

「情報社会を支える新しい高性能情報処理技術」

平成 15 年度採択研究代表者

松井 俊浩

(独) 産業技術総合研究所デジタルヒューマン研究センター・副センター長

ヒューマノイドのための実時間分散情報処理

1. 研究実施の概要

本プロジェクトは、ヒューマノイドロボットの性能と信頼性の向上のため、従来の少数のプロセッサによる集中制御に替えて、30 以上のプロセッサによる分散制御を実現することを目的とする。それによって、処理能力のスケラブルな拡張可能と、マルチスレッドとベクタユニットによる並列処理を活用することで、高い処理性能を低い消費電力で実現する。プロセッサをセンサーやアクチュエータの近くに配置することで、信号線長を削減し、高い信頼性を実現する。一方、従来のモノリシック制御を分散化させるため、通信およびマルチスレッドを実時間化し、優先度や資源割り当て等を行う基盤ソフトウェアが必要になる。

本研究は、次の三つから構成される。

①RMTP の設計・開発

実時間性にすぐれたマルチスレッド向きアーキテクチャ、低クロックでも性能を発揮するための並列性、実時間通信チャネルを備えたプロセッサを開発する。

②実時間基盤ソフトウェア

Linux-2.6 を拡張して実時間スケジューリングを可能にする、またプログラムの実行時間を静的に予測する技術の研究。

③ロボット実証

認識、対話、歩行、経路計画などの機能をモジュールに構成することで分散化し、並列処理を活用して性能向上を図る。

前年度までに、 μ RMTP の第 1、2 次チップ試作、単独および SiP パッケージ化、PCI 評価ボード実装、RMTP ソフトウェアシミュレータ開発、RMTP への Linux カーネル移植、X86 アーキテクチャでの実時間 Shi-Linux の開発などを行った。2007 年度は、 μ RMTP の割り込み、投機実行、ベクトル

計算等に問題が判明したため、プロセッサロジックの検証を継続し、命令レベルシミュレータおよび RTL シミュレータで Linux が起動することを確認した。SIP を再設計すると共に、SIP のメモリ特性に応じて RMTP のメモリインタフェースの最適化を図った。また、シミュレータを用いて実時間 Linux 開発、CORBA 開発を行った。実時間 Linux の成果が RMTP 以外でも活用できるよう、X86 プロセッサへの実装、特に SMP モードで利用できるよう ART-Linux の 2.6 への移植を行った。ロボットアプリケーションプログラムを開発するため、C コンパイラの整備、C++への対応、ライブラリ構築を行った。1 次試作ボードを用いて、ブラシレスモータ制御プログラムを試作し、実時間性を検証した。

2. 研究実施内容

2.1 RMT-LSI グループ

以前のバージョンで見つかったプロセッサコア(RMT PU)のバグを解消し、分散リアルタイム処理用プロセッサである μ RMT Processor Ver.3 (M-RMTP Ver.3)の設計・検証、システムインパッケージ(SiP)化、および SiP 評価ボードへの実装を行った。M-RMTP Ver.3 には、以下に述べるリアルタイム計算、リアルタイム通信、コンピュータ用ペリフェラル、ロボット向き制御 I/O 各機能を System-on-chip として実装している。

- リアルタイム処理機能:RMT PU
- リアルタイム通信機能:Responsive Link
- コンピュータ用周辺機能:PLL (1GHz, Clock control), DDR SDRAM I/F (32bit/128bit), ROM I/F (8bit/16bit/32bit), SRAM (256bit, 256kB), PCI-X (64bit, 66MHz), IEEE-1394a (AV link), Ethernet (MAC), RS-232C x 2, 32bit DMAC 4ch x 4, 256bit DMAC, General purpose bus I/F (32bit, CS x 2, irq x 2, dreq x 2, breq x 1)
- 制御用周辺機能:PWM generators (32bit, 100MHz) x 6, Pulse counters (32bit, ABZ phase, AC/DC motor modes) x3, PWM inputs (32bit) x 3, SPI (SPI + cs x 3) x 2, Digital I/O ports 15bit

