

# 高速モータの制御ソフトウェアに対する モデルベース開発手法の適用

## Applying Model-Based Development ( MBD ) to High-Speed Motor Control Software

藤澤 翔太 技術開発本部技術基盤センター制御・ソフト開発グループ  
片山 啓 技術開発本部技術基盤センター制御・ソフト開発グループ  
五十嵐 亮 技術開発本部技術基盤センター制御・ソフト開発グループ 主査  
藤井 正和 技術開発本部技術基盤センター制御・ソフト開発グループ グループ長

組込みソフトウェア開発においてモデルベース開発と呼ばれる開発プロセスが注目されている。モデルベース開発とは文書では表現しきれないシステムの挙動をモデルにより明確化しながら開発を行う手法である。株式会社 IHI では、モデルを用いて設計および検証を行った後、モデルから自動で生成したプログラムコードによりソフトウェアを調製し、実際の永久磁石同期モータ制御システムに適用するという一連の開発プロセスを試行した。MILS ( Model In the Loop Simulation ) と実際の装置試験の結果を比較し、モデルベース開発の有効性と今後の課題について述べる。

In embedded software development, a development process called Model-Based Development ( MBD ) is drawing attention. MBD is a method of developing while clarifying the dynamic behavior that cannot be expressed by the specification document in the model. We applied MBD to the control software of the permanent magnet synchronous motor. After expressing the specification by the model and verifying the control algorithm on the computer, we created the control software with the program code automatically generated from the model and applied it to the actual motor. In this paper, we describe the effectiveness of MBD and future works through our approach.

## 1. 緒 言

IHI では、世界的な低炭素化、持続可能性の確保といった社会課題を解決するべく、航空機システムや車両過給機など多くの製品の電動化<sup>(1)~(4)</sup>に取り組んでいる。電動化を支える要素技術として制御技術の向上への期待が高まっており、制御ソフトウェアの役割は増加傾向にある。

高速回転モータ制御のように処理が複雑でリアルタイム性の高い組込みソフトウェア開発では、初期段階で完全な仕様書を作成することは困難であり、開発工程のなかで補完・詳細化されることになる。そのため、この仕様書の更新により複数の後戻りが生じ、追加の工数が掛かってしまう。また、従来の文書を用いたソフトウェア開発では、複雑な要求仕様を誤解や漏れなく伝達することは困難である。

加えて、複雑化した組込みソフトウェア開発において、装置の完成を待ってソフトウェア開発を行ったのでは開発期間が長期化してしまう。最適なコンポーネントの組合せやその調整パラメータが無数に存在するため、実際の装置

を用いた設計や調整を行うには莫大<sup>ばく</sup>な時間と費用を要する<sup>(5)</sup>。さらに、仕様書から手作業でコーディングを行う際に不完全な伝達や単純ミスによる不具合が混入する可能性もある。

このような問題を解決する手法の一つとして、近年自動車メーカーを中心に適用が進んでいるモデルベース開発 ( Model-Based Development : MBD ) がある。モデルとは、文章として表現することが困難な装置およびソフトウェアの挙動を、状態遷移図やブロック線図などを用いてグラフィカルに表現して明確化するものであり、このモデルをよりどころとして開発を進める手法が MBD<sup>(6)</sup> である。代表的な特徴は次のとおりである。

- (1) モデルによる仕様の表現
- (2) モデルのシミュレーションによる設計の詳細化、妥当性の検証
- (3) モデルからのプログラムコード自動生成によるソフトウェアの実装

これらの特徴により、従来手法の問題点を改善し、仕様の明確化、検証性の向上、コーディングのミスの低減が可

能となる。本技術を活用することで、ソフトウェアの観点から IHI の電動化製品の開発効率および製品価値の向上を目指していく。

また、株式会社 IHI エアロスペースでは、ロケット電子制御システムの電動アクチュエータの制御ソフトウェア設計において MBD を試行した実績があり、制御設計の開発効率を向上させることに成功している<sup>(7)</sup>。その際、制御対象である装置の非線形要素をモデルに取り入れることが困難であったため、MILS (Model In the Loop Simulation) 単体での検証の正確性は必ずしも制御設計や検証のために十分でなく、実際の装置をリアルタイムで実際に動作させる RCP (Rapid Control Prototyping) を活用している。なお、MILS はシステムを構成する装置とソフトウェアのいずれもモデルを用いて行うシミュレーションのことであり、RCP は制御モデルを組み込んだ汎用コントローラで実際の装置を動作させる試験のことである。

本稿では、モータ制御システムに対する IHI のモデル活用の経緯や設計・モデリング・検証の各プロセスにおける取組みについて述べる。そして、航空機用空冷ユニット向け小型高速回転電動プロアを想定したモータ制御システムに対して、MILS およびモデルからの自動生成コードを適用した装置試験を実施して、その結果を比較し、MBD の有効性を示す。また、自動生成コードを用いたソフトウェアと手作業によるソフトウェアを 1 周期当たりの実行時間およびメモリサイズの観点でその性能を評価し、そのなかで抽出された課題について述べる。

## 2. モータ制御ソフトウェアの MBD

### 2.1 制御アルゴリズム検証へのモデルの適用

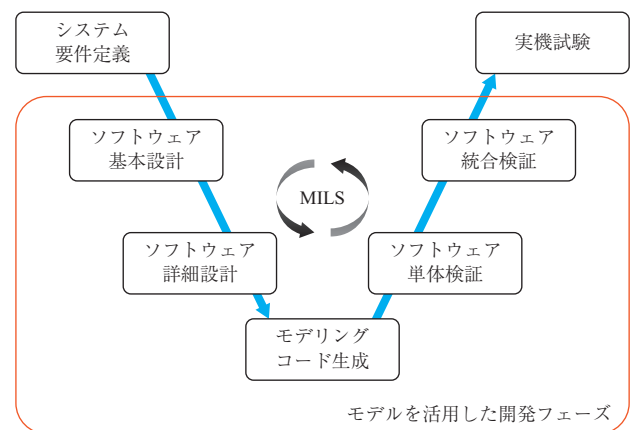
IHI では、MBD を行う以前からモータ制御ソフトウェアの仕様書ベースでの開発が行われている。そこで、モデルを活用する最初の取組みとして、既存のソフトウェアのアルゴリズムをそのままモデルに変換し制御器モデルを作成した。主に電磁気学や運動学などの物理法則を

表現する数式に基づきモデリングされたモータやインバータなどの制御対象モデルと組み合わせることで、モータ制御システムの MILS 環境を実装した。第 1 図にそのモータ制御システムを示す。モデリングには多くの分野で広く採用されている MathWorks 社 (アメリカ) の MATLAB®/Simulink® を用いた。この方法では、制御パラメータの調整や振舞いの解析が可能となった一方、複雑な条件分岐により制御コンポーネントの実行を制御する既存のソフトウェアのアーキテクチャが制御器モデルに引き継がれてしまった。このため処理や信号の流れが不明瞭になり、モデルの利点である可読性や検証性が十分に発揮されず、新機能の実装や不具合の解析を行う際の効率性に課題を残していた。

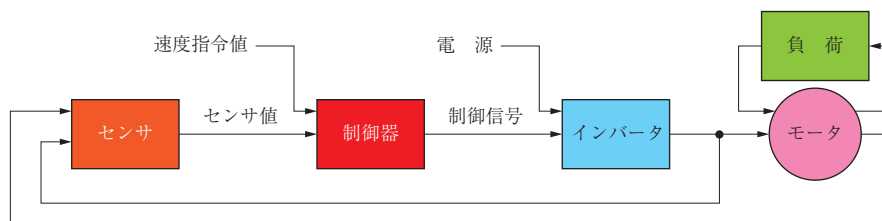
この課題に対処するため、既存のソフトウェアは参考にとどめ、V 字開発プロセスをひとつとおり実施し直すなかで、MBD の利点が得られるアーキテクチャへと改修を行った。第 2 図にソフトウェアの V 字開発プロセスを示す。以降では開発フェーズごとに MBD の適用について述べる。

