する。リアルタイム性の低下は、スムーズなロボットの動作に重大な支障をきたす。各モジュールごとに一つのプロセ

[†] 〒169-0072 東京都新宿区大久保 3-4-1
早稲田大学 理工学部 ヒューマノイド研究所
Humanoid Robotics Institute, School of Science and
Engineering, Waseda University
3-4-1 Okubo, Shinjuku, Tokyo, 169-8555 Japan
{oku, shimizu, yosuke, koba}@tk.elec.waseda.ac.jp



図1 ヒューマノイドロボット“ROBISUKE”と7台の外部サーバ・クライアントPC

ッサを割り当て、マルチプロセッサで並列処理することにより、新たなモジュールの追加によるリアルタイム性の低下を防止することが可能である。

マルチプロセッサ構成のシステムにおいて、プロセッサを増設する際、通信インターフェースが異なるとプロセッサ間同士で接続することができない。通信インターフェースを統一すれば、新しいモジュールを開発して既存モジュールに追加・接続することで、容易に新しい機能やタスクが実現できる。よって、拡張性を持たせるには、各モジュールで使われる情報統合のための通信インターフェースを統一することが望ましい。

このように、ヒューマノイドロボットの開発において、モジュールを組み合わせることによりさまざまなタスクの変換・再利用や機能の追加に対応するには、拡張性・並列性・リアルタイム性・柔軟性を備えた情報共有システムの構築が必要である。以上の点を考慮に入れて、ソフトウェアとハードウェアの両方を互いに考慮しながらアーキテクチャレベルでロボットシステムを設計する。

本稿では、情報共有バスで接続されたマルチプロセッサ構成のモータ制御・画像処理・音声処理ボードを設計・製作する。そして、それらをヒューマノイドロボットに実装し、各モジュール間の高密度に統合された協調動作により実現する。

2. ロボットに必要な機能

人間の視聴覚システムは、センサ系（眼・耳）、情報処理系（脳）、行動系（筋肉）の連携からなる。多機能なヒューマノイドロボットにおいても、同様にセンサ系（カメラ・マイク）、情報処理系（プロセッサ）、行動系（モータ制御）などから構成される。これらを実装するにあたって、必要な主な機能として音声処理・画像処理・モータ制御が挙げられる。

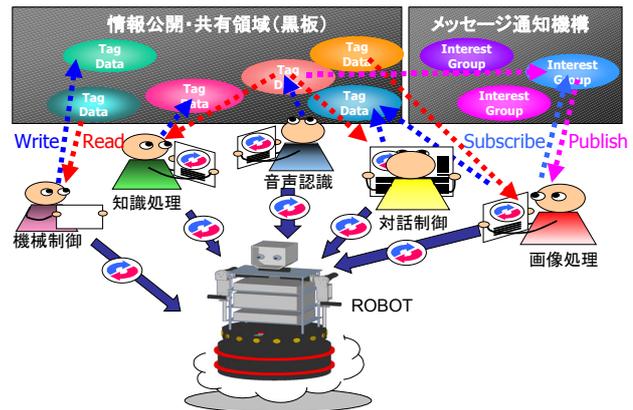


図2 情報公開機構・メッセージ通知機構

2.1. 音声処理機能

ヒューマノイドロボットと人が対話によりコミュニケーションをするにあたって、音声処理は必須である。“ROBITA”の頭部には、左右の耳に2系統のマイクロフォンが設置されており、話かけられた人の方向を検出するための音源方向特定に用いられている。また、2人の話者がもっている2台のワイヤレス・ピン・マイクロフォンに入力された音声を解析して発話内容を理解するため、音声認識を行っている。そして、胴体に設置されたスピーカーからロボットの声を発話させるため、音声合成を行っている。

2.2. 画像処理機能

人に近いロバストなロボットのビジョン・システムの実現のためには、リアルワールドのリアルタイムでの画像処理が必要である。“ROBITA”には左右の目に2台のCCDカメラが内蔵されている。カメラの入力画像から、人の顔の位置を肌色などから推定して追跡する顔追跡、パターンマッチングを行う人物認識、体全体の身体表現を判別するジェスチャ認識、人の視線方向を判別する顔方向認識などさまざまな処理を行っている。