M-RMTP Ver.3 は、バックエンド設計までを慶應義塾大学で行い、ファブリケーションは、TSMC 0.13 μ m LV-FSG(CMOS 8 層銅配線)プロセスで行った。ゲート数は約 14M ゲート、ダイサイズは 9,980 μ m x 9,980 μ m = 99.6mm²、Core 電圧 1.0V、I/O 電圧 3.3V (5V tolerant)、チップの I/O はフリップチップである。消費電力は、ソフトウェアとハードウェアの電力制御機構により 0.1~4W と可変である。レイアウト図とチップ写真を図1に示す。

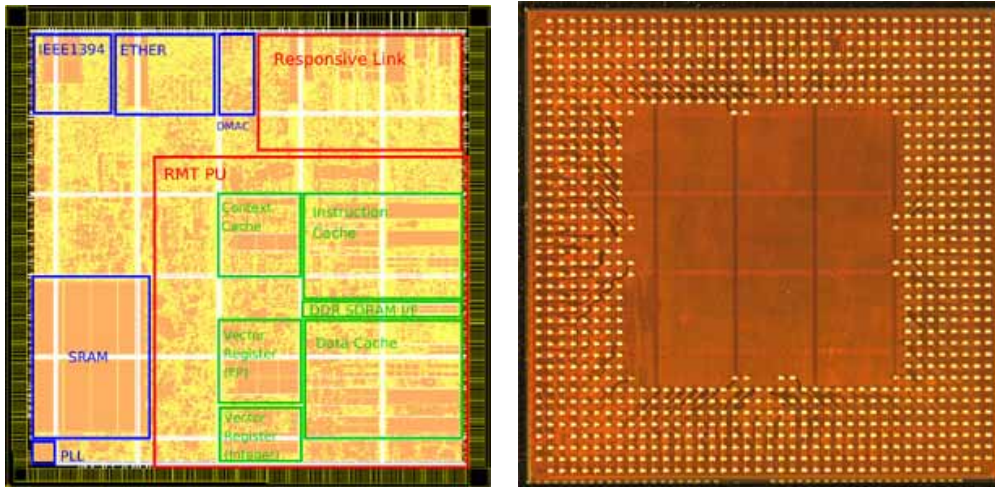


図 1 M-RMTP Ver.3 のレイアウトおよびダイ写真(フリップチップ)

M-RMTP ベアダイ、DDR SDRAM、フラッシュメモリ、Ethernet Phy 等を搭載したシステムインパッケージ(SiP)である M-RMTP SiP および M-RMTP SiP の PCI 評価ボードを設計・実装した。M-RMTP SiP は上下 4 層の最先端のビルドアップ基板(最細線幅 $20\mu\text{m}$)で実現している。PCI 評価ボードは、PCI フルサイズの 12 層基板で作成し、DC-DC 電源(2.5V, 1.0V)、ブート ROM、Responsive Link コネクタ(RJ45 x 2)、64bit 66MHz PCI エッジ、RS-232C、信号を計測するための Mictor コネクタ等を配置している。

開発環境として、M-RMTP 用のクロスコンパイラの改善を行った。具体的には、オリジナルの M-RMTP のバグを回避するために、アセンブラレベルで特定のコード領域に命令を置かないようにすることが可能となった。また、特定の命令に対しては M-RMTP のアウトオブオーダー実行をしないように sync 命令を自動的に挿入することが可能となった。分岐命令に関しても、条件分岐が連続して発生すると分岐予測が失敗してアウトオブオーダー実行が期待通りに動作しないというバグを回避するコード(1 回目の条件分岐を無条件分岐に置き換える)を生成することが可能となった。なお、これらのバグは今年度作成した上記の M-RMTP Ver.3 では全て解消されている。

軽量 OS として、モータ制御用の軽量実時間カーネル TFlight の開発を行った。Linux は汎用的な目的のために作られた OS であり、M-RMTP の機能を十分に発揮させるようにカスタマイズすることは簡単ではない。そこで、モータ制御のようにソフトウェアであまり複雑なプログラムを組む必要のないアプリケーションに対して、M-RMTP のハードウェア実時間処理機能を十分に活用し、実時間性を向上させることを目的とした軽量な実時間カーネル TFlight を開発した。TFlight は Linux よりも遥かに精度の高い周期実行が可能であり、M-RMTP 上で $50\mu\text{s}$ 以下の周期実行も可能である。また、M-RMTP の優先度付き SMT 機能や同期機構を有効利用して、ソフトウェアで複雑な処理をすることなく、八つまでのタスクであれば並列実行することができ、高速な同期処理が可能である。さらに、実時間処理に適したプログラミングインターフェイスを提供している。Linux ではプログラマ自身がソフトウェアタイマを設定したり、ループとスリープを使って周期処理を実現したりする必要

があるが、TFlight ではプログラマはタスク本体の関数ポインタと周期値を指定するだけで、OS が自動的に周期処理を行うようになっている。

2.2 実時間分散系基盤ソフトウェア

2.2.1 WCET 予測ツール RETAS

2.2.1.1 RETAS の概要

安全で信頼性の高い実時間システムを構築するには、実時間タスクの実行がデッドラインを満たすことを保証する必要がある。一般には統計的に実行時間の最悪値を求めるため、デッドラインを満たすことは保証されない。プログラムの実行時間を予測するために、GCC(GNU Compiler Collection)を基にしたツールである RETAS を開発している。RETAS はソースコードレベルでタスクの全ての実行フローを解析し、最悪の実行時間パスを求めて、実行時間の最悪値を保証する。さらに、RETAS は、様々なアーキテクチャへ容易に移植できることを目指している。アーキテクチャの違いは GCC の中間表現を利用して吸収する。

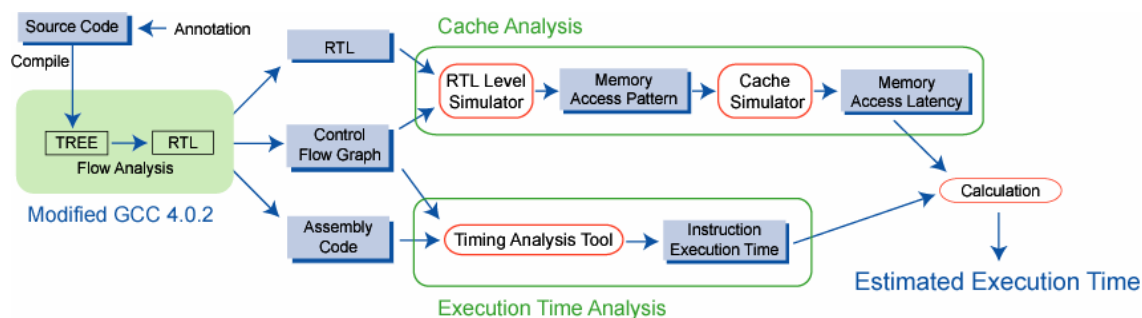


図 2.2.1 実行時間予測システム RETAS

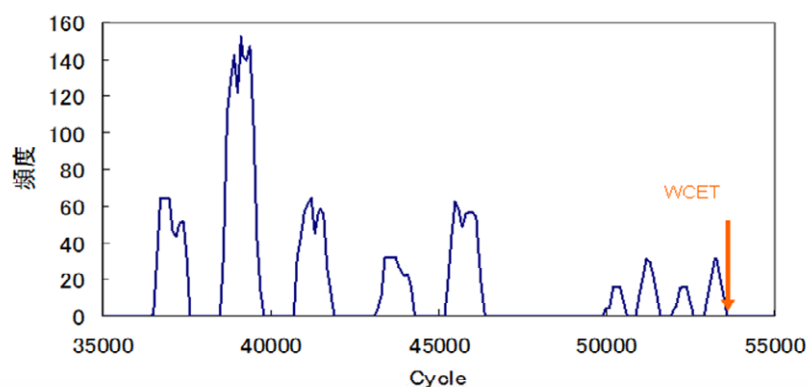
RETAS の構成を図 2.2.1 に示す。予測対象のソースコードから、RTL(Register Transfer Language)、Control Flow Graph(CFG)、アセンブリコードを生成する。RTLはGCC内部で使用されるアーキテクチャに非依存な中間表現である。GCC内部にフロー解析器を実装し、ソースコード中の全ての実行フローを表現したCFGを生成する。プログラムの実行時間は、メモリアクセス命令の実行時間とメモリアクセス以外の命令実行時間に分けて計算する。メモリアクセス命令の実行時間は、キャッシュの振る舞いに依存する。RTLとCFGからメモリアクセスパターンを求めるRTLレベルシミュレータ、メモリアクセスパターンからメモリアクセス命令の実行時間を求めるキャッシュシミュレータによって計算される。メモリアクセス以外の命令実行時間は、CFGとアセンブリコードから基本ブロック毎の実行時間を計測するコードを生成し、ターゲットアーキテクチャで実行、計測される。メモリアクセスの時間および基本ブロックの実行時間からプログラム全体の実行時間を予測する。

