

レートモノトニック・スケジューリング・ポリシーに基づく

4H-1

倒立振子の制御について

平井 幸広ⁱ中道 耕一ⁱ今屋 敬之郎ⁱⁱ坪田 正志ⁱⁱⁱ桑原 孝ⁱⁱ奥田 郁太郎ⁱⁱⁱ

1 はじめに

近年のマイクロプロセッサの性能向上はめざましく、組み込み用途向けの分野でも高速なRISCが使用できるようになった。同時に、より複雑で複合した処理をする必要性が生じ、リアクティブ性のみではなく、リアルタイム性を保証するアルゴリズムが求められている。

本報告では、組み込み用途向けRISCプロセッサを用い、レートモノトニック^{iv}に基づいたスケジューリングにより、現代制御理論に基づく倒立振子の制御を行った事例について述べる。

2 倒立振子システム構成

本倒立振子システムの構成を図1に示す。

振り軸に取り付けられたロータリーエンコーダにより角度を検出し、リニアモータ・テーブルのエンコーダにより振子の位置が検出され、CPUが読み出す。また、CPUからの出力はD/Aコンバータにより電圧に変換された後、ドライブ回路により、リニアモータへのトルクとして出力される。本システムはこの倒立振子を2組有する。

シリアル回線により、端末に接続され、端末に対して実行時間のデータを出力することができる。また、10cm間隔で配置されたLEDを表示させることができ、倒立すべき目標位置を表示することができる。

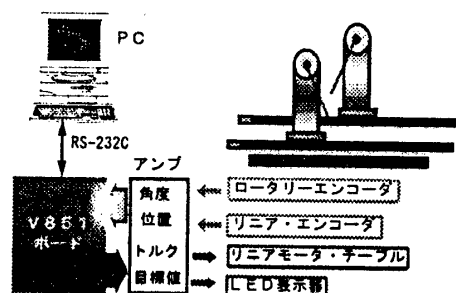


図1：倒立振子構成図

3 倒立振子の制御

本倒立振子は倒立位置を目標値とし、現代制御に基づく、デジタルサーボ制御を行っている。制御系のブロック図を図2に示す。本倒立振子の目的は、振子を倒立させるように振動させること、および、目標値となる位置で倒立した状態を保つことである。

本倒立振子では10ミリ秒毎にセンサのサンプリング、制御量の演算、アクチュエータへの出力を行う必要があり、デッドラインを有するリアルタイム処理となる。

制御系の状態方程式を線形化するため、以下の3つの系について考慮して設計し、3つのフィードバック・ゲインを倒立振子の状態によって動的にソフトウェアで切り替える。

- 倒立状態を制御するために、振子の角度が0度（真上）で線形近似した系
- 台車を左右に動かすことにより、振子を振り上げるため、振子を無視した台車だけの系
- 振子を真下に速やかに静止させるため、180度（真下）で線形近似した系

倒立振子のサーボ系
□最適デジタルサーボ系の構成

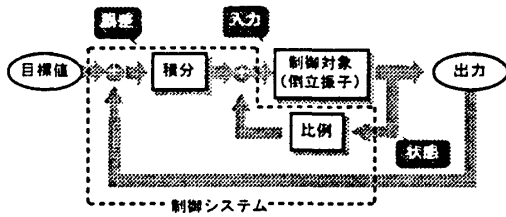


図 2：倒立振子制御系のブロック図

4 ソフトウェアの構成

本システムではデジタルサーボ制御を含め5つの処理を有し、それぞれの処理に対し、レートモニタ理論に基づいた実行制御を行っている。

- シリアル送信処理 (2 m s 間隔)
- デジタル・サーボ処理 (10 m s 間隔)
- LED表示処理 (33 m s 間隔)
- 倒立振子状態出力処理 (250 m s)
- 実行状態データ出力処理 (1000 m s)

5 周期処理について

本システムでは、周期処理を実現するために、CPUに対して1 m s 間隔での周期割り込みを発生させ、割り込み処理内で次に実行すべきタスクを選択する機構を有している。

本周期処理は、タスク5つに対し、実行周期の短い順に0から4の固定の優先度を持たせ、ており、タスクの数と優先度、実行周期は固定である。また、同期通信や排他制御の機構は持たないので、資源について衝突がある場合は、タスク・プログラムの責任において解決しなければならない。

上記の条件のもとでは、周期処理の切り替えは、プリエンパブルであるが、優先度の高いタスクが待ち状態となることはないので、コンテキストの保存エリアをスタック上に持つことが可能となり、コンテキスト切り替えのオーバーヘッドを小さくすることができる。

コンテキスト切り替え時間の最大値はタスクの数を固定とすることにより、予測可能であり、各タスクの最大実行時間も最大実行時間を調べることで予測可能である。

以上の方法によりレートモニタ理論を適用し、システム全体のCPU使用率が最大使用率を越えないことを確認した。

6 実行結果

倒立振子実行モニタの出力画面を図3に示す。この出力は50 μ s 単位の表示である。それぞれの処理がデッドラインを守りながら実行しているおり、CPUとしては性能にゆとりがあり、さらに処理を追加することが可能であることがわかる。

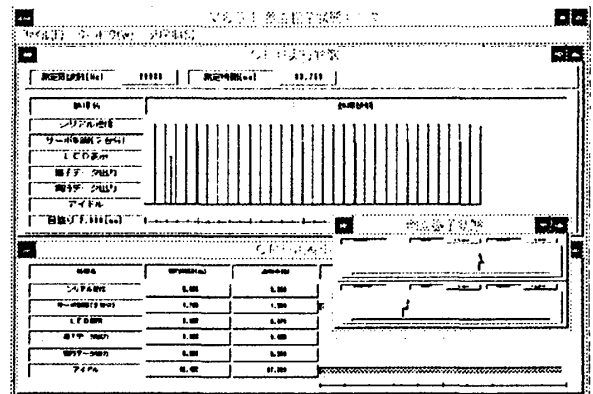


図 3：倒立振子状態モニタの出力画面

7 おわりに

今後、割り込みを考慮した周期処理の方法について実機での評価をしたいと思う。

¹ 日本電気マイコンテクノロジー (株)

² 日本電気アイシーマイコンシステム (株)

³ 日本電気 (株)

⁴ C.L.Liu and James W.Layland, Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment