ANALOG DEVICES

技術資料

先進的モーター制御システムのモデルベースデザイン

この数十年間、先進的なプロセッサ機能を利用して設計を支援する方法が検討されてきました。今日では、設計の柔軟性が高くなり、設計者は MATLAB®による標準的なモデルベースデザインを利用し、Simulink®によってモーター制御システムの機能を最適化することで、設計にかかる全体的な時間を最短化できるようになりました。さらに、このような手法により、シミュレーションモデルを再利用して、最終市場アプリケーションにおけるシステムの正しい機能性と必要な性能を実現することもできるようになりました。

著者：Anders Frederiksen

モデルベースデザイン（MBD）はこの数十年間話題になってきましたが、モデル作成からインプリメンテーション終了までの全設計フローにまで進化したのは最近のことです。1970 年代にはシミュレーションにアナログ・コンピューティング・プラットフォームが使用されていましたが、制御ハードウェアのインプリメンテーションはトランジスタ・レベルで行われていました。2000 年代に至り、シミュレーション・ツールの進歩の結果、グラフィカルな制御回路図作成ツールや制御設計ツールが登場し、制御の設計や評価に関わる複雑なタスクが大幅に簡素化されました。しかし、ハードウェア制御のアルゴリズムを開発する作業において、制御システムの設計者は依然として C コードを記述してシミュレートされた設計を反映しなければなりませんでした。2010 年代初頭の現在、完全な MBD を使用してシミュレーション・プラットフォームとハードウェア・インプリメンテーション・プラットフォームの両方に共通の制御設計を適用し、複雑な制御アルゴリズムをハードウェア・プラットフォーム上に迅速に展開することができるようになりました。

History	Simulation	Implementation
197x	Analog computing	Transistor amplifier
198x	Custom simulation program - Fortran	Mixed signal components
199x	Standard simulation platform	Microcontrollers – Assembly code
200x	Matlab – Simulink control schematic	Fast DSP & RISC – C code
201x	Simulink Model → Embedded C code for embedded target	

図 1 – 能力発展の歴史

Rev. 1.1, 09/13

アナログ・デバイセズ株式会社

本　　社／〒105-6891　東京都港区海岸 1-16-1　ニューピア竹芝サウスタワービル
電話 03（5402）8200
大阪営業所／〒532-0003　大阪府大阪市淀川区宮原 3-5-36　新大阪トラストタワー
電話 06（6350）6868

MBD は、開発全般にわたって実行可能な仕様としてシステム・モデルを使用するプロセスです。このシミュレーションを中心とする方式を使用すれば、従来のハードウェアのプロトタイプを使用する方法よりも設計上の選択肢やトレードオフをよく理解することができ、事前に決めた性能の基準に合わせて設計を最適化することができます。設計者は、複雑な構造や膨大なソフトウェア・コードを使用することなく、連続時間や離散時間のビルディング・ブロックを使用して、高度な機能的特性をもったモデルを定義することができます。また、既存の C コードと標準制御ライブラリ・ブロックを組み合わせることで、設計効率を最大限に高めることができます。これらのモデルをシミュレーション・ツールとともに使用することで、迅速なプロトタイピング、ソフトウェア・テスト、そしてハードウェア・イン・ザ・ループ（HIL）シミュレーションが可能になります。シミュレーションによって、設計サイクルの後の段階を待つことなく、ただちに不整合やモデリング・エラーを発見することができます。また、自動コード生成機能により、同じアルゴリズムを手動で何度も実装することなく、ハードウェア・プラットフォーム上で実行することができます。このようにして設計プロセスが簡素化され、ハードウェア設計時のエラーが最小限に抑えられ、市場投入までの合計時間が短縮します。