2.2.1.2 評価

ロボット実証グループが開発したセンサー・サーボループプログラムによる評価を行った。センサー・サーボループは、センサーの状態取得、サーボの設定を行う周期タスクである。センサーおよびサーボに対する I/O デバイスアクセス部分(read/write システムコール)を除く核部分は、C 言語で 300 行である。I/O デバイスアクセス部分を除いて最悪実行時間を予測した。本コードでは、分岐数 15、ループ数 11 であり、実行可能経路は 3456 通りとなる。Thinkpad X31 (Pentium-M 1.4GHz 768MB)上で 461 秒かかる。Pentium-M における最悪実行時間予測と実測結果を以下に示す。

実測値(クロック)	最悪実行時間 予測値(クロック)	割合(%)
23373	53473	229

下のグラフは、予測した実行時間のヒストグラムである。



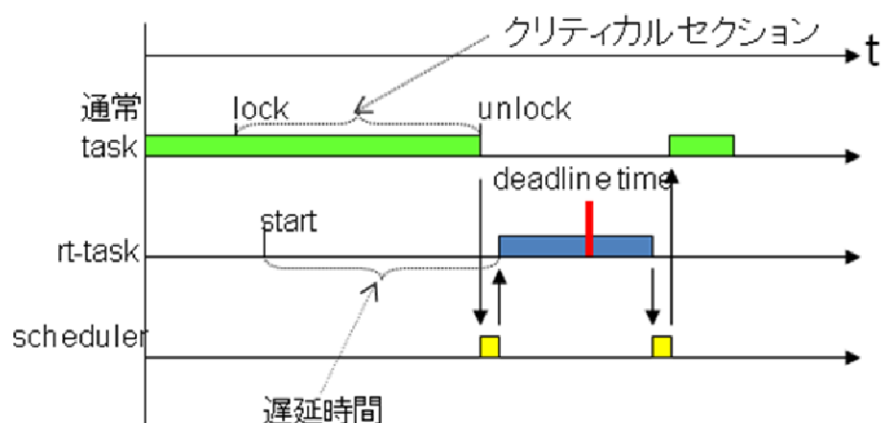
実測値は、予測した最良の実行時間よりもさらに速く実行が終了している。これは二つの原因がある。一つは、本プログラムは浮動小数点演算が含まれているが、データシートに記載された値と実測値とで 2 倍以上の差があること。もう一つは、主記憶のアクセス時間が、実測値では 120 から 220 クロックと変動するのに対して、予測値には最悪値の 220 サイクルとしていることに起因する。

2.2.2 実時間 Linux カーネル Shi-Linux

ロボットなどのハードリアルタイムシステムでは、タスクの実行はデッドライン時間内に終わらなければならない。しかし、通常のカーネルでは、デバイスからの割り込みが発生すると、デッドライン時間に近づいているタスクがあるにもかかわらず、タスクの実行が中断され、割り込み処理が行われる。このため、実時間タスクのデッドラインをミスする可能性がある。本研究で実装している Shi-Linux では、これまでの一般的な実時間 Linux カーネルと違い、デバイスからの割り込みを禁止し、デバイスドライバを一つの実時間周期タスクとして実現される。これにより、デバイス割り込みによるランダムな実行がなくなり、タスクの実行時間の予測性が増す。実時間タスクスケジューリン

グのために、タイマー割り込みのみを許す。

2006 年度までに開発した Shi-Linux Version 2 では、以下に述べる優先度逆転現象により実時間タスクのデッドラインが保証できないことが判明した。下図に示す通り、通常タスクがカーネル内でクリティカルセクションに入っているときに、タイマー割り込みが生じても、クリティカルセクションを通過するまで割り込みは遅延される。このため、実時間タスクが起動できなく、デッドライン時間を満たせなくなる。いわゆる優先度逆転現象が生じる。



このような状態が、Linux2.6.20 カーネルのどの場所で生じるか調べた。測定に使用したマシンは、Pentium 4 3GHz メモリ 512MB で、シングルモードで起動からプロンプトが表示されるまでの期間である。