### 2.2 基本設計

基本設計では、機能の分割およびアーキテクチャの構築を行う。2.1 節で述べた課題を克服するため、制御器モデ

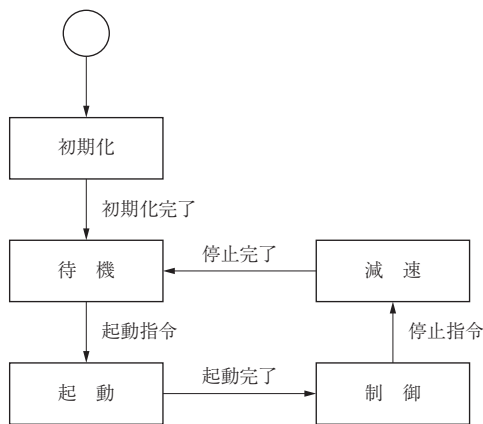


第 2 図 ソフトウェアの V 字開発プロセス  
Fig. 2 V-process of software development

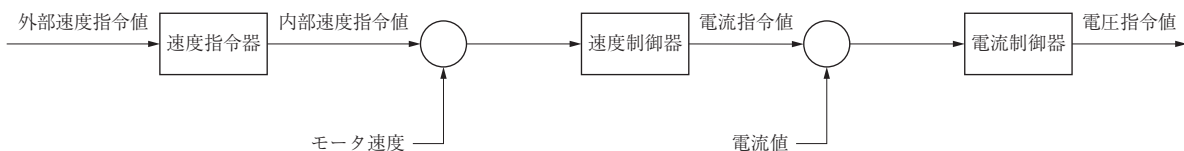


第 1 図 モータ制御システム  
Fig. 1 Motor control system

ルにおいて可読性と検証性が確保されるよう全体を構成する。具体的には、制御コンポーネントの実行を制御する複雑な条件分岐を整理し、制御アルゴリズムの最上位にある状態遷移が主要な実行をすべて管理するような構成への変更である。これにより、最上位の状態遷移と制御コンポーネントの接続関係に着目することで処理と信号の流れを把握することが可能になる。第3図に最上位の状態遷移の例を示し、第4図に制御コンポーネントの接続の例を示す。この例では、第3図に示す最上位の状態遷移が制御の状態である場合に、第4図のコンポーネントの接続で表現される処理が実行される。



第3図 最上位の状態遷移の例  
Fig. 3 Top-level state transition



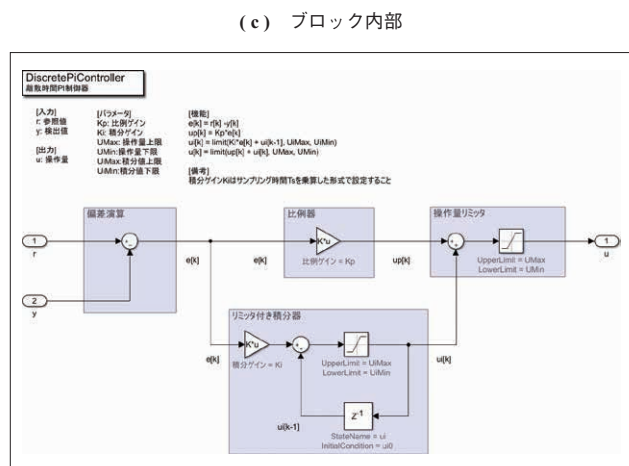
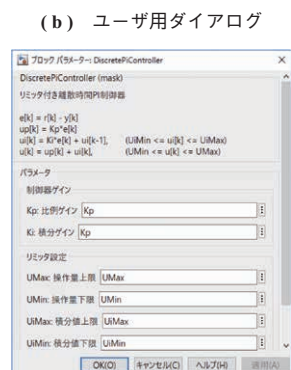
第4図 制御コンポーネントの接続の例  
Fig. 4 Connection of control components

### 2.3 詳細設計およびモデリング

詳細設計では、上述の基本設計に基づいて制御コンポーネントをモデリングにより詳細化する。制御コンポーネントの多くはモータ制御における汎用的な機能であることから、仕様を記したライブラリブロックとして整備した。ライブラリブロックは、Simulink で実装可能なモデルコンポーネントタイプの一つで、C 言語の関数のように一つの機能をモジュール化したものである。第5図にライブラリブロックの例を示す。作成の際にはユーザ用のダイアログを設定し、そのダイアログから入出力や使用方法の確認、モデルやパラメータの設定ができるよう実装した。これにより、各制御コンポーネントの仕様が明確になることに加え、可読性、他機器での再利用性を向上させることができる。

### 2.4 単体検証

単体検証とは、機能単位で妥当性を検証することである。単体検証を行うシミュレーション環境を構築した後、ライブラリブロックに対し適当なテスト入力を与え、期待される出力が得られていることの検証を行う。MATLAB/Simulink によりライブラリブロックにひも付けてシミュレーション環境を構築することで、単体検証のトレーサビリティを実現する。また、検証時にテスト入力により機能がテストされた割合を計測することでテストのカバレッジ（網羅性）を確保する。



第5図 ライブラリブロックの例  
Fig. 5 Library block

## 2.5 統合検証

制御コンポーネントであるライブラリブロックを組み合わせることで制御器モデルを構成し、モータモデルやインバータモデルと接続することでモータ制御システムのMILSが可能となる。統合検証では、このMILSを活用して、想定される状況に対するソフトウェアの振舞いの妥当性や要求仕様に含まれる制御性能について評価する。その結果、設計およびモデリングのミスによる状態遷移条件の間違いや設定値の不具合などを装置試験の前に発見し、取り除くことができた。また、MILSにおいてモータを回転させることができる程度の制御器モデルを構築した後は、設計（モデリング）、単体検証、基本的項目の確認を行う簡易的な統合検証のPDCA（Plan-Do-Check-Action）サイクルを素早く回すことで開発効率を向上させることができた。

