

# 1 G-1 自律移動ロボット用 OS PULSER の設計と実装\*

菅原 智義      矢向 高弘      安西 祐一郎†  
 慶應義塾大学 †

## 1 はじめに

自律移動ロボットは障害物回避などを行いつつ目的地へと移動する。自律移動ロボットの制御には、多数のセンサからの入力に敏速に反応できることと、駆動系をリアクティブに制御できることが必要である。既存のロボット用 OS には駆動系の制御のためにリアルタイム性を重視したものは多いが、センサからの入力に対するリアクティブな駆動系の制御を重視したものは少なかった。そこで、我々は自律移動ロボットのリアクティブな制御を目的とした自律移動ロボット用 OS PULSER を開発した。PULSER では、direct-interrupt 機構と interrupt-driven スケジューラにより、自律移動ロボットのリアクティブな制御が可能である。本稿では、PULSER の設計と実装について述べる。

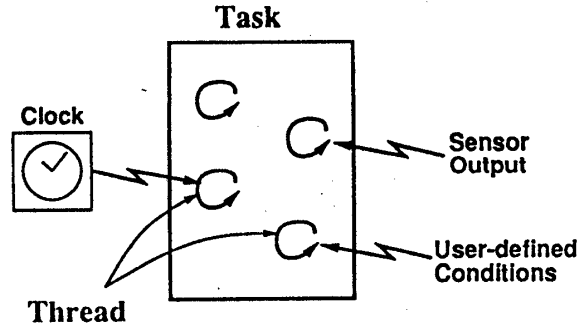


図 1: direct-interrupt 機構の概念図

## 2 Interrupt-driven スケジューラ

### リアクティブシステムの必要性

自律移動ロボットは多数のセンサと駆動系を持つ。自律移動ロボットの制御を行うには多数のセンサからの入力に対し敏速に反応でき、それを駆動系の制御に速やかに反映できることが必要である。

既存のロボット用 OS[1] ではリアルタイムスケジューラが良く用いられている。これはリアルタイムスケジューリングが、厳しいリアルタイム性を要求されるサーボモータなどのデバイスの制御に有効だからである。自律移動ロボットの制御を考えた場合、駆動系の制御にはリアルタイムスケジューリングが適しているが、ロボット全体の制御には、リアルタイム性よりも、センサからの入力を駆動系の制御に速やかに反映させること、すなわち、リアクティブ性を実現することが必要である。

### Interrupt-driven スケジューリング

我々は、リアクティブなシステムを構築するためのスケジューリングポリシーとして interrupt-driven スケジューリングを提案する。interrupt-driven スケジューリングは、ロボットを構成する多数のセンサからの入力やリアルタイムクロックなどを統合した direct-interrupt 信号により敏速にスケジューリングを行う(図1)。direct-interrupt はユーザプログラムからも発生させることができ、プログラム間にもハードウェア割込みと同等の敏速な同期機構を提供している。さらに、ハードウェア割込みと異なり、プログラムに入れ子構造を強いることがないため、リアクティブ制御のプログラムを柔軟に記述することができる。

### Interrupt-driven スケジューラの設計

ロボットプログラムは、各デバイス毎に独立したプログラムモジュールで構成されることが多い。また、ロボットがマルチプロセッサで構成されることもしばしばある。我々は、マルチモジュール・マルチプロセッサに対応するために、PULSER をマルチスレッド・マルチタスクモデル[2]に基づいて設計した。PULSER がサポートするスレッドは次の3つのタイプがある。

**継続スレッド** 他の2つのタイプのスレッドが実行可能状態にない時だけ動作できるスレッドである。通常はこのスレッドのみが動作している。

**割込みスレッド** direct-interrupt が発生した時に、起動されるスレッドである。

**周期スレッド** リアルタイムクロックが発生させる direct-interrupt により周期的に起動されるスレッドである。割込みスレッドの一種である。

これらのタイプは実行中に動的に変更することが可能である。

これら3タイプのスレッドをスケジューリングするのが、direct-interrupt 機構により駆動される interrupt-driven スケジューラである。interrupt-driven スケジューラのスケジューリングポリシーを以下に示す。ただし、スケジューリングの点では周期スレッドと割込みスレッドは同じと見なされるため、ここでは周期スレッドは割込みスレッドに含めるものとする。

direct-interrupt が発生していない時(normal フェーズ)は、継続スレッドのみをラウンドロビンスケジューリングする。センサの入力などにより direct-interrupt が発生すると、interrupt-driven スケジューラは割込みスレッドを起動し、direct-interrupt フェーズに入る。direct-interrupt フェーズでは継続スレッドを実行の対象にせず、割込みス

\*PULSER: An Operating System for Autonomous Mobile Robots

†Tomoyoshi SUGAWARA, Takahiro YAKOH, and Yuichiro ANZAI

†Keio University

レッドのみを実行する。複数の割込みスレッドが実行可能な場合、それらの間でラウンドロビンスケジューリングを行う。割込みスレッドは実行を終了すると休止状態になる。全ての割込みスレッドが休止状態になると、direct-interrupt フェーズは終了して、normal フェーズに戻り、再び継続スレッドを実行する。