2.3. モータ制御機能

“ROBITA”には、頭部の首・眼・口・眉毛、両腕の肩・肘・手首、両指などを含めて全身で36自由度のモータが配置されている。複雑な自由度を持った多数のモータとセンサを制御するために、リアルタイムの並列ループを1[msec]以内に制御しなければならない。

3. システムアーキテクチャの検討

ヒューマノイドロボットのような高機能のシステムをコンピュータ一台でリアルタイム制御するのは、コンピュータへの情報処理の負荷が著しく大きい。音声処理・画像処理・モータ制御などの各機能ごとに複数のコンピュータに各処理を分割し、並列処理を行い、システム全体の負荷を下げることにより、



図3 製作したボードの外観

左上:画像処理ボード 右上:PC 接続ボード
 左中:音声処理ボード 右下:12cm CD
 左下:モータ制御ボード

これまでに述べた性能面・リアルタイム性などの問題を解決できる。

また、ヒューマノイドロボットシステムの自立化・小型化を実現するには、ロボットに搭載したコンピュータでこれらの処理を行う必要がある。しかし、ロボットに複数のコンピュータを搭載するには、サイズ・消費電力など様々な制約を考慮しなければならない。

これらの機能のロボットへの搭載を可能にするために、組み込み CPU と DSP を用いたシステムを構築することにする。組み込み CPU を用いたロボットシステムはコンパクトなシステム構築が可能、蓄積されたソフトウェア資産が利用可能などの利点があり、高速な処理能力を持ったロボットシステムの構築が実現できる。DSP は音声・画像処理に必要な積和演算などのデジタル信号処理を高速に実行することが可能である。

3.1. 情報共有モデル・メッセージ通知モデル

モジュール間の情報共有を実現するためには、各々のモジュールで生成された情報は、システムに接続している全てのモジュールから参照可能でなければならない。このような情報共有の枠組として情報公開モデルである黒板モデルを採用する。図2の

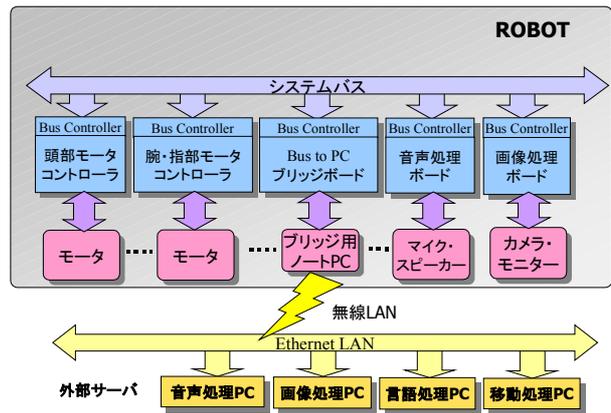


図4 システム構成図

ロボット本体上では、ボード間のシステムバス、ロボット外部では、PC間のシステムソフトウェアにより情報共有を実現している。

ように、黒板モデルでは、各モジュールが共有するデータとタグを黒板に書き込むことで、どのモジュールからもそれらを参照することが可能となる。

黒板モデルは情報共有のモデルとしては優秀なものであるが、共有データの変化を検知する用途には非効率である。そこで、メッセージ通知に適した枠組みである Publish/Subscribe モデルをあわせて採用した。Publish/Subscribe モデルでは、興味のある特定のデータを購読(Subscribe) リストに入れておけば、そのデータに対する変化の通知(Publish)を受けることができる。

3.2. システムバス

3.2.1. バス通信とネットワーク通信

一般的に、並列処理を行うための通信インターフェースとして、マルチプロセッサ(共有メモリ)用のバス通信と、マルチコンピュータ(専用メモリ)用のネットワーク通信がある。

ネットワーク通信は、ルーティングを行うことにより固定された2つのモジュール間の Point to Point のデータ通信を行う。送信側のモジュールはどのモジュールが送信するデータを必要としているか送信先のアドレスを知る必要がある。また、データをリクエストするモジュール側は最新のデータを保持しているモジュールのアドレスを指定しなければならない。様々なモジュールの統合処理により実行される複雑・大規模なタスクを、機能の粒度が細かい部分モジュールの Point to Point のネットワーク結合のみで実現するのは非常に困難である。つまり、情報を全てのモジュールに公開するための通信を行うには、ネットワーク通信には適していない。

バス通信は、接続されたモジュール全てのモジュールに非同期にデータを送信でき、柔軟な情報共有を行うための通信方法として適している。ルーティングを行わないので、1ステップで転送が完了する

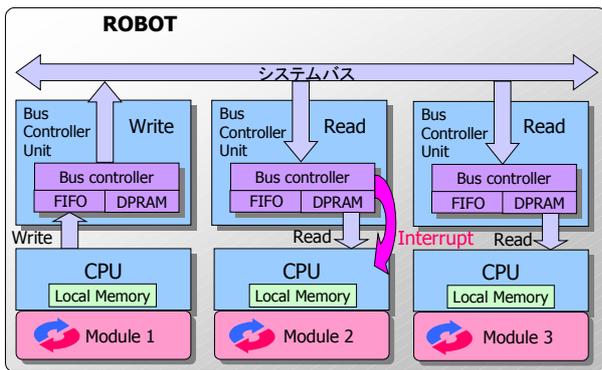


図5 情報共有バスの動作

1. DPRAM による情報公開機能、
2. FIFO によるバッファ機能
3. CPU への割り込みによるイベント通知機能、を実現している。

ため、転送時間のコスト計算が容易である。また、電氣的に並列に接続さえすれば、コネクタを分岐や延長させることにより様々なトポロジーが構築可能である。

このように Point to Point の通信しか行えないネットワーク通信に比べて、データが全てのモジュールにブロードキャストされるバス通信は拡張性や柔軟性において利点が多い。

3.2.2. バス型マルチプロセッサ

通常、バス型マルチプロセッサは複数のプロセッサと1つの共有メモリから構成されており、共通のバスに接続している。典型的なバスは8~64ビットのデータ線・アドレス線と複数の制御線を持ち、これらはクロックに同期しながら動作する。メモリにあるデータを読み出すため、CPUは必要なデータのアドレスをバスアドレス線に送信し、メモリからのデータを受信する。CPUは同様な方法でメモリへのデータの書き込み作業を行う。バス型マルチプロセッサでは、メモリが1つしかないので、データの一貫性が取れる。しかし、この方法では通常4~5個のCPUがある場合、バスは過負荷状態になり性能は急激に低下するため、リアルタイム性が失われてしまう[14]。

本システムでは、各CPUの内部のみで使用するローカルデータを高速なローカルメモリに記憶し、CPU間で共有するグローバルデータのみを共有メモリに記憶することにより、処理速度の低下となるバス経由の通信が最小限に抑えて高速化を実現する。

3.2.3. 汎用のバス型アーキテクチャ

ロボットの制御用通信として、ISA、PCI、Ethernet、USB[7]、IEEE-1394(FireWire)、レスポンスリンク[8]、CAN(Controlled Area Network)バス[9]、I2Cバス[10]などの標準規格や、TITechWire[11]、CATENA-SAGE Network[12]、Open-R[13]などの独自規格を用いた各

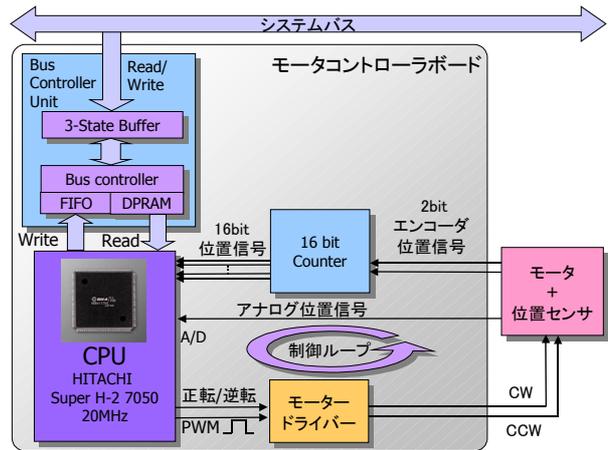


図6 モータ制御ボードのブロック図

種アーキテクチャが提案されている。

ISA や PCI は、バックプレーンボードが必要であり、配置の制約がある。CAN バス・I2C バスは、高速性を満たさない。3.2.1 章で述べたように、我々は IEEE-1394・USB・Ethernet・レスポンスリンクのような Point to Point のネットワーク型トポロジーは、多機能で柔軟なタスク変更を行うロボットには適さないと考えている。TITechWire・CATENA-SAGE Network はモータ制御のモジュールのみ実装されている。