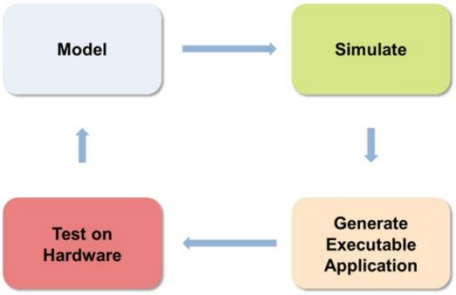


図 2 - MBD の設計フロー

MBD では、設計全体に含まれる個々のタスクを最適化するステップがいくつもあります。これらのタスクは異なる設計技術者や設計チームが行い、最終的にまとめて全体の設計と完全なシステムを作ることができます。この方式では、個々のタスクを高いレベルで抽象化し、全体的な設計フローを所定の最終アプリケーションに合わせて最適化することができます。要するに、MBD を使用すれば、設計者は従来の設計方式を拡張し、モデルの作成からシミュレーション、コード生成、HIL テストへと制御された方法で直接進み、システム挙動を段階的に変更することができるため、システム全体を設計し直す必要がありません。

さまざまな設計フェーズと、MBD フローの個々のステップのスケールを図 3 - MBD インプリメンテーションの概念に示します。これらのステップ全体で MBD の「標準」フローを構成しています。モーター制御の設計という観点では、次のように分類することができます。

- 動作コンセプト
  - モーター・システムの全体的機能
- プラント・モデリング／アーキテクチャ

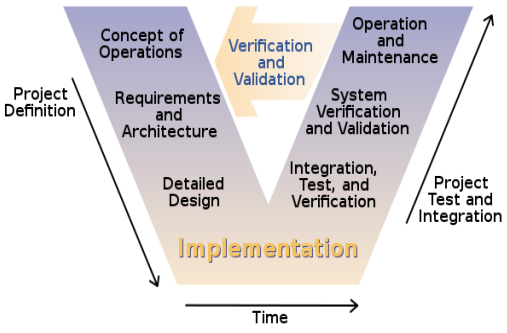


図 3 - MBD インプリメンテーションの概念

  - モーターのモデル、負荷、パワー・エレクトロニクス、シグナル・コンディショニング等の開発
- コントローラ・モデリングと要求事項
  - 3 相 PM モーターのエンコーダ・ベースのベクトル制御
- 解析と合成 – 詳細設計
  - 開発したモデルを使用してプラント・モデルの動的特性を確認
  - システムの調整と設定
- 妥当性確認とテスト
  - オフライン・シミュレーションおよび／またはリアルタイム・シミュレーション
  - 動的システムの時間応答の調査
- 組込みターゲットへの展開 ― 全動作
  - 自動コード生成
  - テストと検証
  - コントローラ・モデルのアップデート

これらすべてを組み合わせ、設計全体を調整するためのマルチステップ・アプローチが構成され、個々の制御ステップを個別に解析できるようになります。ハードウェアとソフトウェアの仕様が決まった段階で、システムの全アーキテクチャをセットアップし、システム全体の具体的なアルゴリズムと機能の両方を実際に展開します（図 4 を参照）。また、コントローラとプラント・モデルのシミュレーションを評価することもできます。さらに、ハードウェアを使用せずにオフラインでアルゴリズムの開発を設計し、全体的なシステム性能要求に合わせて精密な調整ができるほか、最初に生成されたコード（既存コードの再利用やコード生成ツールで生成したもの）を組込みコントローラに展開して、PC 上のシステム・シミュレーションとハードウェア・ターゲット上の実際のインプリメンテーション・データを比較することができます。適切なバランスがとれた MBD 構造を決定するには、モデルの複雑さを考慮しなければなりません。しかし、バランスのとれたコンセプトがあれば、設計作業のなかで個々のモデルをただちに変更して、ドライブ・システム全体から正確な結果を引き出すことができます。