実際には、direct-interrupt と interrupt-driven スケジューラは以下のように動作する (図2)。ただし、図2中の intr.1, intr.2 は割込みスレッドを、cont.1, cont.2 は継続スレッドを表す。

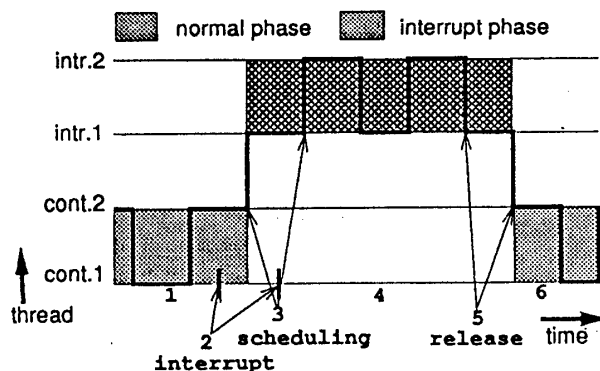


図2: スケジューリングのタイミングチャート

1. 通常、継続スレッドのみが動作している。ハードウェアタイマ割込みにより 10ms のタイムスライスでラウンドロビンスケジューリングが行われる。
2. direct-interrupt は、センサからの入力あるいはユーザプログラムからのシステムコールなどにより発生する。
3. interrupt-driven スケジューラは発生した direct-interrupt に対応するスレッドを実行可能状態にする。
4. direct-interrupt フェーズでは割込みスレッド間でラウンドロビンスケジューリングが行われ、継続スレッドはスケジューリング対象にならない。
5. 一度スケジューリング対象になった割込みスレッドは明示的に休止するか、タイムアウトになるまでスケジューリング対象になる。
6. 全ての割込みスレッドが休止状態になると、再び継続スレッドがスケジューリング対象になる。

### 2.1 監視スレッド

通常のロボット用 OS では、ユーザがプログラムを書く場合、センサの値の変化を常に監視するスレッドを多数生成する必要がある。PULSER では direct-interrupt を効率的に利用するために、状態の変化を監視する専用のスレッドを用意している。これは、ユーザの指定した手続きを一括して処理を行うスレッドである。これにより、状態の監視を実行するスレッドをつくることによるメモリや CPU パワーのオーバーヘッドを減少させることができる。ユーザは、監視する状態をチェックする関数と、その関数の返り

値に従い発生させる direct-interrupt 番号を登録することにより監視スレッドを使うことができる。

## 3 実装

我々の研究室では、自律移動ロボットを開発している。その仕様を表 1 に示す。我々は PULSER をそのロボット上に実装した。

PULSER は、センサと駆動系の制御機能以外に、ロボット間およびロボット-ワークステーション間の通信機能を提供している。この通信機能では直接、電波の届かない場所にいるロボットとも通信ができるように、動的に中継局となるロボットを探し出すことが可能である [3]。

制御部:	CPU 東芝 TMP680301 × 1 (モトローラ MC68000 互換、含む シリアルインタフェース × 3、パラレルインタフェース × 1、タイマ × 3) ROM 128KB、RAM 1MB
駆動部:	パルスモータ × 2
センサ部:	超音波センサ × 4
通信部:	無線通信機 × 1、TNC モデム × 1
その他:	電磁石ハンド × 1

表 1: 自律移動ロボットの仕様

## 4 まとめ

我々が、開発した自律移動ロボット用 OS PULSER では direct-interrupt と interrupt-driven スケジューラにより、自律移動ロボットの制御に必要なリアクティブ性を実現することができた。

PULSER の提供するこれらの機能は、物理世界の動作や状態の変化に敏感なシステム [4] を構築するためにも有効であると考えられる。

## 参考文献

- [1] David B. Stewart, Donald E. Schmitz and Pradeep K. Khosla, "Implementing Real-Time Robotic Systems Using CHIMERA II", Proceedings of 1990 IEEE International Conference on Robotics and Automation, pp.598-603, 1990.
- [2] A. Tevanian, R. Rasid, D. Golub, D. Black, E. Cooper and M. Young, "Mach threads and the UNIX kernel: The battle for control", Proceedings of the Summer USENIX Conference, 185-197, 1987.
- [3] 秋庭朋宏, 矢向高弘, 安西祐一郎, "無線により通信する自律移動ロボットの動的な中継局選択法", 情報処理学会 第 44 回 (平成 4 年前期) 全国大会 講演論文集, 1992.
- [4] 矢向高弘, 菅原智義, 安西祐一郎, "物理世界の状態変化に敏感なシステム PULSER-II の提案と実装", 情報処理学会 第 44 回 (平成 4 年前期) 全国大会 講演論文集, 1992.