本研究では、ロボットが必要とする情報共有を実現するため、これらのバス規格と比較して、各モジュールの並列性、リアルタイム性を失わない拡張性、タスクの変更が可能な柔軟性を備えたバスアーキテクチャを策定・実装する。

4. システム構成

図4にシステム構成図を示す。開発したシステムは、モータ制御ボード、画像処理ボード、音声処理ボード、ノートPC接続用ボードから構成され、各ボード間の通信を情報共有バスで行う。

4.1. システムバス

[2]において提案された情報公開・メッセージ通知モデルのシステムアーキテクチャをベースに、システムバスを実装した。バス通信に必要な機能として、情報公開機能、イベント通知機能、バッファ機能が必要である。

各々のモジュールで生成された情報は、システムに接続している全てのモジュールから参照可能でなければならないので、黒板モデルにもとづいて、共有メモリを用いた情報公開機能を実現する。

データが更新されるとCPUに自動的に割り込みを行うBus Controllerを実装して、Publisher/Subscribeモデルにおけるイベント通知機能を実現する。

また、バスが使用中の場合、CPUはバスが解放するまで待つ必要がある。そこで、バスへ書き込む際のバッファとしてFIFOメモリを搭載する。

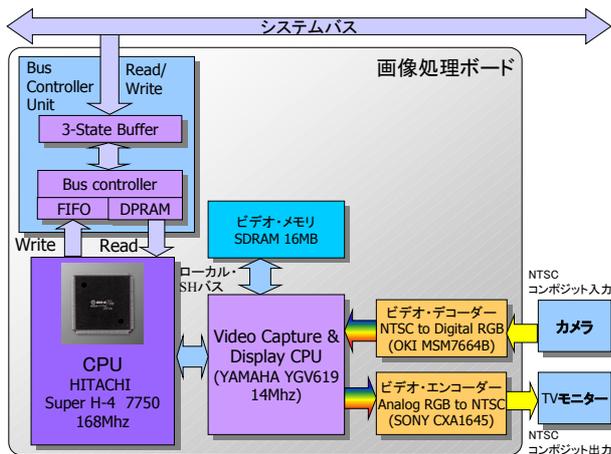


図7 画像処理ボードのブロック図

データの内容が同期している複数の並列分散型の高速メモリを用いて、ロボット用情報共有型システムバスを開発した。各ボード上の Bus Controller、DPRAM(Dual Port RAM)、FIFO から構成されるBCU(Bus Controller Unit)がバスを制御する。

CPU が DPRAM にデータを書き込む際は、バッファCPU が DPRAM にデータを書き込む際は、バッファとして機能する FIFO にデータを書き込む。Bus Controller は、バスが空いた状態になるまで待つ。バスが空いたら、Bus Controller は、FIFO のデータをバスに接続されている全ての DPRAM に同時にデータを書き込む。このように、Bus Controller はデータの同期をとることにより、最新のデータの管理・更新を実現する。また、Bus Controller にあらかじめプログラムされてあるデータが更新されると、自動的に CPU へ割り込んでモジュールにイベント通知を行う。

逆に CPU が DPRAM から値を読み込む場合はメモリーポートから DPRAM に直接アクセスして、データを読み込む。DPRAM は CPU に直結されており、高速アクセスが可能である。

実装した情報共有バスは、8bit の I/O 信号線・制御線・クロック信号・バス電源から構成される 8Mhz のクロック同期式バスで、16bit のアドレス・16bit のデータから構成される 1 パケットを 9 クロックで配信可能であり、最大 9 ノード接続できる。DPRAM 容量は 512byte で、最大 256 個のタグ・データが作成可能である。

