3A-01

モデル解析によるマルチレートSimulinkモデル並列化

池田 良裕† 市橋 友樹† 仲田 壮佑† 枝廣 正人†
名古屋大学大学院情報科学研究科†

1. はじめに

車載分野を始めとする組込みシステムでは大規模・複雑化が進んでおり，モデルベース開発を用いることが多くなっている．制御システム設計でよく用いられるモデルベース開発支援ツールにMATLAB/Simulink[1] があり，シミュレーション・自動コード生成機能を用いることでシステム設計をスムーズに進めることが可能である．

一方で，プロセッサの周波数の上昇が止まりつつあり，消費電力の面からも組込みシステム分野においてマルチコア CPU の導入が進んでいる．マルチコアCPUを利用するにあたりソフトウェアを並列化する必要があるが，容易ではない．

本研究では MATLAB/Simulink によるモデルベース開発に対して，Simulink モデル解析によるソフトウェア並列化を提案する．複数周期を持つSimulink モデルを対象とし，長周期タスクに対して負荷分割を行い適切なオフセットを設定することで短周期と同じ周期で動作させることを可能にする．

2. モデルベース開発へのソフトウェア並列化の適用における課題

MATLAB/Simulink を用いたモデルベース開発では，Simulink モデルと呼ばれるブロック線図によりシステムを記述する．Simulink モデルとして記述されたシステムは自動コード生成機能を通してC言語などの形で出力され，実際に動作させることができる．現在，我々枝廣研究室ではSimulink モデルからシステムの構造情報を抽出し，各ブロックの処理量を見積もり・自動で並列化コードを生成する研究[2]が進めている．

しかし，本研究で対象としている複数の周期(マルチレート)を持つ Simulink モデルにはブロック毎の処理量から負荷分散を行い，並列化を行うだけでは不十分な場合が存在する．それはSimulink モデル内に長周期の処理結果を短周期で受け取る箇所が存在する場合である．

図 1 の実行パターンは短周期タスクをコア 0，長周期タスクをコア 1 で動作させた場合である．1 回目の長周期タスクの実行が 2 回目の短周期タスクの実行を遅らせていることが確認できる．

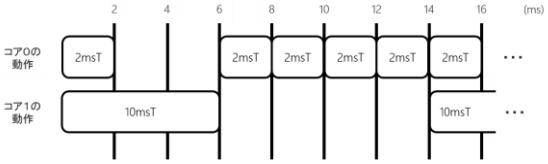

図 1 現状のマルチレートモデルの実行パターン

そこで本研究では，マルチレート Simulink モデルを対象としたソフトウェア並列化手法を提案する．

3. 提案手法

Simulink モデル内の長周期タスクと短周期タスクを別々のコアで動作させることとし，長周期タスクに対するソフトウェア並列化手法について述べる．また本稿では Simulink モデルの例として以下の図 2 のモデルを使用する．0.002s と 0.01s の 2 周期を持ち，0.002s のタスクをコア 0，0.01s のタスクをコア 1 で動作させる．

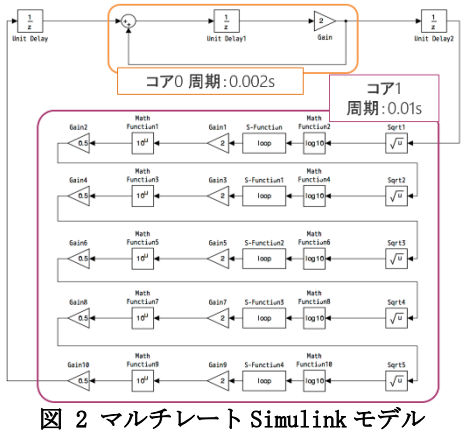

図 2 マルチレート Simulink モデル

3.1. オフセット設定によるソフトウェア並列化

Simulink モデルでは，全体の実行周期とは別に各ブロックに対し全体の実行時間の整数倍の実行周期を設定することでマルチレートモデルを作成することが可能である．

Multirate simulink model parallelization by model analysis
Yoshihiro Ikeda† Tomoki Ichihashi† Sosuke Nakada† Masato Edahiro†
† Graduate School of Information Science, Nagoya University

加えて，長周期を設定したブロックにオフセット値を設定することで起動するタイミングを任意に遅らせることが出来る．図３はマルチレートモデルでオフセットを設定した場合の実行パターンである．図中の 10msT の 2 はオフセット値として 2ms 与えられており，2ms，12ms，22ms，…で動作する．

