

[招待論文]

ヒューマノイド・ロボットのための実時間分散情報処理

松井俊浩^{*1}、比留川博久^{*2}、石川裕^{*3}、山崎信行^{*4}
加賀美聡^{*1}、堀俊夫^{*1}、金広文男^{*2}、斎藤元^{*2}、稲邑哲也^{*3}

*1 産業技術総合研究所デジタルヒューマン研究センター

*2 産業技術総合研究所知能システム研究部門

*3 東京大学大学院情報理工学系研究科

*4 慶應義塾大学理工学部情報工学科

t.matsui@aist.go.jp <http://drtp.dip.jp/>

梗概: ヒューマノイド・ロボットがより高度の作業を行えるようにするには、内蔵する情報処理系の能力拡大と信頼性向上が必要である。ヒューマノイド用の高信頼かつスケーラブルな実時間・分散処理系を研究開発する計画について述べる。実時間処理能力に優れたプロセッサのシステム LSI としての実現、実時間分散処理ソフトウェア開発、ヒューマノイドロボットによる実証、の三つの研究で構成する。

キーワード: 実時間処理、分散処理、ヒューマノイド・ロボット、実時間 OS、並列処理、システム・オン・チップ

[Invited Paper]

Distributed Real-Time Processing for Humanoid Robots

Toshihiro MATSUI^{*1}, Hirohisa HIRUKAWA^{*2}, Yutaka ISHIKAWA^{*3}, Nobuyuki YAMASAKI^{*4}
Satoshi KAGAMI^{*1}, Toshio HORI^{*1}, Fumio KANEHIRO^{*2}, Hajime SAITO^{*2}, Tetsuya INAMURA^{*3}

*1 Digital Human Research Center, AIST

*2 Research Institute of Intelligent Systems, AIST

*3 Department of Computer Science, The University of Tokyo

*4 Department of Information and Computer Science, Keio University

Abstract: In order for humanoid robots to perform skillful tasks, a reliable and scalable computing system that supports hard real-time processing is required. This paper describes a five-year project to thoroughly upgrade the computing system of HRP-2 series humanoid robot on the basis of the real-time distributed architecture for higher reliability and scalable computing power. The project consist of three parts; development of two processor ULSI for real-time control, implementation of Linux based real-time operating system, and validation by real robot task execution.

Keywords: real-time processing, distributed processing, humanoid robot, real-time operating system, parallel processing, system-on-chip

1. はじめに

ロボットの発展は、コンピュータの発展と軌を一にしている。工場の産業用ロボットは、ミニコンによって可能になり、ラインに並ぶ数々のロボットはネットワークによって協調した作業を行えるようになった。初期の人間型ロボットは、リモートブレイン方式と呼ばれるように体外に処理系を持たざるを得なかったが、マイクロプロセッサの発達によって、1996年に完全自立型の本格的ヒューマノイドが実現された[Hirai1997]。

ヒューマノイド実現の要となる情報処理系は、処理能力、信頼性、消費電力、物理寸法の点で困難な壁に直面している。本論文は、ヒューマノイドロボットに分散型の情報処理を導入することにより、この限界を突破する計画について述べる。ロボットとしては、既開発のHRP-2(図1)[五十棲 2004]をベースに、その次世代機であるHRP-3の改造版において提案する実時間分散系を導入する。プロセッサノード、ノードを結合する実時間通信方式、Linuxを基礎とするオペレーティングシステム、認識・計画に関わるロボットソフトウェアを用いた機能実証について、2003年から5年間で実施するCRESTプロジェクトを概観する。

2. ヒューマノイドの情報処理系の現状

1996年に登場したヒューマノイドロボットホンダ P-2の滑らかな歩行を実現したのは、背中に搭載された、4個のMicro Sparc IIプロセッサ(60Mips)であった[Honda URL]。産総研と川田工業が中心になって開発した2002年型のHRP-2には、1.2GHzのPentium-IIIプロセッサが2個用いられている(図2)。Pentium-IIIのmips値は、クロックの2倍程度であるとして単純なmips値比較を行うと、6年で20倍の性能向上が図られている¹。後述するが、この計算能力で歩行は可能であるが、動作の計画や高度な認識処理には十分とは言えない。

