モデルベース開発におけるクロスレイヤ設計手法のマルチコア上モータ制御実装への適用

竹松 慎弥[1] 鍾 兆前[1] 井上 雅理[2] 横山 静香[2] 小島 流石[3] 近藤 真己[5]
中本 幸一[4] 安積 卓也[3] 道木 慎二[2] 本田 晋也[1] 枝廣 正人[1, a)]

アブストラクト 組込み制御分野での設計生産性向上，高性能化要求に応えるべく，Simulink に代表されるモデルベース開発と，マルチコア向け実装設計を協調させるクロスレイヤ設計手法の研究開発を進めている．本発表では，提案するクロスレイヤ設計手法をモータ制御に適用した事例について報告する．

Applying Cross-Layer Design in Model-Based Development to Motor Control Systems on Multicore Processors

Shinya Takematsu[1] Zhaoqian Zhong[1] Masamichi Inoue[2] Shizuka Yokoyama[2] Sasuga Kojima[3]
Masaki Kondo[5] Yukikazu Nakamoto[4] Takuya Azumi[3] Shinji Doki[2] Shinya Honda[1] Masato Edahiro[1,a)]

Abstract A cross-layer design methodology is presented for motor control systems. In our cross layer design, to increase productivity and achieve higher-performance with lower power consumption for embedded control systems, a co-design technique is utilized between model-based development and parallel implementation on multicore processors.

1. はじめに

車載等の高性能組込み制御設計においては，大規模化に対応した設計生産性向上を目的としてモデルベース開発が，高速低消費電力を目的としてマルチコアプロセッサが，それぞれ使われ始めている．ところが，現状のモデルベース開発ではマルチコアを意識せず設計を進めるため，実装後に期待する性能とかい離し，手戻りコストが大きくなる場合が多い．

この課題を解決するため，著者らは，Simulink[1]に代表されるモデルベース開発と，並列実装設計を協調させるクロスレイヤ設計手法の研究開発を進めている．本発表では，提案するクロスレイヤ設計手法をモータ制御に適用した事例について報告する．

1 名古屋大学大学院情報科学研究科．Graduate School of Information Science, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan.
2 名古屋大学大学院工学研究科. Graduate School of Engineering, Nagoya University.
3 大阪大学大学院基礎工学研究科. Graduate School of Engineering Science, Osaka University.
4 兵庫県立大学大学院応用情報科学研究科. Graduate School of Applied Informatics, University of Hyogo.
5 NEC ソリューションイノベータ株式会社. NEC Solution Innovators, Ltd.
a) eda@ertl.jp

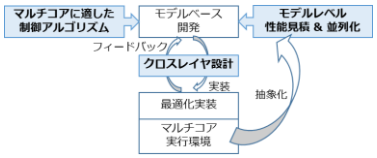

図 1. クロスレイヤ設計環境
Fig. 1. Cross-layer design methodology

2. クロスレイヤ設計手法

図 1 に提案するクロスレイヤ設計環境を示す．制御設計と実装設計を協調させるため，著者らが提案したモデルベース並列化（**図 2**）[2]を用いている．

図 2 において，中心にあるブロックレベル構造 BLXML は，Simulink ブロックの接続関係，対応する C コード，性能情報，コア割当情報等を持ち，制御設計と実装設計の協調を支える．左上が制御設計ループである．ハードウェア抽象化記述 SHIM[3]により性能を見積り，モデルレベル並列化[4]を用いてブロックのコア割当を算出，並列性能情報をフィードバックする．

コア割当情報よりマルチコア対応 RTOS 向けの並列化コードを生成し[5]，実機上で動作させ，性能情報をフィードバックする．BLXML から CSP モデルを生成し，並行実行時の読み書き順序逆転検出等を行うことも可能である[6]．

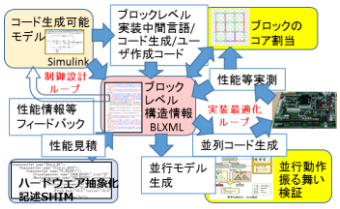

図 2. 並列化設計フロー [2]

Fig. 2. Parallelization design flow [2]