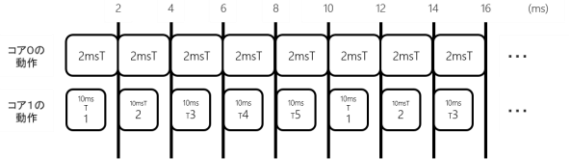

図 3　オフセット値設定後のマルチレートモデルの実行パターン

長周期タスクをレート比(長周期/短周期)の数に分割し，オフセット値をそれぞれ設定していくことで短周期に合わせて一部ずつ実行していくことが可能となる．

3.2. Simulink モデルにおける負荷分散

前節でオフセット設定により長周期タスクを短周期に合わせて実行可能にする方法について提案したが，本節では同じオフセット値を設定するブロック群の決定方法について述べる．

Simulink モデル内の長周期タスクを分割する際に各ブロックの処理量を見積もる必要がある．本研究では RH850 ボードの性能見積もりをハードウェア抽象化記述である SHIM で記述したものを使用し，各ブロックの処理量とした．

次に，Simulink モデルからブロック間の接続情報，ブロックに対応するコード情報，ブロックごとの処理量などを抽出・追加した BLXML を生成した．本研究では BLXML からより探索が容易な形である BLGraph を生成し使用した．

長周期タスク内の始点となるブロックから BLGraph 上で探索を始め，終点のブロックまで順次ブロックの処理量を加算していく．合計処理量を算出後，レート比で除算した結果が長周期タスクを分割した一つ当たりの処理量の目安となる．再び，長周期タスクの始点ブロックから探索を始め，処理量を加算していき，目安の処理量に達するまでのブロック群に同じオフセット値を設定する．再び探索を続行し，目安の処理量の 2 倍の値までブロックの処理量を加算する．値に達したら一度目と同じよう該当ブロック群にオフセット値を設定する．これを終点のブロックまで繰り返す．

設定するオフセット値は(短周期)×(分割位置 - 1)である．例えば長周期 10ms，短周期 2ms の場合オフセット値は始点側から 0.0ms，2.0ms，4.0ms，6.0ms，8.0ms となる．

4. 性能評価

負荷分散を行い，オフセット値を設定した Simulink モデルから生成される並列化コードと，オフセット値が設定されていない Simulink モデルから生成される並列化コードの比較・評価を行った．

図２のマルチレート Simulink モデルを対象とし，長周期側の実行時間を約 0.1sec，短周期側の実行時間をほぼ 0sec に調整した．今回の評価では短周期タスクの最後で実行時間を計測した．

表１はオフセット値設定前後の実行結果である．オフセット値設定前は長周期タスクの実行がある場合のみ実行時間が長くなっていることが確認できる．対して，オフセット値設定後は各サイクルでほぼ同じ実行時間である．また，この評価から長周期側タスクの分割が適当であることも確認できる．

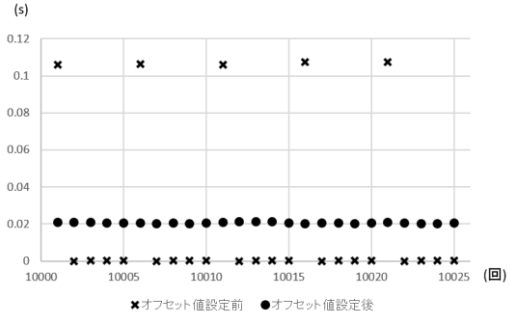

図 4　オフセット値設定前後の比較結果

5. おわりに

本論文では，複数周期を持つ Simulink モデルを対象としたソフトウェア並列化手法を提案した．処理量を目安に長周期タスクを分割し，オフセット値を設定することで短周期タスクの実行を一定の周期で実行できることを確認した．

今後は，長周期タスクの一部を短周期タスク側のコアで動作させるような異なる周期のタスクを一つのコアで動作させることを考慮した負荷分散方法について検討したい．

参考文献

[1] MathWorks: MATLAB/Simulink, http://www.mathworks.co.jp

[2] 山口，池田，枝廣他: Simulink モデルからのブロックレベル並列化，組込みシステムシンポジウム 2015(ESS2015),2015,pp.123-124.

[3] M.Gondo, F.Arakawa, and M.Edahiro: Establishing a Standard Interface between Multi-Manycore and Software Tools - SHIM, COOL Chips XVII, VI-1, 2014.