P-2は、VMEバスを介した密結合のマルチプロセッサ構成であった。HRP-2では、2個のCPUは別々のPCマザーボードに分けて実装され、2枚のマザーボードは100MbpsのEthernetで接続されている。D/A、A/D、カウンタなどセンサやアクチュエータ用I/O

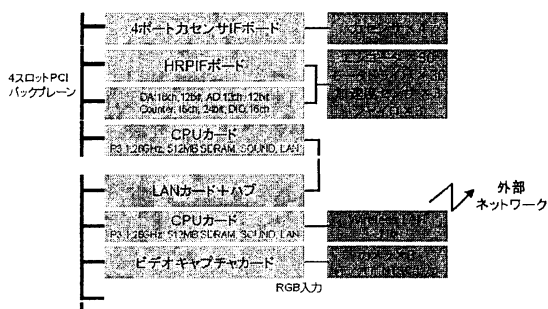


図2 HRP-2の制御コンピュータシステム構成

¹ 一般には、プロセッサの性能は10年で100倍向上すると言われる。この法則に従えば、6年では16倍の向上となる。

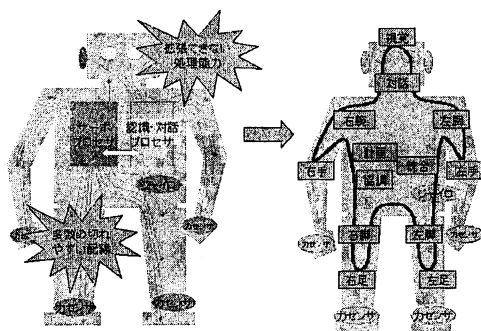


図3 集中型の情報処理系によるヒューマノイドロボット制御の問題とその分散ネットワーク系による解決

は、PCIカードによって実装される(図2)。チャネル数は、関節の30と同数以上必要であり、数が多いためにコネクタやケーブルの配置が、問題となりやすい。また、VMEバスシステムや、PCマザーボードは、据え付け型の使用が前提であり、ヒューマノイドの筐体中で強い加速度や振動を受けながら動作することを想定していない。センサー、アクチュエータへの配線は、HRP-2では180本を越え、それらが屈曲する狭い関節部分をモーターユニット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が発生しやすい。通常動作時に受ける加速度は最大1.5G程度であるが、背臀部にクッションを付けて転倒着地させたときは10G以上に達する。実際に、HRP-2のオペレーションにおけるハードウェアトラブルの多くは、配線に起因する。断線によって制御が停止すれば、ロボットは転倒し、甚大な被害を生じる。

消費電力および熱についても限界に近づきつつある。CPU単体の消費電力は、P-2での20Wから、HRP-2では60Wに増大している。2003年現在、3GHz級のPentium-4は、1個で80Wの電力を消費する。これは、1時間あたり67Kcalに相当する。ヒューマノイドに搭載する電池は、5-600Kcal程度であり、活動時間を数時間とするためには、プロセッサにこれ以上の電力を流す余裕はない²。また、CPUが発生する熱の処理も深刻な問題となりうる。ヒューマノイドの筐体内には、電池、電源ユニット、モータなどが発生する熱がこもりやすい。ファンの使用はノイズ源となり、音声認識等に支障を来す。

以上をまとめると、現状のヒューマノイドロボットの情報処理系は、集中制御型であり、①情報処理能力の不足と、②その拡張が物理的・エネルギー的に困難であること、③配線等の信頼性が、深刻な問題となっていることが指摘できる。

3. ヒューマノイドロボットに必要な情報処理

Hans Moravecは、人間の情報処理能力は1億mipsと試算した[Moravec2000]。ヒューマノイドロボットに必要なコンピューティン

² 80Wは一日で1600Kcal、すなわち大人の基礎代謝に匹敵する。睡眠中は消費電力を下げるができるが、人間の体のエネルギー配分率と比較すると、ロボットの脳は電力を過大に消費している。

グパワーは、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、ロボットが行う作業を定めれば、必要な能力を求めることができそうである。しかし、実際には、たとえば飛んでくるボールを見て捕球する動作に必要な mips 値を見積もることは、その動作戦略やアルゴリズムがわからないため不可能である。しかし、現在のロボットの動作と比較して、何倍の能力があればどの程度の作業ができそうかを見積もることは全く不可能ではない。ここでは、ロボットの情報処理を動作計画、認識、実時間実行、通信・管理の4つに分けて必要な情報処理機能を考察する。

