

ホワイトペーパー

# IEC 62304 準拠の 医療機器向け組み込みソフトウェアの開発

このホワイトペーパーでは、組み込みソフトウェアの開発者や医療機器のエンジニアを対象として、要件管理から設計、実装、統合、検証、テストに至るまでの、医療機器向けソフトウェア設計のベストプラクティスをご紹介します。ソフトウェア品質を向上させながら、設計エラーを低減し、開発期間を短縮する方法について説明します。

チームで手術用ロボットシステムの作動部分の制御を開発しているところを思い浮かべてみてください。システムは、触覚フィードバックの制御を備えた電気モーターで駆動するものとします。設計を始める前に、たとえば次のような主な疑問を解消しておきたいと考えられます。

- 最適な消費電力と精度を実現するモーターのサイズはどれくらいか？
- プロジェクトを遅延させることなく、開発の後期に要件を変更できるか？
- 統合前に設計をシステムレベルでどうやってテストするか？
- 設計プロセスは IEC 62304 などの医療規格に準拠しているか？



Corindus CorPath GRX  
手術用ロボットシステム。

チームが手作業でコードを書き、ドキュメントベースで要件を記録している場合、これらの疑問を解消するには、試行錯誤や物理的なプロトタイプでのテストを行うしかありません。これは、手術器具や人工呼吸器、輸液ポンプ、透析装置などのアプリケーションでも同様です。要件が 1 つでも変更されたら、システム全体を再コーディングしてテストしなければならなくなり、プロジェクトが数週間から数か月遅延することもあります。

手書きのコードやドキュメントの代わりに、MATLAB® および Simulink® によるモデルベースデザイン (MBD、モデルベース開発) を使用すれば、システムモデルを作成できます。たとえば、手術用ロボットシステムの場合は、ロボットアームやモーター、制御ユニットにより構成されるモデルです。どの段階でもモデルをシミュレーションして、リスクや遅延なしで、高価なハードウェアを用いることなく、システム挙動を即座に確認し、複数の what-if シナリオをテストできます。そして、IEC 62304 に準拠したプロセスで、最終製品に統合できます。

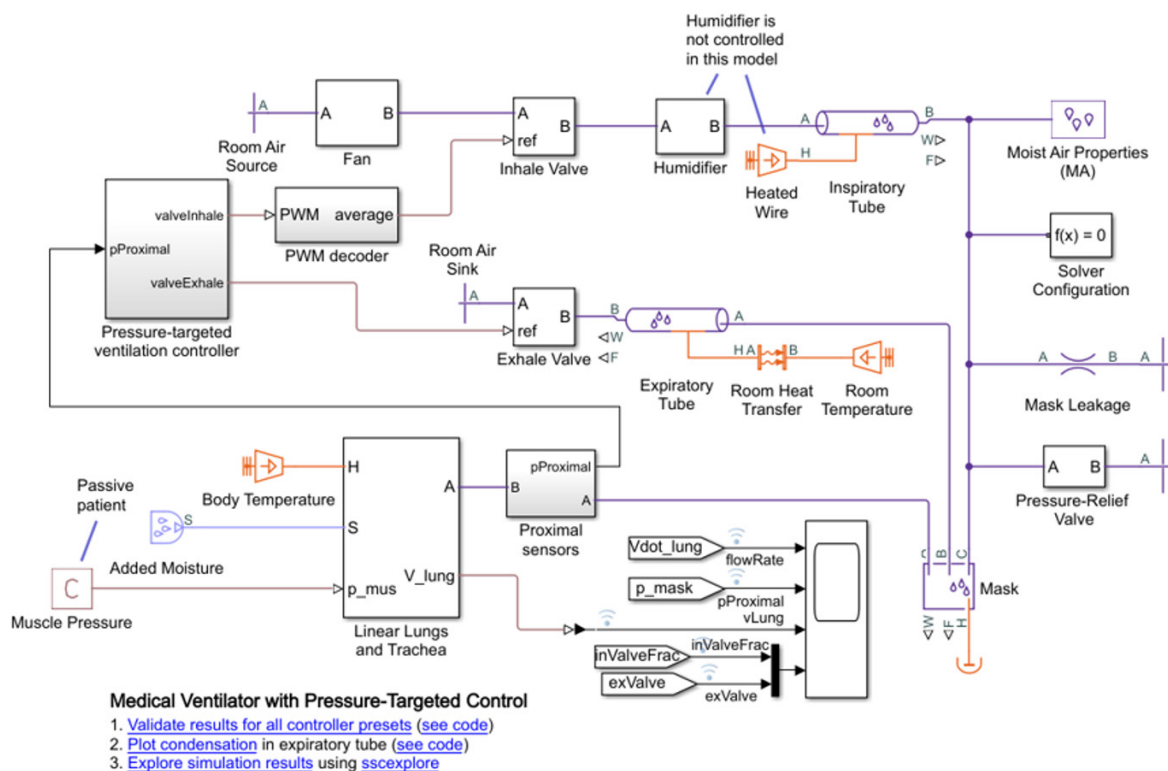
「モデルベースデザインを採用したことで、少人数の製品開発チームでも、わずか 4 か月で遠隔操作ロボットの機能を開発および検証することができ、コスト削減と開発期間の短縮を実現することができました。」