ファイル名	開始行	終了行	最大遅延時間(us)
net/ipv6/mcast.c	710	716	150.653
include/linux/netdevice.h	932	953	110.528
drivers/net/e1000/e1000_main.c	3608	3758	83.117
fs/dcache.c	768	816	65.429
drivers/ata/libata-scsi.c	2802	2814	55.564

このように、最大、150 マイクロ秒の遅延時間が生じる可能性があることがわかる。この遅延時間を最小限にするか、あるいは生じないようにするための手法を現在開発中である。

2.3 ロボット実証グループ

RMTP をロボットに適用するため、2006 年度には、関節に組み込むための小型モータ制御モジュールを開発した。2007 年度は、PCI 評価ボード上の RMTP を用いて、モータを制御するプログラム開発を行った。RMTP の割り込みが異常動作するため、マルチタスク構成とはできないが、ブラシレスモータで回転磁界を発生させ、角度、電流フィードバックをかけてモータを制御できることを

確認した。制御周波数は 20KHzで、PWM 切り替え時のデッドタイムは 6%以下に抑えることができた。

ロボットの動作計画の研究を進めた。物体把持のためのフォースクロージャを指標としてプリコンピュテーションしてテーブルを作成し、実際の把持の際には障害物との干渉や胴体からの距離などを鑑みて、物体を把持するソフトウェアを開発した。HRP2 を用いて物体を把持する実験を x86 を用いて行った。

RMTP を組み込むべきヒューマノイドロボットの設計・製作を開始した。別プロジェクトで開発を進めた HRP-3 をベースにして、2007 年度は、HRP-3 改の全身を設計し、上半身を開発した。下半身は 2008 年度に開発する。HRP-3 改は、HRP-3 (川田工業が設計・製作し、産業技術総合研究所がソフトウェアを開発) を元に RMTP 実証機として用いるために一部設計を変更した。上半身は既存の集中制御系を用い、分散処理系はロボットの下半身に搭載する。2008 年度開発予定の下半身分を含めたロボットのシステム構成の概要を図 2 に示す。(同図で、黄緑色は 2007 年度開発済み、ピンク色は 2008 年度開発予定の部分である。)

上半身は、胴部と2腕からなり、それぞれの腕部は7自由度を有し、その先には、3指6自由度のハンドを有する。また、胴部下方には腰関節として2自由度を有し、上方には首関節として2自由度を有する。すなわち、上半身部は計30自由度を有する。(下半身部は12自由度となる。) 上半身部の写真および、全身完成後の形状モデルを図3に示す。

胴部背面にはコンパクト PCI 規格の CPU ボードおよび各種 I/O ボードを搭載し、1[kHz]での各関節の制御を始めとして、各種センサーの入力が行える構成となっており、また無線 LAN により実験環境の LAN と接続することができる。さらに、2008 年度にコンパクト PCI 規格の RMTP 通信ボードを搭載することにより、本プロジェクトで開発している RMTP プロセッサとの通信を行うことが可能となり、2008 年度開発の下半身モジュールとあわせて、ロボット実証実験を行う。