3.1. 動作計画

ヒューマノイドロボットが安定して歩行するためには、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軌道計算、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1-5msの周期で計30個のアクチュエータをサーボする必要がある。歩行に最低限必要なセンサーは、各軸の角度や足の接地覚、ジャイロなどであるが、障害物と衝突しないで動作するためには、触覚、視覚、聴覚などを通じて環境モデルを獲得し、多次元空間³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、ロボットを一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、xyz各軸を512に分割すると、6次元の場合で探索空間の大きさは 6×10^{15} に達する。完全探索は全く不可能であり、遺伝的アルゴリズム、ランダムマイズプランナー、パターン分類など、ヒューリスティックな戦略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、30の関節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、ロボットは冗長な自由度を持つので、もっと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。

2000年にH7の前身に当たるH5ロボットを用いて、ジョイスティックのリアルタイム指令に従ってヒューマノイドを歩かせるというオンライン歩行プランニングを世界で初めて達成した[Kagami2002]。ここでは、次の1歩をどこに踏み出すか、一歩分の動作期間中の下半身の約15の関節の角度とトルクの軌跡を数十ミリ秒で計算している。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と歩行できないのは、1秒分の動作をあらかじめ作っているため、接地時に受ける予測できない力に応じたコンフィギュレーション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、よりロバスタな歩行を実現するには、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では、 $5m \times 5m$ 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

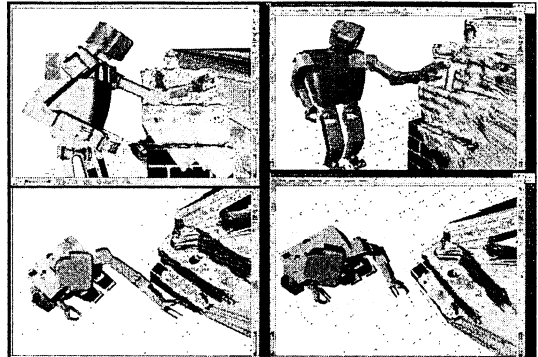


図4. 視覚で環境を認識し、障害物を避けながら(ペットボトル)を把握する動作の計画

パターンを計算する。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、たとえば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、1mあたり5万の探索枝の処理が必要であり、1.4GHz Pentium-IIIで1秒を要する [Chestnut2003]。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る。隣の部屋に行くより長い経路の探索では、より大きな計算パワーが必要となる。

3.2. 認識

ロボットが環境に応じて自立的に動作を計画したり、変化する環境の中でも安全に歩行を行うには環境の認識が必要である。また、物体把握、動体追跡などの動作、人との音声対話などの能力を発揮するためには、特にパターン認識処理に大きな処理能力を必要とする。

最も大きなCPUパワーを消費する認識処理は、視覚である。従来、画像理解のためにはエッジ、テクスチャ、面の法線などさまざまな特徴に基づく手法が提案されたが、3次元世界で動くロボットでは、両眼立体視による距離画像を用いる方法が、最も有力である。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、Pentium-III@1.4GHzプロセッサでMMX命令を駆使して、20ms程度である。

上記速度は、毎秒50枚の距離画像生成が可能であるが、解像度が十分でない。一般的な35mm(35ミリカメラ換算)の準広角レンズを用いると、320×240の画像は、人間の視力0.1に相当する。4倍の1280×960にすれば視力0.4に相当するが、処理能力は16倍必要となる。35mmの視野角も十分ではない。

音響処理は視覚処理に次いで大きな処理能力を必要とする。その内容は、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は、一秒間の音声の生成に、Pentium-III@1GHzで0.1-1秒を要する。

³ 平面上で点の動きを計画するには2次元、回転する多角形であれば3次元。3次元空間中での質点の動きは3次元、回転する剛体の動作計画は6次元。多関節物体の場合は、関節ごとに1次元ずつ上昇する。

3.3. 実時間実行

HRP-2は、1個の1.2GHz Pentium-IIIを歩行制御に、もう一つを画像処理等に用いている。第一のプロセッサの能力は、歩行パターン生成に20%、IO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、5msごとの関節目標値が生成される。この目標値を達成するべく、各関節では1msごとの位置サーボを実行する。