## 2.6 実際の装置を用いたシステム検証

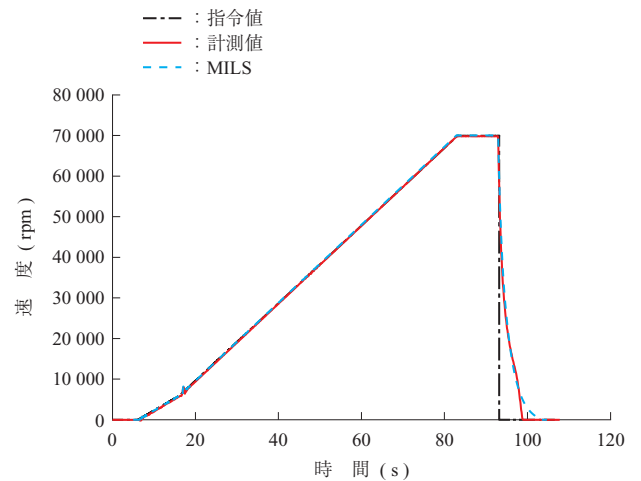
MBDによる自動生成コードを小型高速回転電動ブロー向けの永久磁石同期モータへ適用する。本モータの制御器モデルを構築し、MILSで検証した後、自動生成したコードによりモータ回転試験を行った。コードの自動生成はコーディングのミスの低減やコーディング時間の削減に非常に効果的だが、モデルを用いて開発することが困難なセンサ値取得やドライバ操作などハードウェアとのインタフェースを担うコードについては別途調製する必要がある。

試験時には、MILSにより検証済みであるソフトウェアの制御アルゴリズムや制御ゲインを修正することなく対象のシステムを動作させることができた。

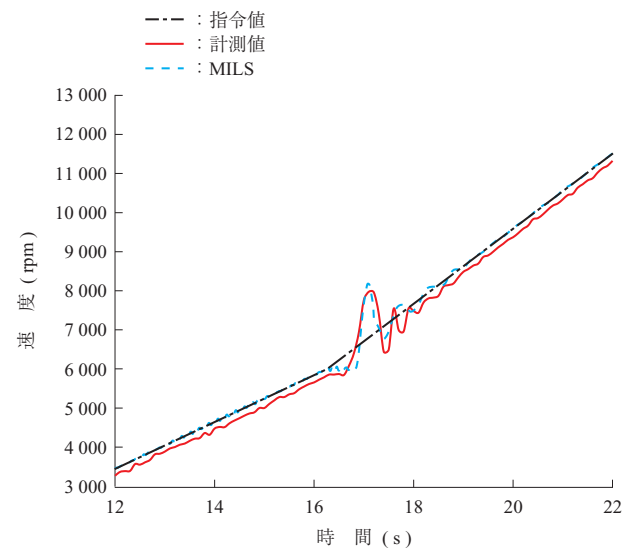
装置試験とMILSの結果を比較し、MILSの精度を評価する。停止状態から70,000 rpmまで速度指令値を上昇させた場合の速度について装置試験の計測値とMILSの結果を重ねた状態を第6図に示す。加えて、性能向上のため速度6,000 rpm程度で制御方法を切り替えることによる一時的な速度上昇が観測されたため、その箇所の拡大を第7図に示す。第6図および第7図から、装置試験の計測値とMILSの線はほぼ重なっており、MILSは制御アルゴリズムの検証やモータ制御システムの簡易的な性能解析には十分な精度であるといえる。

## 3. 自動生成コードと手作業コードの比較

自動生成コードを用いたソフトウェアとすべて手作業により開発した既存のソフトウェアについて、1周期当たり



第6図 モータ速度の比較（全体）  
Fig. 6 Comparison of motor speed (overall)



第7図 モータ速度の比較（制御切り替え時の拡大）  
Fig. 7 Comparison of motor speed (enlarged figure of control switching timing)

の最大実行時間およびメモリサイズを比較した結果を第1表にまとめる。なお、メモリサイズはソフトウェアのプログラムを格納するために必要なメモリの容量である。自動生成コードを用いたソフトウェアは、手作業のソフトウェアに比べ1周期当たりの最大実行時間が長くなっている。

