

8F-1

## パーソナルロボット制御に対する リアルタイムスケジューラの適応性評価\*

飯田 浩二      菅原 智義      矢向 高弘      安西 祐一郎†

慶應義塾大学‡

### 1 はじめに

ロボットのモータ制御等に用いられる周期的タスクや厳しいリアルタイム性の要求されるタスクを効率的に実行するには、時間的制約を考慮したスケジューリングを行なう必要がある。同時にロボットの制御では環境の変化等によって起動されるプロセスを迅速に実行できる必要がある。スケジューラはそれらの要求を満足せねばならず、どのようなスケジューリングアルゴリズムが適当であるかは一概には言えない。

本稿では、現在我々が開発中のパーソナルロボット [5] を対象とした OS  $\mu$ -PULSER[4] へ数種のリアルタイムスケジューラを実装し、それらのロボット制御に対する適応性を比較し検討する。

### 2 スケジューラへの要求

パーソナルロボット用の OS に要求される能力として、次のようなことが考えられる。

- 敏感さと機敏さ  
ロボットの動作を妨害するような事象が起るなど、環境の変化へ敏感にかつ機敏に反応できなければならない。
- 移植性  
パーソナルロボットのハードウェアは、その目的・用途・機能に応じて様々なアーキテクチャを持つと考えられる。従って、多様なアーキテクチャに適合するためには移植性が高くなければならない。
- 柔軟性と拡張性  
我々はパーソナルロボットの一つの形として、「ブロックを組み立てるように構成を変更できるロボット」を考えている。従って、OS は柔軟にその構成を変更出来たり、機能を拡張できなければならない。

我々の開発した OS  $\mu$ -PULSER には、ハードウェアとソフトウェアの割り込みを統合して扱うダイレクトインタラプト機構により起動される、ダイレクトインタラプト駆動スケジューラが実装されており、敏感さと機敏さを備えている。しかし、モータ制御等のための周期的なタスクやデッドラインのような厳しい時間的制約を持つタスクを実行するためには、周期やデッドライン等に基づくリアルタイム性を考慮してスケジューリングを行なう必要がある。今日まで数々のリアルタイムスケジューリングアルゴリズムが提案されており、様々な成果が報告されている。パーソナルロボット用 OS はタスクの時間的制約を満足できるだけでなく、さらに機敏さと敏感さを兼ね備え

る必要があり、それらの要求を満足するようリアルタイムスケジューラを必要とする。

パーソナルロボット用 OS のスケジューラには次のようなことが必要とされると考えられる。

- モータ制御等の周期的タスクを実行出来ること
- ハードデッドラインのタスクを実行出来ること
- 環境の変化によって起動されるタスクを迅速に実行出来ること

このような要求に適当なスケジューリングアルゴリズムを一概に決定することは難しいので、種々のスケジューリングアルゴリズムを実装し実験的に比較・検討する必要がある。

### 3 リアルタイムスケジューラの設計と実装

#### 3.1 スケジューラの設計

種々のスケジューリングアルゴリズムを比較、検討する必要性と、 $\mu$ -PULSER のマイクロカーネル化した構成とから、スケジューラを次のように設計した。

- スレッドの分類  
 $\mu$ -PULSER は計算モデルとしてマルチタスク・マルチスレッドモデルを採用している。リアルタイム性と機敏さ、敏感さを得るため、スレッドを次のように分類した。  
リアルタイムスレッド: 時間制約を持ったスレッド  
イベントスレッド: 割り込み等によって起動されるスレッド  
継続スレッド: 通常の状態で行われるスレッド
- リアルタイムスレッドの記述  
リアルタイムスレッドとして周期的、非周期的スレッドを扱えるようにするため、スレッドの周期、デッドラインといった時間属性を図1の様に記述し、スケジューラに渡す。
- 方針/機構の分離  
 $\mu$ -PULSER ではスレッドのスケジューリングはタスク・スレッド管理スレッド (TTM) により行なわれるが、TTM からスケジューリングアルゴリズムに依存したルーチンの呼び出しを図2で示すような手続きで実装し、種々のスケジューリングアルゴリズムを方針モジュールの変更のみで実現する。スレッドの生成・起動・休止といった処理は、すべて TTM が行なう。

スケジューリングポリシーとして、Rate Monotonic 方式、Rate Monotonic を拡張した Sporadic Server[1] 方式と、最短デッドライン優先方式 [3]、さらに比較のためにラウンドロビン方式を実装する。