一般に、サーボ周期が短い方が系は安定し、サーボゲインをあげることができる。しかし、パワー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。現在まで、すべてのヒューマノイドロボットは、位置制御を行っている。減速比が大きいので、通常の力制御を適用することは困難であるが、力をセンサ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そって一定の力で押しつけながら滑らさない動作等も実現できると予測される。そのために、目標値の生成を1ms、サーボを0.2msで行うことを目標とする。

3.4. 通信・管理

上記の3種類の処理の他に、プロセス間の同期と通信、名前空間の管理やディレクトリサービス、メモリ等の資源管理、デバッグ支援など、オペレーティングシステムやライブラリが受け持つ広範な処理が必要となる。重要なのは、実時間処理をサポートすること、すなわち決められた時刻にプログラムが走ることで、予測時間内に処理が終わることが保証できることである。実時間OSとして、H7では、RT-Linuxを、HRP-2ではART-Linuxを使用している⁴。3.3で述べた0.2msのサーボ周期をロボット全体で同期をとって実行するためには、システムソフトレベルで0.1ms以下、ハードウェアで0.05ms以下の遅延に抑える必要がある。

一般に、処理が必要になることにプロセッサを割り当てるリアルタイム処理は、処理要求が複数たまった後にまとめて処理を行うUnix方式に比べてトータルのスループットが低下する。したがって、システムコールやタスクスイッチの無駄時間を最小限にしなければならない。

ヒューマノイドロボットの情報処理系は、固定的なプログラムを実行するのではなく、多くの研究者がそのソフトウェアを開発するプラットフォームとしての価値も持つ。次第に大規模なシステムに成長するソフトウェアをモジュールに開発していくために、インタフェース仕様を明確にしつつ実装の細部を隠蔽するオブジェクト指向のフレームワークが有効である。HRP-2では、オブジェクト指向型分散プログラミング標準であるCORBAに則ったOpenHRPをソフトウェアプラットフォームとしている[金広2003]。本プロジェクトにおいてもこのフレームワークは維持する。

4. 分散情報処理によるヒューマノイドロボット

⁴ HRP-1、ASIMO、MorphはVxWorks、SDR-4XやQRIOはAperios、HOAPはRT-Linuxを使用している。

神経系の構築

2節に述べた諸問題を解決し、3節に述べた諸要件を実現するには、集中型の情報処理系を改め、分散系として再構築することが有効と考えられる。その理由は、信号線長を短縮して信頼性を改善できること、プロセッサ数を増やすことでスケラブルな処理能力が得られるからである。ロボットを分散制御しようとする試みは多数あるが[神村2001、於久2002]、本論文で提案する手法は、プロセッサチップから根本的に新設計することで、実時間通信に基づく実時間OSを実装し、並列・分散処理を生かした高度の計画・認識処理を可能にすること、高い信頼性の実現、低消費電力などを特徴とする。

4.1. 信頼性・耐故障性

信頼性向上の常套手段は、①障害が起りやすい部分を識別し、最低限に抑えること、②系を分割して局所的な障害が全体におよばないようにすること、③重要な部分は冗長系とすることである。もちろん、障害の発生を検知する手段が備わっていることが必要条件である。

①の観点から、障害が起りやすい部分は、長い配線やコネクタにある。制御プロセッサをセンサーやアクチュエータの近傍に配置することにより、配線の長さおよびコネクタ接合部を減らし、信頼性を向上させることができる。アナログ配線長が短縮できることにより、SN比も改善される。②の系の分割は、分散系の本分である。機能分散よりも負荷分散方式の方が故障に対して頑健になるが、ロボットの身体が手、足などの機能に分割されている以上、機能分散の導入が不可欠である。③の冗長性としては、ネットワークを2重系にするほか、プロセスのマイグレーションを可能にする。

障害の主たる原因である信号配線長を最小にし、分割数と冗長性を高めるために、分散の度合いは、2腕2脚の単位ではなく、約30の各関節単位とすべきである。すなわち、プロセッサをモータモジュールと一体化することで信頼性、メンテナンス性を最大にする。そのためには、プロセッサモジュールにサーボ、通信、センサ情報収集の機能を持たせ、モータと同等のサイズに収める必要がある。