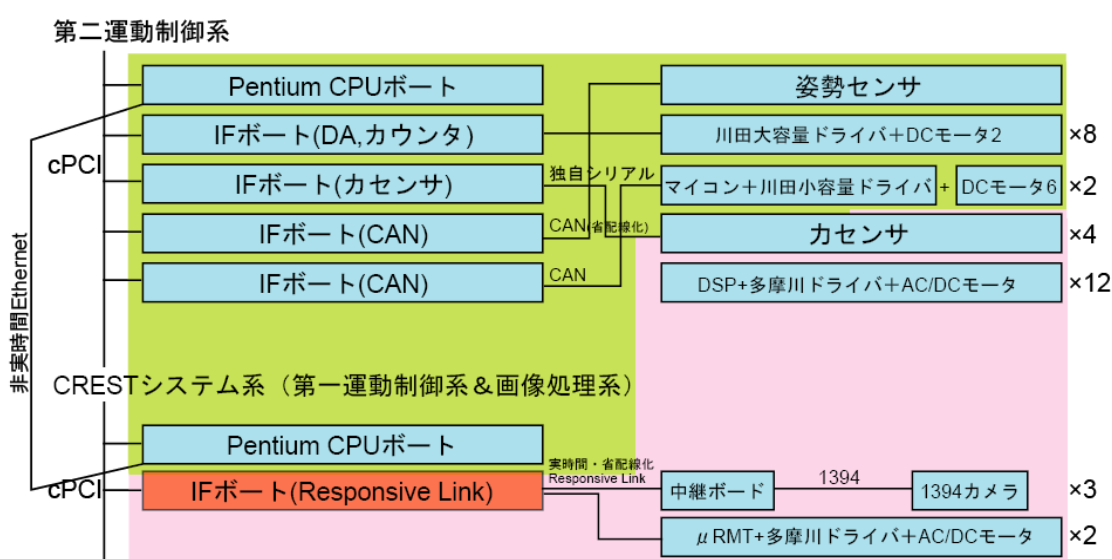


図 2 HRP-3 改のシステム構成図

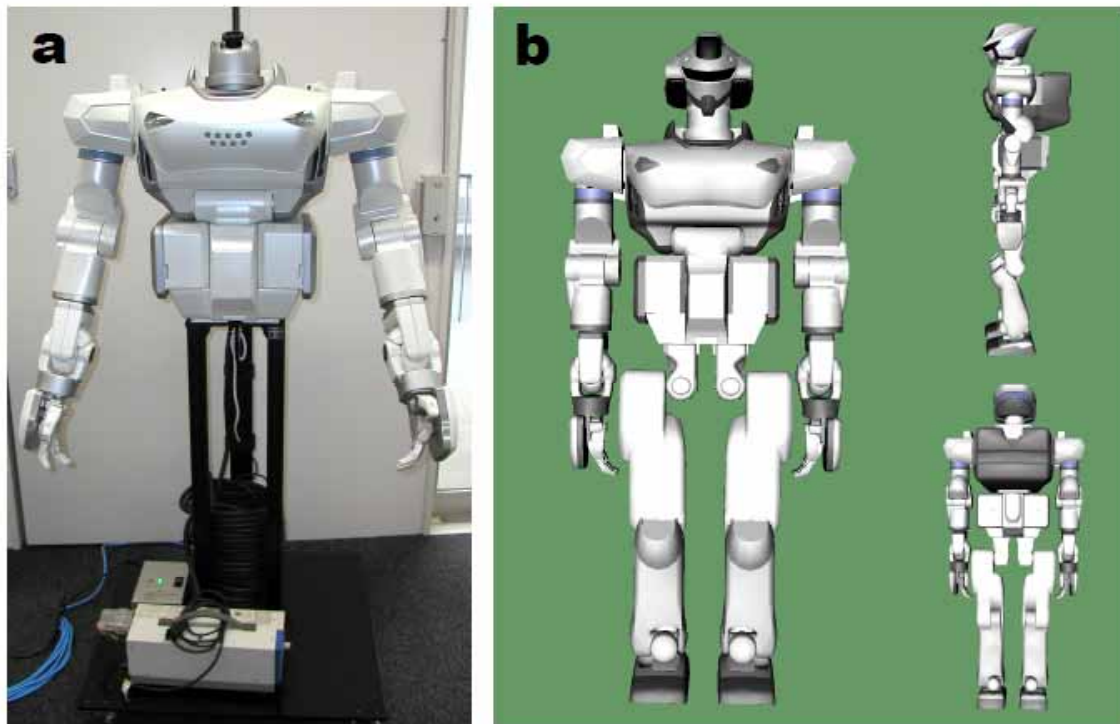


図 3 HRP-3 改の上半身の写真(a)と完成後の形状モデル(b)

3. 研究実施体制

(1)「ロボット実証」グループ

- ①研究者名: 松井 俊浩 (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研究項目
 - ・ ヒューマノイドロボットの分散制御系の研究

(2)「RMT-LSI」グループ

- ①研究者名: 山崎 信行 (慶應義塾大学大学院)
- ②研究項目
 - ・ 実時間通信・処理・制御用システムオンチップの研究開発

(3)「基盤ソフトウェア」グループ

- ①研究者名: 石川 裕 (東京大学大学院情報理工学系研究科)
- ②研究項目
 - ・ 実時間オペレーティングシステムと開発環境の開発

(4)「ロボット実証」グループ

①研究者名:比留川 博久 (産業技術総合研究所知能システム研究部門)

②研究項目

- ・ ヒューマノイドロボットの分散制御系の研究

4. 研究成果の発表等

(1) 論文発表(原著論文)