4.2. モータ制御ボード

図6にモータコントローラの構成図を示す。

角度センサとしてモータに搭載されたエンコーダまたはポテンショメータの現在角度の値はパルスカウンタまたはADコンバーターを経由してCPUに入力される。CPUはDPRAMの目標角度と角度センサ

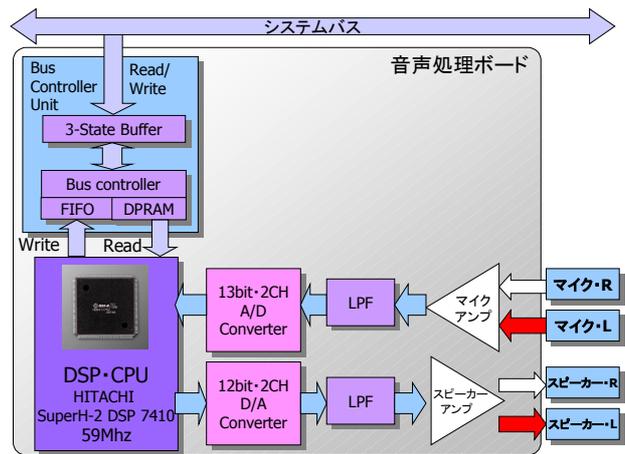


図8 音声処理ボードのブロック図

からの現在角度を読みとり、制御計算を実行する。CPUは、PWM信号と正転/逆転信号を出力する。モータドライバは信号をDC12V/24V[標準 3A]に増幅してモータに出力する。1つのボードで、最大11個のモータを制御可能である。

4.3. 画像処理ボード

図7に画像処理ボードの構成図を示す。

ロボットの目に取り付けられたNTSCカメラから入力されたコンポジット・ビデオ入力信号は、ビデオ・デコーダにデジタルRGBデータに変換される。ビデオ・キャプチャCPUは、ビデオ・フレーム・メモリ(SDRAM)に描画する。画像処理CPUはこのメモリのデータにアクセスして、各種画像処理をリアルタイムで実行する。そして、処理した画像は、ビデオ・エンコーダでNTSC信号に変換されて出力される。

未処理の音声・画像データは膨大なものである。センサから取り込んだデータを直接システムバスに流すのではなく、得られた膨大なデータから必要な情報を抽出するプロセッサをシステムバスに接続する前段階におくことで、システム全体のリアルタイム性の低下の原因となるバスへの負担を最小限に抑えることができる。

4.4. 音声処理ボード

図8に音声処理ボードの構成図を示す。

左右の耳に取り付けられたマイクから入力されたステレオ音声は、アンプ・LPF・A/Dを経由してCPUで各種処理を行う。また、D/A・LPF・アンプを通じてスピーカーに合成したステレオ音声を出力することも可能である。

5. 動作試験

上記検討に基づき、ロボットと外部PC複数台からなる実験システムを構築した。ロボットの本体上に、バス接続された組み込みCPU搭載型モータ制御ボード・画像処理ボード・音声処理ボード・PC接続ボードと、無線LAN内蔵のノートPCを搭載した。

表1 各モジュールの実行場所・使用する資源・入出力データ・更新速度

実行される場所	カテゴリ	モジュール	使用する資源	入力データ	出力データ	更新速度
オンボード	モータ制御	モータ制御モジュール	モータ・角度センサ	目標角度	現在角度	1[msec]
	画像	物体追跡モジュール	CCDカメラ	(入力画像)	物体の座標	30[msec]
	音声	音源定位モジュール	マイク	(入力音声)	音源の角度	10[msec]
外部 PC	画像	人物特定モジュール	CCDカメラ	(入力画像)	人物特定結果	3000[msec]
		顔方向推定モジュール	CCDカメラ	(入力画像)	顔方向	500[msec]
	音声	音声認識モジュール	マイク	(入力音声)	音声認識結果	10[msec]
		音声合成モジュール	スピーカー	質問の解答	(出力音声)	10[msec]
	言語	言語解析モジュール	登録データベース	音声認識結果	質問の解答	1000[msec]
	移動	移動MAPモジュール	台車	目標位置	現在位置	100[msec]