第1表 最大実行時間およびメモリサイズの比較  
Table 1 Comparison of processing time and memory size

項目	手作業によるソフトウェア	自動生成コードを用いたソフトウェア
1周期当たりの実行時間	1.00	1.56
ROM使用量	1.00	0.98
RAM使用量	1.00	0.42

（注）手作業によるソフトウェアの数値を1.00とする

自動生成コードと既存のコードで制御アルゴリズムに差異があり単純な比較はできないが、自動生成コードを確認すると実行効率の意味で不利なコードとなっている箇所も散見されている。本試験においては、自動生成コードを用いたソフトウェアの実行時間は想定されていた実行周期に収まったが、制御性能の向上や製品の低コスト化のために、ソフトウェアの処理負荷を軽減させることは重要である。この課題に対処するべく、ソフトウェア全体のアーキテクチャおよび自動生成コードを効率化させるためのモデリング手法について検討する必要がある。メモリサイズについては自動生成コードの効率は手作業のコードと比べてROM 使用量で同程度、RAM 使用量で半分以下となった。このため、コードの自動生成によりハードウェアに対し要求されるメモリ容量が増大していないことが確認できた。

#### 4. 結 言

本稿では、モータ制御ソフトウェアに対する MBD 手法の適用に関する IHI の取組みをつうじて、本技術がソフトウェアの仕様の明確化、検証性の向上、コーディングのミスの低減に効果的であることを示した。また、MILS とモデルからの生成コードを適用した装置試験の結果を比較することで、MILS は制御アルゴリズムの検証やモータ制御システムの簡易的な性能解析のために十分な精度をもっていることを示した。

一方で、自動生成コードを用いたソフトウェアの実行時間は、すべて手作業で開発されたソフトウェアに比べて長くなっている。実行時間の短縮は制御周期を短く設定できるようにすることを意味し、制御性能の向上に直結する。また、処理負荷を低減することにより安価なマイクロコンピュータの選定が可能となり、低コスト化にも寄与するため非常に重要である。今後は、この課題の解決に向け、生成コードを含むソフトウェア全体のアーキテクチャや自動生成コードを効率化させるためのモデリング手法について検討を進めていく。

また、今後の展望として、複数機種での適用を見据えた制御器モデルの要求仕様の整理とアーキテクチャの構築を

行っていく。現状のモデルは、共通に必要な機能と特定の機種にのみ必要な機能が存在している。これらについて切り分け・整理することで、新しい機種に対するモデルの再利用性や開発効率の向上を可能にする。加えて、より迅速にお客さまのニーズに対応できるよう IHI の電動化製品開発への本技術の適用を推進していく。

#### 参 考 文 献

- (1) 大依 仁：次世代モビリティのための旅客機と電動化システムの将来, IHI 技報, Vol. 59, No. 2, 2019 年 6 月, pp. 34 - 41
- (2) 株式会社 IHI：持続可能なモビリティ社会の実現に挑戦する, IHI 技報, Vol. 59, No. 2, 2019 年 6 月, pp. 18 - 21
- (3) 吉永誠一郎, 青山茂一, 山口浩二, 軸丸武弘：高速回転機械の電動化を支える要素技術, IHI 技報, Vol. 59, No. 2, 2019 年 6 月, pp. 42 - 47
- (4) 株式会社 IHI：プレスリリース「世界初、航空機用 100 kW 級高出力パワーエレクトロニクスの空冷化に成功 ～クリーンな空冷技術で、パワーエレクトロニクス適用分野の拡大へ～」, [https://www.ihico.jp/ihico/all\\_news/2020/aeroengine\\_space\\_defense/1196478\\_1607.html](https://www.ihico.jp/ihico/all_news/2020/aeroengine_space_defense/1196478_1607.html), (参照 2020. 9. 1)
- (5) 脇谷 伸, 山本 透, 森重智年, 足立智彦, 原田靖裕, 村岡 正, 仁井内進：自動車エンジニアを対象としたモデルベース開発 (MBD) 基礎研修の実践と評価, 工学教育, Vol. 66, No. 1, 2018 年 1 月, pp. 60 - 66
- (6) A. Ohata and K. R. Butts : Improving Model-based Design for Automotive Control Systems Development, Proceedings of the 17th IFAC World Congress, (2008. 7), pp. 1 062 - 1 065
- (7) 下浦美那, 辻本圭史, 森田康志, 大依 仁：ロケット電子制御システムへのモデルベース開発手法の適用, IHI 技報, Vol. 54, No. 1, 2014 年 3 月, pp. 34 - 40