— Per Bergman, Corindus

このホワイトペーパーでは、医療機器の開発向けモデルベースデザインをご紹介しますとともに、導入のためのベストプラクティスについて説明します。モデルベースデザインを採用したチームが、開発期間を短縮し、統合の問題を最小限に抑え、医療機器の規制や標準に準拠しながら安全性の高い製品の提供を実現した方法を見ていきましょう。

## I. モデルベースデザインとは

モデルベースデザインを理解する最良の方法は、実例を見ることです。人工呼吸器の設計を例に説明します。



Simulink と Simscape を使用した人工呼吸器モデル。

医療機器のエンジニアチームが、人工呼吸器の流量や吐出量、吸気/呼気の機能を調整するバルブの制御ユニットの構築に着手している場合を考えてみましょう。モデルベースデザインを採用しているため、エンジニアは、はじめにシステム要件からアーキテクチャモデルを構築します。

次に、チューブやバルブ、加湿器を含む人工呼吸器のモデルを開発します。この高水準かつ低忠実度のモデルには、制御ユニットで実行される制御アルゴリズムと、プラント（この場合は、人工呼吸器を装着した患者）のモデルも含まれます。

チームは、この高水準なモデルを病院や緊急時のさまざまなシナリオでシミュレーションすることにより、システムや統合の初期テストを行い、システムが適切に機能しているか、また患者にリスクを及ぼすことなくさまざまな状況に対応しているかを検証します。

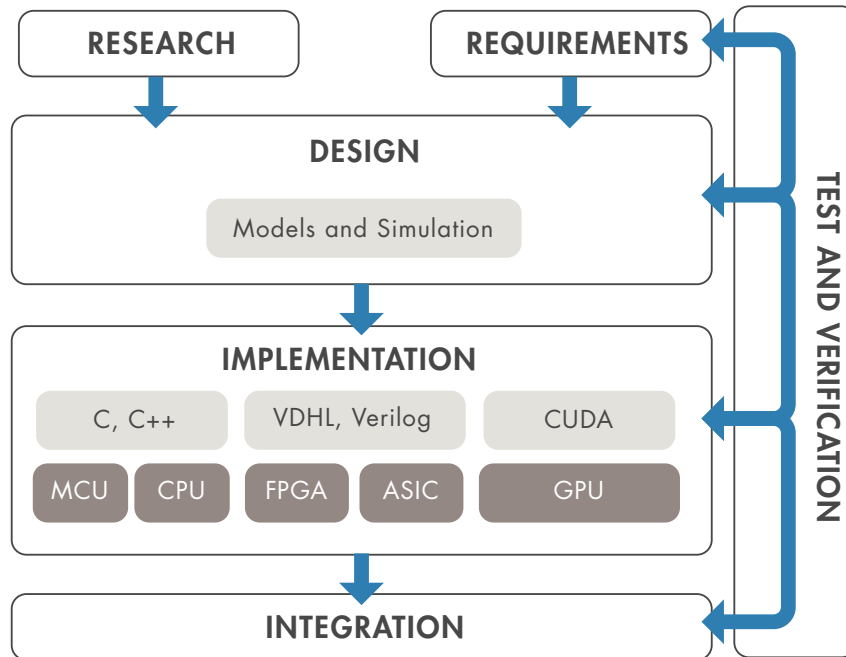
人工呼吸器の応答性をテストするために、実際の患者に使用する場合を想定してシミュレーションに詳細設定を追加します。そして、仕様書と比較しながら、引き続きシステムレベルの挙動のテストや検証を行います。システムが大規模で複雑な場合、エンジニアは個別のコンポーネントを単独で開発およびテストし、さらにシステム全体のシミュレーションでそれらを頻繁にテストすることができます。