そして、ロボット外部の情報公開サーバ PC と 7 台のクライアント PC に接続した。

ロボット本体上に搭載するのが困難なモジュール・高いレスポンス性が要求されないモジュール・タスク切替などの操作は、ロボット外部に設置されたクライアント PC 上で処理を行う。外部サーバ PC とクライアント PC は、ロボット上に搭載された無線 LAN 搭載・ノート PC 接続ボード経由で情報を共有している。

表 1 に各モジュールの詳細を示す。画像処理ボードに人の顔を追跡する物体追跡モジュールを実装し、音声処理ボードに発話者の音源特定モジュールを実装した。そして、ヒューマノイドロボットにおける人とのグループ対話のタスクの実装を通して、高密度に統合されたシステムが実現できることを確認した。

6. まとめ

大規模統合システムであるヒューマノイドロボットの開発において、リアルタイム性を失わない拡張性と、モジュール間での円滑な情報共有を実現するためのシステムアーキテクチャについて検討した。各モジュールの処理をそれぞれ専用プロセッサに割り当てることにより並列処理してそれらを情報共有バスで接続するマルチプロセッサアーキテクチャを提案した。マルチモーダル対話ロボットの実装を通して、高密度に統合されたシステムが実現できることを確認した。

謝辞

本研究は、株式会社日立製作所・中央研究所・SH フォーラムの一環として行われた。

参考文献

- 1) 松坂要佐、東條剛史、小林哲則、"グループ会話に参与する対話ロボットの構築"、信学論 vol.J84-D-II, no.6, pp.898-908, 2001.
- 2) 松坂要佐、小林哲則、"異分野エキスパート集団によるインタラクティブロボットの共同開発のための基本ソフトウェア"、情報処理学会研究報告 2001-ICS-126, pp.43-48, 2001.
- 3) 松坂要佐、於久健太郎、小林哲則、"マルチモーダル対話ロボットのための情報共有・メッセージ通知システム"、計測自動制御学会 システムインテグレーション部

門学術講演会(SI2000)予稿集、pp.191-192, 2000.

- 4) 金景柱、松坂要佐、小林哲則、"インタラクティブロボットのための複数機能モジュールの協調処理アーキテクチャ"、第7回ロボティクス・シンポジウム、pp215-220, March 2002.
- 5) 於久健太郎、松坂要佐、小林哲則、"ロボット用情報共有型システムバスと組み込み CPU 一体型小型モータコントローラの開発"、第19回 ロボット学会学術講演会論文集、pp.1271-1272, 2001.
- 6) 北垣高成・小笠原司、末広尚士、劉雲輝、"並列処理型センサバスド・マニピュレーション・システム"、日本ロボット学会誌、Vol.15 No.3, pp.363-372, 1997.
- 7) 徳山陽人、精廬幹人、橋本周司、"USB インターフェース搭載型の小型モータコントローラの開発"、第18回日本ロボット学会学術講演会予稿集、pp.119-120, Sept. 2000.
- 8) 山崎信行、松井俊浩、"並列分散リアルタイム制御用レスポンスプロセッサ"、日本ロボット学会誌 Vol.19, No.3, pp.352-36, 1999.
- 9) Kaiser and Michael Mock, "Implementing the Real-Time Publisher/Subscriber Model on the CAN-Bus", Proceedings of Int. Symposium on Object-oriented Real-time distributed Computing (ISORC 99), May 1999.
- 10) 小屋迫光太郎、水内郁夫、星野由紀子、稲葉雅幸、井上博允、"ロボットボディの組み込みプロセッサモジュールの開発"、第14回ロボット学会学術講演会、pp.355-356, 1996.
- 11) 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回ロボット学会学術講演会、pp.791-792, 2001.
- 12) 福島 E.文彦、滝田謙介、広瀬茂男、中村亨、"知能ロボット用シリアル・バス型制御システムの研究"、第7回ロボティクス・シンポジウム、pp155-160, March 2002.
- 13) 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"、pp.1089-1090, 第19回ロボット学会学術講演会、2001.
- 14) 笠原博徳、「並列処理技術」、コロナ社、1991