\*Evaluation of Real-Time Schedulers applied to Personal Robots

†Kouji IIDA, Tomoyoshi SUGAWARA, Takahiro YAKOH, and Yuichiro ANZAI

‡Keio University

```

struct thread_type {
    int rtype; /* type of rt thread */
    u_int arrivaltime; /* arrival time */
    u_int starttime; /* earliest start time */
    u_int deadline; /* deadline */
    u_int period; /* period of thread */
    u_int calctime; /* worst case calc.time */
}

```

図 1: スレッドの時間属性の記述

```

int policy.init(); /* initialize */
int policy.add( th ); /* sched policy */
int policy.del( th ); /* remove th from policy */
th_queue_t *policy.choose(); /* select next thread */
int policy.end(); /* end using policy */

```

図 2: 方針モジュールの手続きの定義

### 3.2 スケジューラの実装

現在 $\mu$ -PULSER は MC68030 (25MHz) ベースの AVME-130-1 (RAM 8MB、ROM 2MB、シリアルポート×3、パラレルポート×1、イーサポート×1) ボード上に実装されている。リアルタイムスケジューラはこの $\mu$ -PULSER に変更を加えることで実装している。

## 4 評価

一般的にリアルタイムスケジューラの性能評価で重要視される項目として、

- ハードデッドラインのタスクがそのデッドラインを常に満足すること
- ハードデッドラインタスクで高い CPU 利用率を得ること
- ソフトデッドラインタスクの応答時間が速いこと
- 一時的な過負荷時のスケジューリングの安定性を確保すること

が挙げられる。

ここでは、各ポリシー毎に同一の条件を課し、その元で種々の比較を行なう予定である。評価方法としては、

1. ある周期的なタスクセットの元で、スケジューラのオーバーヘッドを測定
2. ある周期的なタスクセットの元で、ダイレクトインタラプトの応答時間を測定
3. 適当な分散、平均の実行時間、デッドラインを持つタスクを実行させ、デッドラインを満足できる割合を測定

等が考えられるが、これらは出来るだけ実際にシステムで実行されるタスクを考慮したタスクセットを用いて評価をとる必要があるであろう。

Rate Monotonic 方式が非周期的タスクのデッドラインの保証に関して制限があるため周期的タスクセットに関しては Rate Monotonic 方式が、非周期的タスクの多いタスクセットに関しては最短デッドライン方式の方が適当であろうと予測される。

Sporadic Server 方式でハードデッドラインの非周期的タスクが非常に高い頻度で発生した場合、タスクのデッドラインを満足できなくなる。そのような場合は最短デッドライン優先方式の方が有利であると思われる。

また、Sporadic Server 方式で、イベントスレッドをリアルタイムスレッドではないが非周期的サーバのタスクとして実行させることにより、応答時間が向上すると思われる。

## 5 まとめ

本稿では、リアルタイムスケジューラの $\mu$ -PULSER 上への実装とその評価方法について述べた。スケジューラの実装では、方針/機構を分離して実装することによりスケジューリングアルゴリズムの変更を容易に出来るようになった。

スケジューリングに関する評価は、実際にシステムで実行されるタスクの性質を考慮して行なわなければならない。従って、実際にシステム上で動かされるアプリケーションの性質を調べ、それを元にしてスケジューリングアルゴリズムの評価を行う必要があるであろう。

今後、複数のプロセッサへの対応や、スケジューリングポリシーの動的変更等の研究をしていきたいと考えている。

## 参考文献

- [1] Brinkley Sprunt, Lui Sha, John Lehoczky: Aperiodic Task Scheduling for Hard-Real-Time Systems, *The Journal of Real-Time Systems*, Vol. 1, No. 1, pp 27-60, 1989.
- [2] Hideyuki Tokuda, Clifford W. Mercer: ARTS: A Distributed Real-Time Kernel, *ACM Operating Systems Review*, Vol. 23, No. 3, pp 29-53, 1989.
- [3] Karsten Schwan, Hongyu Zhou: Dynamic Scheduling of Hard Real-Time Tasks and Real-Time Threads, *IEEE Trans. on SOFTWARE ENGINEERING*, Vol. 18, No.8, pp736-748, 1992
- [4] 菅原 智義, 飯田 浩二, 秋庭 朋宏, 紺田 和宣, 矢向 高弘, 安西 祐一郎: パーソナルロボットのための敏感で柔軟な OS $\mu$ -PULSER の設計と実装, 日本ソフトウェア科学会第 9 回大会, pp 253-256, 1992
- [5] 山崎 信行, 安西 祐一郎: パーソナルロボットのためのアーキテクチャの提案, ロボティクス・メカトロニクス講演会'92 講演論文集, vol. A, pp 51-56, 1992.