最後に、システムとそれが動作する条件を調整したモデルを構築します。このモデルには、研究開発チームのあらゆる活動によって蓄積されたシステムに関する知識 (IP) が取り込まれています。続いて、エンジニアは、ソフトウェアのテストと検証を行う制御アルゴリズムのモデルからコードを生成します。また、規制当局の承認プロセスに必要なユニットテストやシステムレベルテストの文書もすべて作成します。

ラピッド プロトタイピングによるハードウェアインザループ テストを実施した後に、生成されたコードを量産ハードウェアに実装し、人工呼吸器の動作を検証します。

このシナリオから分かるように、モデルベースデザインは従来の開発ワークフローと同じ要素を使用する一方で、主に次の2つの違いがあります。

- 要件の記録から設計、実装、テストに至るまで、システムモデルが開発プロセスの中心となっている。
- 時間がかかるうえに間違いの発生しやすいワークフロー上の手順 (コードの記述、手動テスト、文書作成など) の多くが自動化されている。



モデルベースデザインのワークフロー

### 要件の記録および管理

要件が文書で記録される従来のワークフローでは、引き継ぎにより誤りや遅延が生じることがあります。設計文書や要件を作成するエンジニアと、システムを設計するエンジニアが違う人物であることはよくあることです。要件が「丸投げ」されることもあります。これは、2つのチーム間に明確なまたは一貫したコミュニケーションがないという意味です。

モデルベースデザインでは、Simulink モデル内で要件の作成、解析、管理を行うことができます。また、カスタム属性を使用したリッチテキストの要件を作成し、それを設計、コード、およびテストにリンクさせることができます。要件は、要件管理ツールのような外部ソースからインポートして同期することもできます。設計にリンクされている要件が変更されれば、自動的に通知が送られます。さらに、トレーサビリティ マトリクスを使用して、生成されたコードまで要件をトレースできます。その結果、要件の変更がモデルやコードに及ぼす影響を直接評価し、適切に対処できます。

### ケーススタディ: Corindus



延長アーム、タッチパネル付きの  
Corindus ベッドサイドユニット

「モデルベースデザインは、新機能を開発して迅速に展開できるため、当社の技術革新には欠かせません。プロトタイプを短期間で作成し、要件を満たしていると確認できれば完成させて、製品開発段階に移行できます。」

— Doug Teany,  
Chief Operating Officer, Corindus

Corindus は、数百から数千マイル離れた場所にいる患者に、医師が経皮的冠動脈インターベンション (PCI) や神経血管インターベンション (NVI) を施術できる、ロボティクス プラットフォームを開発および展開しています。

このプラットフォームは、同社の CorPath® GRX システムの拡張版であり、医師がカテーテル検査室の放射線遮蔽されたワークステーションから患者に施術できる仕様になっています。さらに、MATLAB と Simulink によるモデルベースデザインを使用してシステムを構築し、ビデオや制御データのリアルタイム伝送をサポートする機能が追加されました。