4.2. スケラブルな処理能力

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と高速化、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によってきた。しかし、ゲート長が90nmに達するに及んで漏れ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。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やクラスターのような粗結合並列化が有効な手段と見なされつつある。すなわち、プロセッサ内で複数のスレッドの並列実行、メモリを共有しない複数のプロセッサでの分散処理である。

スレッド数やノードの数に応じて計算パワーが得られるかどうかは、対象問題の並列性に依存する。3節で述べた動作計画、認識処理について並列性を検討した。動作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

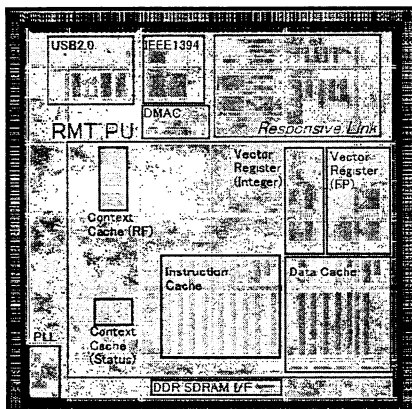


図 5. Responsive Multi-Thread プロセッサ
0.13 μm , CMOS8 層, Cu 配線, 約 1,000 万ゲート、
10 \times 10 mm, 約 1000 ピン

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が可能である。認識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 のような SIMD 命令によって並列化が可能である。音声認識における音素系列の探索処理も並列性がある。

4.3. ノード間通信

並列・分散処理が、高い性能を発揮するには、ノード間の通信が高速で行われることが必須である。特に、ヒューマノイドロボットへの適用においては、実時間通信が保証される必要がある。次に述べるレスポンスリンクによる実時間通信を採用する。

5. プロセッサの仕様

5.1. RMT プロセッサ(RMTP)

4 節の要件を満たす分散処理系の実装には、新たな ULSI を設計する必要がある。小型化の要求が強いサーボプロセッサと、大きな処理能力を必要とする認識・計画用プロセッサは、必要な特性が大きく異なるので、同様なアーキテクチャに基づく 2 種類のプロセッサとして設計する。それぞれを μ RMTP と HP-RMTP と呼ぶ。RMT とは、Responsive Multi-Thread の略である[山崎-URL]。

この二つの新プロセッサには、RMTP を元としている⁵。RMTP は、全て新設計のプロセッサ・システム LSI である(図 5)。命令セットアーキテクチャを MIPS 互換とし、4 命令同時発行、512 本のベクタレジスタ、ベクタと SIMD による並列計算、8 本の優先度付きマルチスレッドの並列実行[伊藤 2003]、コンテキストキャッシュにより 40 スレッドまでを高速でスイッチできる細粒度マルチスレッド制御、レスポンスリンクおよび各種 I/O を集積している。[山崎 2003, Yamasaki2004]。並列機能が最大に働いた場合は、Pentium-4

⁵ RMT プロセッサは、科振費によるセンサーネットワークプロジェクト 2000-2004 において慶應義塾大学が開発中である。

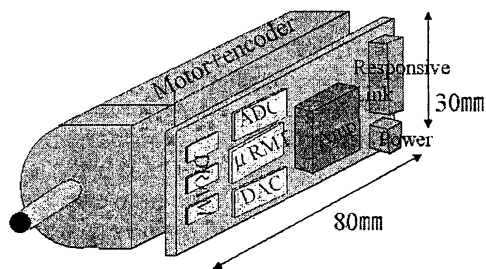


図 6 μ RMT モジュール

の数倍の処理能力を発揮する。並列計算によって性能を上げているため、クロックを 300MHz に抑えることによって、わずか 8W の消費電力でピーク動作する。使用しないユニットを動的に制御することにより、0.5W 程度での小電力動作も可能である。

5.2. レスポンスリンク(RL)

RMTP は、実時間処理を目的にしている。プロセッサ内では、スレッド切り替えが 4 サイクルで実行できる機能、40 本のスレッドのコンテキストキャッシュを備えることなどがそれに当たる。また、ロボット等を分散制御するために、レスポンスリンクと呼ぶプロセッサ間の実時間通信を実現している[山崎 2001]。