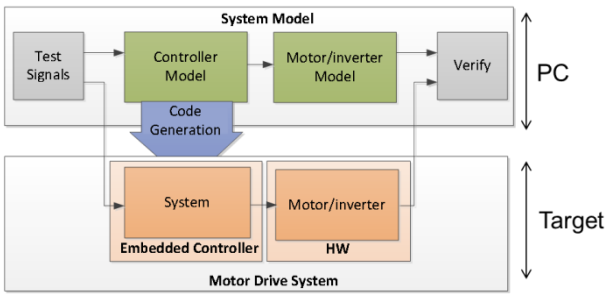
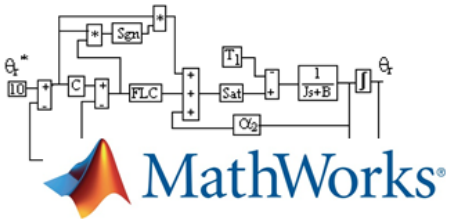
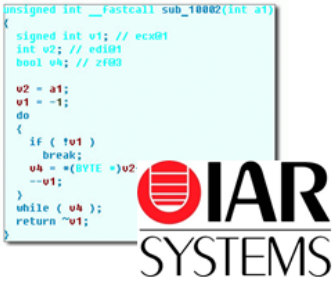


図 4 - MBD セットアップ

本稿を執筆するにあたり使用した実験用のセットアップは、アナログ・デバイセズの ARM® Cortex™-M4 ミックスド・シグナル制御プロセッサをベースとして使用しています。これに IAR および MathWorks のツールを組み合わせ、MBD プラットフォームの完全なインプリメンテーションを実現します。上述の各ステップは、使用可能なツールとインプリメンテーション全体に直接リンクしています。図 5 に示すように、それぞれのツール・チェーンの利用価値はさまざまです。MBD では、これらのツール・チェーンの利用と個々の MBD プラットフォームの

全体的な価値創出とのバランスをどのようにとるかは、設計者が選択しなければなりません。





Strength	- Solving differential equation - Time domain modeling - Visualization	- Register setup and control - Managing System resources - Time scheduling
Weakness	- Target specific setup - Managing system resources - Time scheduling	- Real time control systems - Debugging and test - Visualization

図 5 – MathWorks と IAR システムの長所

ターゲット・プラットフォームに関しては、リアルタイムの開発環境を使用してシステム全体の性能および能力のモデル作成、シミュレート、評価、展開、最適化を行います。これはすべて、MBD およびバランスのとれたシステム・パラメータの選択に基づくもので、特定の最適化が必要な部分についてはクラス最良の柔軟性を発揮します。このようにしてシステムのスケーラブルなモデルができると、既存のレガシー・コードや機能に基づくコードの使用や再利用、標準 C やグラフィカル機能によるまったく新しいビルディング・ブロック（シミュレーションおよび展開フェーズ全体に適合させた Simulink/MATLAB モデル）の作成が容易になります。全体的なセットアップの変更がソフトウェア上でできるだけではありません。システム用の適切なデバイス・ドライバを設計した後に、システム・リソース、ハードウェア要素、最終アプリケーションまたは最終システム用のアプリケーション・ソフトウェア全体を変更することができます。また、全体的なシステム・タイミングのリアルタイム面を制御できるため、この環境上で直接システム・スケジューリングを最適化することができます。

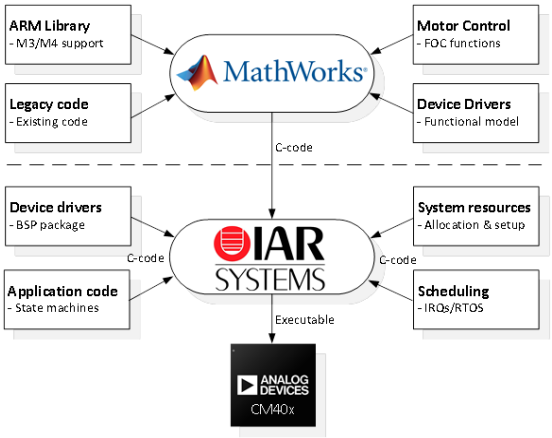


図 6 -展開環境

ドライブ・システムの従来型の模式図をよく調べれば、このアーキテクチャの能力を図示することができます。また、「ドライブ」内の各要素を最適化して、最終システムにとって最も重要な要素に設計の努力を集中することができます。たとえば、保護機能とスケールが最も重要な場合は、電気制御システムや電力システムとともに、機械システムにも重点を置くことになります。シミュレーションの結果とリアルタイム・データを組み合わせてシステム動作を監視し、これによって「ライブ」での最適化を実現します。一方、ノイズによる擾乱がシステムの全体的な効率レベルを低下させている場合は、スケーラブルなフィルタとオブザーバでノイズを測定し、これによってハードウェア・ノイズの問題を最小限に抑え、最適な状態を実現することができます。すべての要素をモデル化してひとつにまとめれば、展開フェーズの最終ステップを開始し、ターゲットシステム上で完全なインプリメンテーション・フェーズを実行することができます。