Corindus は、オリジナルのシステムでは、シミュレーションでロボット制御設計を検証してからハードウェア開発に着手することで、開発のスピードアップを目指していました。また、リアルタイム シミュレーションおよびテストで設計を検証し、組み込みマイクロコントローラーに実装することも検討されていました。

遠隔操作機能を追加するには、患者のいる場所から X 線検査や血行動態のビデオデータをリアルタイムで医師に送信し、ジョイスティックなどの制御データを返送する必要がありました。このような遠隔操作ロボット機能を組み込むため、Corindus の開発チームは、Simulink Real-Time™ を実行する 2 台の Speedgoat ターゲット コンピューター経由で、遠隔地と現地の間でビデオデータと制御コマンドを送信する通信リンクを構築しました。

遠隔操作によるロボット援用 PCI 手術の初の成功例は、Tejas Patel 博士によるものです。インドのアーメダバードにある Apex Heart Institute において、32 km 離れた場所にいる患者を対象に、2 日間で 5 件の手術を成功させました。

## 設計

従来の手法では、設計のアイデアはどれも物理プロトタイプでテストする必要がありました。その結果、各テストによりプロジェクトの開発期間とコストが増大するため、調査できる設計のアイデアとシナリオの数は限られていました。

モデルベースデザインでは、事実上、アイデアを無制限に検討することができます。要件、システムコンポーネント、IP、およびテストシナリオのすべてがモデルに含まれています。また、モデルをシミュレーションすることができるため、高価なハードウェアを構築する前に、設計上の問題点と疑問点を調査することができます。素早く複数の設計アイデアを評価し、設計トレードオフを調べ、それぞれの設計変更がシステムにどう影響するかを確認できます。

### ケーススタディ: Weinmann



MEDUMAT Transport 人工呼吸器  
画像 © Weinmann Medical Technology.

「複雑さを増した設計にも、MATLAB および Simulink によるモデルベースデザインは対応することができ、認証の取得にも役立ちました。手書きのコードではなくモデルを使用することで、組み込みソフトウェアのメンテナンスや再利用が簡単になり、認証機関に技術を説明する際にも役立ちます。」

— Dr. Florian Dietz, [Weinmann](#)

MEDUMAT Transport 人工呼吸器により、呼吸のサポートを必要とする患者の肺に、空気と酸素を送り込むことができます。この人工呼吸器は、救急医療や院内および院外での搬送時に使用することを目的としています。

MEDUMAT Transport には、圧力、流量、温度、モル質量 (酸素濃度の測定に使用) を測定するさまざまなセンサーが搭載されています。これらのセンサーを高性能な空気弁や電磁弁と組み合わせることで、MEDUMAT Transport は、Weinmann 社がこれまでに開発した中で最も先進的、かつ最も複雑な人工呼吸器となりました。このシステムに最適なアルゴリズムを見つけるために、エンジニアは数多くの設計案を評価する必要がありました。

Weinmann 社のエンジニアは、組み込みソフトウェアを手書きで作成していた従来のプロセスでは、このプロジェクトは実現不可能であると認識していました。

この課題を克服するために開発したプラントモデルには、ハードウェア コンポーネントだけでなく、人間の肺の機械モデルが組み込まれています。さらに、チームはコントローラーとそのステートマシンをモデル化しました。その中には、スタンバイ、スタートアップ、シャットダウンなどの動作モードを追跡するステートマシンと、人工呼吸器プロセス全体を管理する 2 つ目のステートマシンが搭載されています。システムレベルのコントローラーモデルは、サブシステム単位でみたとき、トップレベルにあります。そして、モジュール設計やアーキテクチャなど、基本的な要件をカバーします。

チームは、コントローラーとプラントの閉ループのシミュレーションを行った後、制御システムとセンサー信号処理サブシステムの量産コードを作成しました。このコードを Infineon® 社の MCU と Texas Instruments™ 社の MCU にそれぞれ展開し、モデル内の各サブシステムに対してユニットテストを実施しました。