レスポンスリンク(Responsive Link, RL と略)は、1998 年に提案し、1999 年に Sparc コアや IO と共にレスポンス・プロセッサとしてシステム LSI 実装を行った⁶。その後、上記 RMTP の製作に当たって、通信速度を引き上げた実装を行っている[山崎 2004]。

レスポンスリンクは、次のような特徴を有する。

- 通信路は、高スループット通信向けのデータリンクと、低遅延通信向けのイベントリンク(各々は双方向)から構成される
- データリンクは、64 バイト固定長パケット、イベントリンクは、16 バイト固定長パケット
- パケットは、優先度を持ち、ルーティングノード(RMTP 等)において優先度に基づく追い越し制御をハードウェアで行う
- パケットごとにハードウェアによる通信エラー訂正を行う
- 通信速度は、最高 800Mbps であるが、信頼性向上と低消費電力化のために 400, 200, 100Mbps にスイッチすることが可能

各ノードには 4 つの RL ポートが実装され、ルーティングは、各ノードで行う。32 ビットのヘッダを読み込んだだけでルーティング先が決まるので、ノード 1 段の遅延は 160ns、ロボット全体では 3 マイクロ秒程度で通信可能である。数百程度までのノードを低遅延かつ高バンド幅でネットワークすることが可能である。

⁶ レスポンスプロセッサは、旧電子技術総合研究所において、通産省国際標準創生事業および科振費 COE プロジェクトの中で開発された。レスポンスリンクは、2003 年に情報処理学会の学会試行標準が成立し、現在 ISO/SC25 に国際標準化の提案を行っている[RL 標準 URL]。

5.3. μ RMT プロセッサ(μ RMTP)

μ RMTP は、RMTのアーキテクチャを継承し、特に物理寸法を極小にし、モータモジュールに一体化してサーボを実行するプロセッサである。処理性能の拡大よりも、モーターの側面に取り付けられるように、アンプを含めて $3 \times 8 \times 1$ cmの大きさに収めることが主要な目標となる(図 6)。

μ RMPTが目的とするサーボ処理は、I/Oが中心であり、並列処理によって性能を高める必要性は少ない。そのため、 μ RMTPの命令パイプラインは2並列にとどめ、ベクトルユニットは必須とはしない。認識、計画処理を μ RMTPに負荷分散させることも可能となるようマルチスレッド機能は実装する。

CPU、メモリ、アナログ回路を1チップに収めることは不可能である。一方、これらを別チップとするとピン数が1000本程度に達し、 μ RMTPのダイは小さくてもパッケージが過大な寸法となる。プロセッサ、メモリ、モータ駆動回路、ネットワーク(レスポンス・リンク)コネクタなどをSIP(system in package)として実現する(図 6)。

処理性能に寄与しないのに物理寸法に大きく影響する要素に信号線のコネクタがある。コネクタを極小化するには、入出力線の数を抑えなければならぬ。I/Oは、ACサーボモータの制御に必要な機能だけに限定する。以下に仕様概要を示す。

1. スケール: 0.13 μ m, コア 7-8mm角, SIP 3×8 cm角
2. 電源: コア 1V, 周辺 2.5V, 消費電力 0.5W
3. CPU性能 133MHz, dual thread
4. チップ搭載のIO
 - (ア) レスポンス・リンク \times 4
 - (イ) エンコーダ、カウンタ、(32bit) および PWM 発生器 12ch
 - (ウ) DA 6ch (0-2.5V)
 - (エ) Serial (RS232c) 1-2ch
6. SIP 搭載の機能

メモリ 32bit DDR SDRAM PC2100 64MB ECCなし、ブート用 Flash ROM 数MB、AD 8bit \times 10ch、モータ駆動アンプ

5.4. HP-RMT プロセッサ (HP-RMTP)

HP-RMTPは、時期的には μ RMTPの次に設計予定なので、より新しい(微細な)スケールを利用できる。クロック速度は4-500MHzとし、並列度をさらに上げることによってRMTの2倍以上の性能を発揮する予定である。TVカメラが直接接続できるようにIEEE1394インタフェースを内蔵する。また、ホストとなるPCとインタフェースするために、PCIバス、USBクライアントを内蔵する。AD変換器は外付けとする。