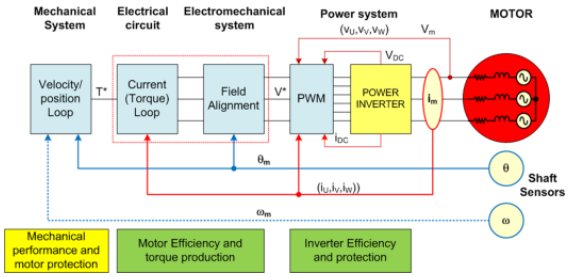


図 7 - 「ドライブ・システム」の模式図

MBD 設計フローでは、MathWorks と IAR を活用することによって全体的なモデルを実装してコードをコンパイルすることができます。ここでは、「ドライブ・システム」のモデルに含まれるそれぞれの段階または要素が MATLAB および Simulink モデルで表現され、適切なレベルの最適設計基準でこれらのモデルがスケーリングされます。モデルの各要素は MathWorks の標準ツールボックスとブロックセットに基づくもので、特定の設計に含まれるあらゆる要素に再利用できます。これらの要素はドライブ・システムのさまざまな領域も表しており、精密な調整を十分に行うことによってモデルと実装済みのシステムの誤差を最小限に抑えることができます。この混合環境におけるリアルタイムの展開とコンパイルを通じ、手書きで作成した既存の C コードと、MATLAB および Simulink 用のプロダクション・コード生成ツール Embedded Coder®によって生成した ARM Cortex M4 用に最適化された C コードを組み合わせることもできます。こうしたプロセス全体において、モーター制御設計に関してユーザーがもつ既存の知識を適切なレベルで再利用できます。この時点で、IAR Embedded Work Bench が生成コードを取り込んで ARM Cortex M4 用の完全なプロジェクトをコンパイルできます。これで、このシステムの MBD インプリメンテーション・フェーズは完了です。

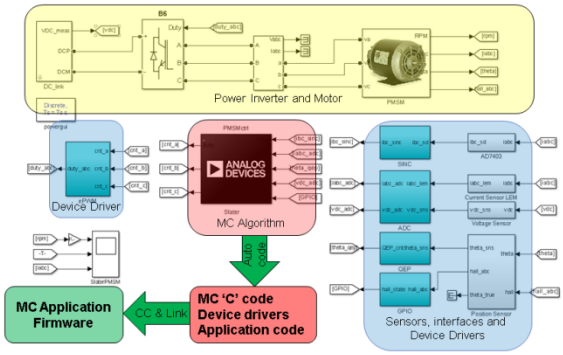


図 8 - インプリメンテーションとコンパイル

当初から MBD については、従来型システム開発に比べた場合の能力と機能、さらにシステム・リソース全体の効率的な利用に関して疑問視されていました。この点において、コンポーネント・サプライヤ、シミュレーションおよびデプロイメント・ベンダー、ツール・コンパイラのプロバイダなどが特に努力を重ね、そ

れらが「融合」された結果、今日では、従来型の展開方法に匹敵するものが実現しています。もちろん、使用するインプリメンテーションの方法によっては、リアルタイム・システム用の開発コード作成が非効率的なものとなってしまう恐れもあります。MBD を使用すれば、安全性が重視されるシステムの開発においてプロファイリング、クロス最適化オプション、その他の強力な利点を組み合わせることができるため、コード開発のオーバーヘッドが最小限に抑えられ、性能において最大限の結果が達成できます。MathWorks は、IEC 61508、ISO 26262、および関連する機能安全標準に基づき Embedded Coder を使用するためのツール認証を受けています。