1. 加藤 真平, 山崎信行, “RMd2-SIP: マルチプロセッサにおける実時間スケジューリングアルゴリズム”, 情報処理学会論文誌: コンピューティングシステム, Vol. 48, SIG 13, pp. 270-286, September, 2007.
2. 加藤 真平, 山崎信行, ポーショニング方式に基づく動的優先度スケジューリング. 情報処理学会論文誌: コンピューティングシステム, Vol. 49, SIG 2, pp.113-126, March, 2008.
3. N. Yamasaki, I. Magaki, T. Itou, “Prioritized SMT Architecture with IPC Control Method for Real-Time Processing”, The 13th IEEE Real-Time and Embedded Technology and Applications Symposium, pp. 12-21, Bellevue, April 3-6, 2007.
4. S. Kato, N. Yamasaki, “Feedback Controlled Server for Scheduling Aperiodic Tasks”, The 4th International Conference on Embedded Real-Time Computing Systems, pp. 402-407, Prague, July 27-29, 2007.
5. S. Kato, N. Yamasaki, “Real-Time Scheduling with Task Splitting on Multiprocessors”, The 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, pp. 441-450, Daegu, August 21-23, 2007.
6. K. Funaoka, S. Kato, N. Yamasaki, “Real-Time Static Voltage Scaling on Multiprocessors”, The 19th IASTED International Conference on Parallel and Distributed Computing and Systems, pp. 142-149, November 19-21, 2007.
7. S. Kato, N. Yamasaki, “Fixed-Priority Scheduling on Prioritized SMT Processor”, The 19th IASTED International Conference on Parallel and Distributed Computing and Systems, pp. 116-123, November 19-21, 2007.
8. K. Funaoka, S. Kato, N. Yamasaki, “Optimal Real-Time Scheduling for Efficient Aperiodic Services on Multiprocessors”, The IASTED International Conference on Parallel and Distributed Computing and Networks, pp. 245-251, February 12-14, 2008.
9. A. Takeda, K. Funaoka, S. Kato, N. Yamasaki, “Soft Real-Time Scheduling with Tardiness Bounds on Prioritized SMT Processors”, The IASTED International Conference on Parallel and Distributed Computing and Networks, pp. 252-257, February 12-14, 2008.

10. Y. Fujita, S. Kato, N. Yamasaki, “Real-Time Communication and Admission Control over Responsive Link”, The IASTED International Conference on Parallel and Distributed Computing and Networks, pp. 131–138, February 12–14, 2008.
11. K. Nishiwaki, S. Kagami, “Sensor Feedback Modification Methods that are Suitable for the Short Cycle Pattern Generation of Humanoid Walking”, Proceedings of IEEE/RSJ 2007 International Conference on Intelligent, Robots and Systems(IROS2007), pp.4214--4220, San Diego, USA, November, 2007.
12. K. Nishiwaki, S. Kagami, “Walking Control on Uneven Terrain with Short Cycle Pattern Generation”, Proceedings of IEEE-RAS International Conference on Humanoid Robots (Humanoids2007), Pittsburgh, PA, USA, November, 2007.
13. M. Takahashi, K. Nishiwaki, S. Kagami, H. Mizoguchi, “Development and Evaluation of An Attitude Measuring System that uses, Acceleration Information of Walking for Biped Robots”, Proceedings of The 6th Annual IEEE Conference on Sensors(Sensors2007), pp.292--295, Atlanta, USA, October, 2007.
14. S. Kagami, S. Thompson, Y. Sasaki, Y. Nishida, H. Mizoguchi, T. Enomoto, ““Calling from the Other Room” by Ceiling Ultrasonic Locator and Microphone Array”, Proceedings of IEEE/RSJ 2007 International Conference on Intelligent Robots and Systems(IROS2007), pp.2028--2034, San Diego, USA, October, 2007.
15. S. Thompson, T. Horiuchi, S. Kagami, “An Environment Driven Model of Human Navigation Intention for Mobile Robots”, Proceedings of The 13th IASTED International Conference on Robotics and Applications, Wurzburg, Germany, August, 2007.
16. D. Miyakawa, Y. Ishikawa, “Process Oriented Power Management”, The IEEE Second Internal Symposium on Industrial Embedded Systems -- SIES'2007, 2007 (CD-ROM)

(2) 特許出願

平成 19 年度 国内特許出願件数:1 件(CREST 研究期間累積件数:4 件)