3. モータ制御への適用

提案するクロスレイヤ設計環境を基にモータ制御にベアメタル実装した．実験環境としてブラシレスモータ評価環境[7]を用い，マルチコアマイコンとしてルネサスエレクトロニクス社製 RH850/F1H を用いた（**図 3**）

図 4 に Simulink モデル，**図 5** にコア割当結果（プラント部分は除いてあり，割当コアで色分け）を示す．モデルには制御周期があり，図 5 に囲いで示したブロック群が 10ms 周期，他はすべて 100us 周期である．**図 6** にシミュレータと実機での回転速度，**図 7** に並列と逐次との実行時間比較を示す．期待通りの制御が，2 コアで約 1.8 倍の性能向上で実現できている．

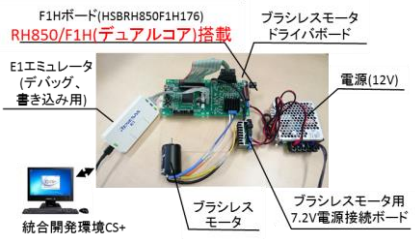

図 3. モータ制御評価環境 [7]

Fig. 3. Evaluation kit for motor control [7]

4. まとめと今後の課題

本発表では，モデルベース開発と，マルチコア向け実装設計を協調させるクロスレイヤ設計手法を，モータ制御に適用した事例を報告した．実験の結果，2 コアで 1.8 倍の高速化を達成し，有効性を示すことができた．

今後の課題として，クロスレイヤ設計手法を用いた RTOS 上への実装，マルチコアに適した制御アルゴリズムの研究開発があげられる．

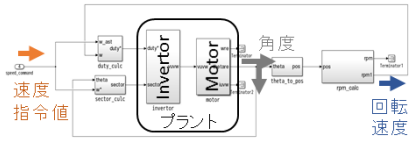

図 4. モータ制御およびプラントの Simulink モデル

Fig. 4. Simulink model for motor control and plant

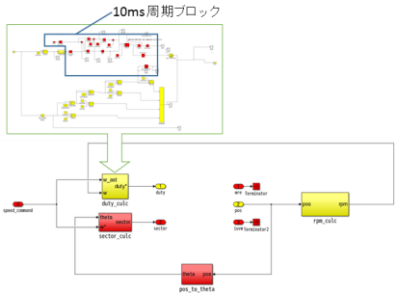

図 5. コア割当結果（色分けはコア割当）

Fig. 5. Core assignment (colored by assigned core)

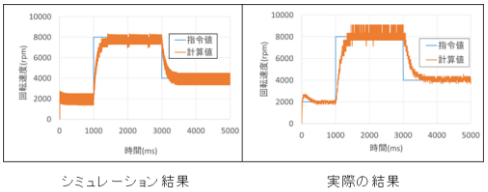

図 6. シミュレーションと実機の回転速度

Fig. 6. RPM in simulation and evaluation

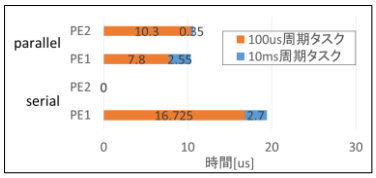

図 7. 並列実行と逐次実行の実行時間

Fig. 7. Execution time in parallel and serial execution

謝辞 本研究は科研費 16H02800 の助成を受けている．

参考文献

[1] Mathworks, https://jp.mathworks.com/products/simulink.html
[2] 山口他, Simulink モデルからのブロックレベル並列化, ESS2015 (21), 2015.
[3] Multicore Association, http://www.multicore-association.org/workgroup/shim.php
[4] 鍾他，モデルベース開発におけるマルチ・メニーコア向け自動並列化, ETNET2017 (47), 2017.
[5] 中野他，モデルベース並列化（MBP）におけるマルチレートモデルの車載 RTOS 向けランタイムとコード生成, ETNET2017 (4), 2017.
[6] 山本他，モデルベース並列化における CSP モデルを利用した形式検証の適用, ETNET2017 (6), 2017.
[7] 北斗電子, http://www.hokutodenshi.co.jp/7/HSBRH850F1L100.htm