

リアルタイム制御とデータ駆動アーキテクチャの相性評価

Effectiveness of Data-Driven Processor Embedded in Real-Time Systems

川村 亨久 通堂 真† 服部 健輔† 酒居 敬一†
Takahisa Kawamura Makoto Tsudo Kensuke Hattori Keiichi Sakai

1 序論

本論ではリアルタイム処理が必要なシステムとデータ駆動型プロセッサとの相性を評価する。評価にあたっては、データ駆動型プロセッサが何らかのデバイスを制御する場合をモデル化し検証した。今回の検証ではデータ駆動型プロセッサはリアルタイムシステムへの応用への可能性を大いに示す事ができた。

ロボットやヘリコプターの姿勢制御などに使われる組み込みシステムでは、1つ以上のプロセッサが複数のモーター等のデバイスを制御する。このような状況では各制御の終了時間の遅延は許されない。また、組み込みシステムはデスクトップパソコンなどと比べ、プロセッサの処理能力と個数の確保に関して制限が厳しい。ロボットの姿勢制御を例にとっても、制御の遅延が一定の範囲を越えてしまうと姿勢は乱れ、正常な姿勢を保つ事が困難になる。結果、人間と同じ様に転んでしまうのである。そこで、リアルタイムシステムが必要となる。すなわち、リアルタイムシステムは組み込みシステムでよく用いられる。

組み込みシステムでは使用できる電力や、ハードウェア(基盤や、チップ)などを配置するスペースの制限があるため、強力なプロセッサを使用できない場合がある。特に、自動車のブレーキ制御などの様に、バッテリーなどからの限られた電力によってデバイスの制御を行う場合、最近のデスクトップ向けパソコンに見られるような消費電力が高く、高性能なプロセッサの搭載は現実的でない。そこで、比較的省スペースで省電力なハードウェアとソフトウェアの両面から作りこむ事によって十分なパフォーマンスを引き出す必要がある。

ソフトウェアとは、システムのプロセッサの処理内容や、処理を行う際に参照するデータなどを記したビット配列である。システムのほとんどをハードウェアで構成する事もできるが、時期や要求、市場の動向などによって仕様変更が発生した時に柔軟に対応できない

ため、ソフトウェアによって柔軟に対応できる余地を残した方が得策である。場合によっては在庫となった製品のソフトウェアを書き換えて他の製品へと転化させる事も可能である。

現在のデスクトップパソコンのプロセッサのアーキテクチャのほとんどはフォン・ノイマン型のアーキテクチャを有している。しかし、このアーキテクチャではタスクスケジューリングやリアルタイム処理の実現に際して大きな壁がある。ノイマン型では割込みを用いて処理対象のタスクの切替を行うが、割込みは時間単位で行われる上に、タスクのスケジューリング作業に多少の時間を消費するため、スケジューリングの粒度が粗い。また、スケジューリングの粒度が粗いとリアルタイム処理におけるタスク終了時間のデッドラインの厳守が困難になってしまう。これらの要因は組み込みシステム開発をより複雑で、コストの高いものにしてしまう。

一方、本論で取り上げるデータ駆動型プロセッサは各ノードが非同期で動作するため、処理性能に対する電力効率が良く、命令単位でのスケジューリングを行える[1]。したがってノイマン型よりもタスクスケジューリングの粒度が細かいため、リアルタイム処理にも有利になり、ノイマン型よりもシステム開発を容易にするのではないかと考える。

そこで、組み込みシステム開発の複雑さと開発コストの低減を目指すためにデータ駆動型プロセッサと組み込みシステムの相性を検証する。

2 データ駆動型プロセッサによる制御モデル実現可能性の検証

我々はまず、データ駆動型プロセッサで1つのモータを制御した。制御モデルは図1とした。

機材は、データ駆動型プロセッサの評価用ボードである USB-DDMP を用いた。これには、マイコン

†高知工科大学 情報システム工学科

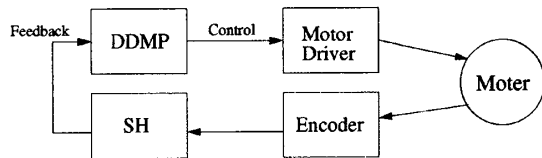


図 1: データ駆動型プロセッサによる 1 モータ制御モデル

SH7145F も附属している。このボード上に DC モータとモータのドライバとしてブリッジドライバ回路を実装し、これを DDMP の出力とつないだ。モータにはその回転位置を計測するエンコーダがつながれており、その出力はマイコンへ送られる。マイコンはエンコーダからの観測値と次のモータの回転位置の目標値を定め、DDMP 用のパケットを生成し、送信する。DDMP はそれを受け取ると内部で PID に基づく制御量の計算 [2] を行い、その結果に応じてドライバに制御信号を送信する。

今回の実験では我々はデータ駆動型プロセッサによってモータを制御する事ができた。

3 制御波形

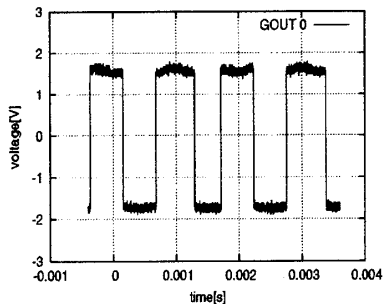


図 2: データ駆動型プロセッサの出力波形

実機によるモータ 1 個の制御は成功した。また、図 2 に見られるようにその出力波形ははっきりとしたパルスを描いており、歪みはほとんど見られない。

4 制御対象の増加に伴う性能低下の度合

今回構築した制御モデルにおいては制御対象が増加してもプロセッサの応答速度に大した変化を及ぼさない。シミュレーション結果を図 3 に示す。このシミュレーションはモータ制御用プログラムに対して制御す

べきモータの数を徐々に増やしていった様子を示す。制御すべきモータが増加しても、応答時間はほとんど変化していないことがわかる。したがって、比較的小数のプロセッサで多数のデバイスを安定して制御できる。

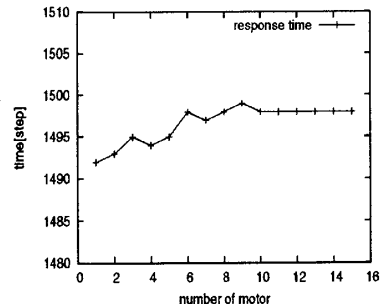


図 3: プロセッサの応答速度の推移

5 タスクスケジューリング

データ駆動型プロセッサでは投入されたパケットを順に処理するので、制御の順番などを考慮するという概念がない。したがって、開発の段階においてもタスクスケジューリングの方法や機構に対してコストをかける必要がない。

6 総括

今回使用したデータ駆動型プロセッサはシグナルプロセッサやベトリネットをベースにして各ノードが非同期に動作する。本論で示した様に波形は安定し、複数のデバイス制御にも期待できる。また、マルチタスクに対しても高い親和性があることからリアルタイムシステム開発時においても高い開発効率が期待できる。

謝辞

本研究において利用させて頂いたデータ駆動型プロセッサの評価ボード並びに各種ツールの開発に尽力された方々に感謝の念を申し上げる。

参考文献

- [1] 弓場敏嗣, 山口喜教, 社団法人 電子情報通信学会. コンピュータアーキテクチャシリーズ データ駆動型並列計算機, 株式会社 オーム社, 1993.
- [2] 山本重彦, 加藤尚武. PID 制御の基礎と応用, 株式会社 朝倉書店, 1999.