6. 実時間分散系基盤ソフトウェア

レスポンスリンクで結合される約40のプロセッサで、ロボットを実時間・分散制御するためのオペレーティングシステム等について述べる。

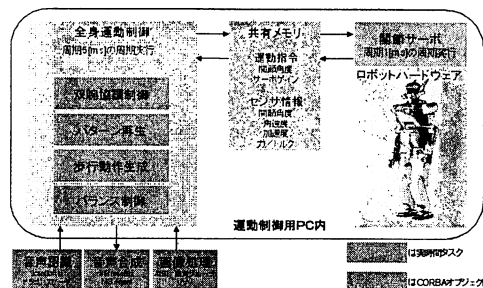


図 7 HRP-2のソフトウェア構成

6.1. 実時間オペレーティングシステム

HRP-2は、ART-LinuxにCORBAを搭載し、その上で図7に示すようなロボット制御のためのプロセスを実行している。ソフトウェアの継承の観点から、オペレーティングシステムとしてLinux2.6をベースに実時間スケジューリングの拡張を行う。すなわち、システムコールコストを規定し、応答遅延時間をバウンドする。3.3で述べたように、ロボット全体で100マイクロ秒以下の応答性が必要である。

現在のHRP-2のソフトウェアが大きく5つのプロセスで実行されるのに対し、新しい分散系は30を越えるプロセッサを持つ。従って、1プロセッサ上で同時に動作するアプリケーションプロセスは、たかだか数個であり、システムスレッドを含めてRMTプロセッサがハードウェアでサポートするスレッド数40に収めることができる。

実時間OSでは、しばしば優先度逆転が問題にされる。一つのプロセッサ上のプロセス数が小さい状況では、ユーザプロセス間の優先度よりもカーネルプロセスおよび他のプロセッサ上のプロセスへの優先度の継承が重要である。

これらの優先度継承が困難になるのは、I/O処理が、ユーザプロセスとの関係が明確でない割り込みハンドラで取り扱われることに起因する。そこで、デバイスハンドラは、割り込みではなく、周期タスクとして実現することを計画している。【石川 1989, Ishikawa1992】。

6.2. 通信・分散制御

レスポンス・リンク上のソケットインタフェースを実現する。レスポンス・リンク上のデータとイベントリンクの違いは、ユーザから隠蔽する。メッセージには優先度を付けることができる。フロー制御は行いが、レスポンス・リンク上がECCを内蔵しているためソフトウェア・プロトコルによるエラー訂正は行わない。耐故障性を高めるために、断線とパケットの未達を検出し、1ms以内に自動的に再ルーティングする機能を実現する。

CORBAは、omni-ORB over RLを実装する。また、Real-Time CORBA標準の機能を実現する。ネットワークワイドの実時間処理のために、グローバル時刻合わせ、グローバル周期プロセス実行・同期、リモートメモリアクセス、リモートシグナル、リモートプロセス起動などを実現する。

7. ヒューマノイドロボット実装

現在、HRP-2 のプロセッサとネットワークを簡単な分散系で再構成する HRP-3 プロジェクトが進行中である。本プロジェクトの成果は、この HRP-3 ロボットの神経系の置き換えになる。

開発は、段階的に進める。 μ RMT プロセッサとモジュールの試作および RMT アーキテクチャ上の実時間 Linux 開発が 2004 年、HP-RMT 試作が 2005-6 年、テストベッド上での試作と並列化プログラミングが 2006 年、2007 年に HRP-3 上に統合し、2008 年には、ネットワークの断線やプロセッサノードの故障があっても安定性が損なわれないこと、レスポンスな実時間処理に基づく高精度力制御、高性能実時間処理に基づく高度な認識処理やオンライン動作計画機能をデモすることを計画している。

8. 結論

ヒューマノイドロボットのスケラブルな能力拡大と信頼性向上をねらって、その情報処理系を分散系とする方式について述べた。各関節に約 30 の μ RMT を配置してサーボを行うことで信号線長を最低限に抑え、動作計画、認識、対話用には、中央に HP-RMT ノードを数個配置する。全体を二重のレスポンス・リンクで結合し、実時間通信を行う。ソフトウェアは、Linux2.6 に実時間向けの拡張を施す。分散プログラミング環境として CORBA の実時間拡張を行う。100 マイクロ秒オーダのハードリアルタイム制御を保証し、1ms で生成される軌道を 200 マイクロ秒周期でサーボする。ネットワークの断線が生じて 1ms 以内にリコンフィギュレーションを行い、転倒の危険性を最小限に抑える。結果として、ヒューマノイドロボットは高度の認識、計画等を駆使し、高速サーボによる力制御、不整地歩行、全身動作、転倒回避などを可能とする。

謝辞

本研究は、科学技術振興機構(JST)の戦略的創造研究推進事業 CREST において 2003 年秋から開始した。JST および研究総括の東京大学田中英彦教授ならびに領域アドバイザーの方々のご支援に感謝申し上げます。

参考文献

- [Hirai1997] K. Hirai, "Current and Future Perspective of Honda Humanoid Robot," Proc. IEEE/RSJ Int. Conf. On Intelligent Robots and Systems (IROS-97), pp. 500-508, 1997.
- [五十棲 2004] 五十棲隆勝、赤池一彦、平田勝、金子健二、梶田秀司、比留川博久、「ヒューマノイドロボット HRP-2 の開発」、日本ロボット学会誌 Vol. 22, No. 1, 2004.
- [Honda-URL] <http://www.honda.co.jp/robot/>
- [Moravec2000] Hans Moravec, ロボットは人間を越えるか、日経サイエンス Vol. 30, No. 1, pp. 104-111, 2000.
- [Kagami2002] Satoshi Kagami et al., "A Fast Dynamically Equilibrated Walking Trajectory Generation Method of Humanoid Robot," Autonomous Robots 12, pp. 71-82, Kluwer Academic Publishers, 2002.
- [Chestnut2003] Joel Chestnut, James J. Kuffner, Koichi

Nishiwaki, Satoshi Kagami, "Planning Biped Navigation Strategies in Complex Environments," IEEE International Conference on Humanoid Robots (Humanoids2003), Oct., 2003

- [金広 2003] 金広、藤原、梶田、横井、金子、比留川、中村、山根、ヒューマノイドロボットソフトウェアプラットフォーム OpenHRP、日本ロボット学会誌、Vol. 21, No. 7, pp. 89-97, 2003.
- [金広 2004] 金広、金子、藤原、他、「ヒューマノイドロボットの転倒回復機能の実現」、日本ロボット学会誌、Vol. 22, No. 1, pp. 37-45, 2004.
- [神村 2001] 神村明哉、村田智、吉田英一、黒河治久、富田康治、小鍛治繁、「自己組み立て可能なモジュール型ロボットシステムに関する研究(複数ユニットによる変形・移動実験)」、日本機械学会論文集 68 巻 667 号 C 編、pp. 174-180, 2001.
- [於久 2002] 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 研究報告「計算機アーキテクチャ」、No.150-010, 2002.
- [山崎-URL] <http://www.ny.ics.keio.ac.jp/>
- [伊藤 2003] 伊藤務、山崎信行、「Responsive Multithreaded Processor の命令実行機構」、情報処理学会誌コンピューティングシステム(ACS), Vol. 44, No. SIG11, pp. 226-235, 2003.
- [山崎 2003] 山崎信行、堀俊夫、「分散リアルタイムネットワーク用プロセッサとその応用」、情報処理、Vol.44, No.1, pp.6-13, 2003.
- [山崎 2004] 山崎信行、「分散制御用リアルタイム通信 Responsive Link の設計及び実装」、情報処理学会論文誌コンピューティングシステム(ACS), Vol. 41, No. 5 (to appear), 2004.
- [Yamasaki2004] Nobuyuki Yamasaki, "Design Concept of Responsive Multithreaded Processor for Distributed Real-Time Control, Journal of Robotics and Mechatronics (to appear).
- [山崎 2001] 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.68-77, 2001.
- [RL 標準-URL] <http://www.itscj.ipsj.or.jp/ipsj-ts/02-06/toc.htm>
- [石川 1989] 石川裕、Tokuda, Mercer, "Priority Inversion in Protocol Handling" 日本ソフトウェア科学会第 6 回全国大会予稿集、pp. 357-360, 1989.
- [Ishikawa1992] Y. Ishikawa, H. Tokuda, C. Mercer, "An Object-Oriented Real-Time Programming Language," IEEE Computer Vol.25, No. 10., pp. 66-73, 1992.