標準的な設計フローでは、このような能力の組み合わせは非常に難しいものとなります。上の例では、アナログ・デバイセズの ADSP-CM40x シリーズ上に標準 FOC モデルが実装されています。このモデルは、位置および電流ループのフィードバックを 15 µs で実行し、電流方式とデバッグ機能両方のリアルタイム・プロファイリングに対応します。また、FOC 方式全体のトラッキング機能も利用できます。以上により、MBD シミュレーションの結果とリアルタイム・データの両方を評価して、ターゲット仕様に関する理想的なシステム機能と比較することができます。これによって最終的には、設計者はシステムの効率、機能性、性能を継続的に改善するとともに、シグナル・チェーン内の特定の要素やコンポーネントが仕様と比較してどの程度の性能を発揮しているのかを評価することができます。

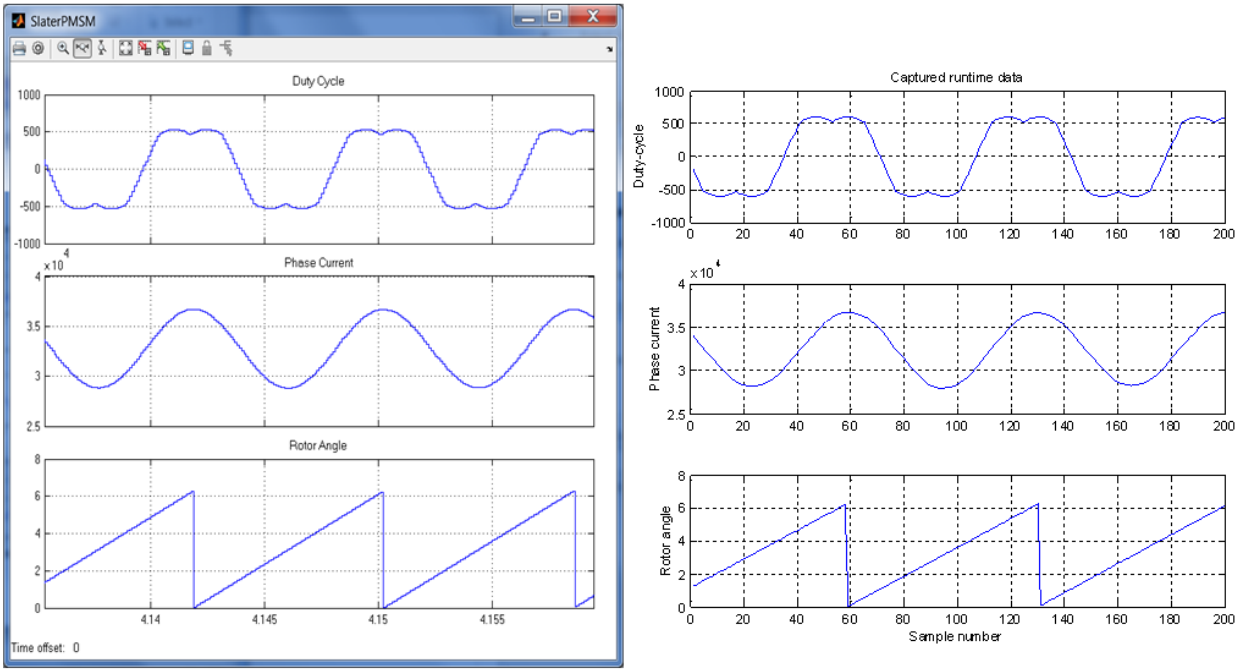


図 9 – モデルベースデザインで作成したシステムのシミュレーションとランタイム・データ（シミュレーション対プロセッサ・データ）

本稿では、MBD を使用してモーター制御システムを構築する「新しい」方法について説明しました。今日の組込みプロセッサは、グラフィックツールの開発と高水準の抽象化を実現できるように性能、コスト、サイズのバランスを図らなければなりません。また同時に、市場投入までの時間、安全性、性能、スケーラビリティなどに関する要求を満たし、高度に最適化されたシステムの基礎とならなければなりません。これらのトピックやアナログ・デバイセズの提供する製品の詳細については、motorcontrol.analog.com/jp をご覧ください。