## コード生成

従来のワークフローでは、組み込みのコードはシステムモデルからまたはゼロから手書きする必要がありました。ソフトウェア エンジニアは制御システムエンジニアが書いた仕様に基づいて制御アルゴリズムを書きます。仕様書の作成、アルゴリズムのコーディング、手書きのコードのデバッグといった、プロセスの手順は、時間がかかるだけでなく、エラーが発生しやすくなる可能性があります。

モデルベースデザインでは、数千行のコードを手書きするのではなく、モデルから直接コードを生成します。手書きのコーディングによるエラーがなくなり、モデルが制御システムのエンジニアとソフトウェア エンジニアの橋渡しをします。生成したコードは、プロトタイピングまたは量産用に使用できます。また、特定のプロセッサ アーキテクチャに対して最適化するか、手書きのレガシーコードと統合できます。

### ケーススタディ: World of Medicine



WOM 社の 50L の気腹装置。

腹腔鏡検査などの低侵襲的な治療手段は、腹部の狭い空間で行う必要があります。手術器具の動きの自由度を高めるには、気腹装置を使用し、CO<sub>2</sub> ガスを吹き込んで体腔を拡張します。腹腔鏡検査や子宮鏡検査に使用されている気腹装置やポンプの技術における業界大手である WOM 社は、モデルベースデザインを使用して、高品質な気腹装置の制御ソフトウェアの開発を加速しています。

WOM のエンジニアは、過去の同様のプロジェクトで、コードを手書きするという従来の開発ワークフローを採用していました。この手法では、プロセスの終盤になるまで設計やコーディングのエラーを発見して修正するのが難しいため、ソフトウェアの納品に遅延が生じていました。

WOM のエンジニアは、新しい気腹装置にモデルベースデザインを採用しました。また、測定された入出力データを使用して腹腔の非線形数理モデルを作成し、このモデルを圧力センサーやアクチュエータなどのハードウェアコンポーネントを搭載したプラントモデルに組み込みました。

次に、カスケード接続された 2 つの比例積分 (PI) 制御器 (流量用と圧力用) を使用して制御モデルを開発しました。チームは、制御機能を検証するために、この制御モデルとプラントモデルの閉ループのシミュレーションを行いました。

この設計のリアルタイム性能を検証するために、制御モデルから C コードを生成し、気腹装置のプロトタイプのセンサーやアクチュエータに接続されたリアルタイム ハードウェアに展開しました。顧客の意見をもとに設計を改良した後、ターゲットとなる Arm® Cortex-M® プロセッサ用の量産コードを作成しました。

WOM は、包括的な統合テストとシステムレベルのテストを行った後、FDA および欧州の規制当局から新しい気腹装置の承認を取得しました。現在は量産され、実際の臨床に使用されています。

「Simulink を使うことで、安定した制御システムを短時間で作成することができました。ステートマシンやカスケード接続された PI 制御を含む、システム全体をモデル化しました。このモデルをロバスト化しつつ、応答時間を短縮した後、ラピッドコントロールプロトタイピングで検証し、組み込みコードを生成しました。」

— René Pätznick, WOM

## テストと検証

従来の開発ワークフローでは、テストや検証はアプリケーションが完成してから始めるのが一般的で、設計やコーディングの段階で生じたエラーを特定して修正することは困難でした。

モデルベースデザインでは、要件や仕様のモデル化を開始してから、完成した設計を統合する準備が完了するまでの、開発サイクル全体を通してテストと検証を行います。テストの頻度や項目は増えますが、設計が要件を満たしていることを証明できるので、時間の短縮につながります。モデルに取り込まれた要件を検証し、設計、テスト、コードへとトレースすることができます。静的解析や形式的手法を使用した、テストの自動生成、テストレポートの作成、コーディング標準やガイドラインへの準拠の確認などが可能です。

### ケーススタディ: ITK Engineering



ITK Engineering のセンサーなしのブラシレス  
モーター制御を採用した歯科用ドリル。

「プラントモデルがモーターの動作を正確に反映しているため、コントローラーやハードウェアの検証を開発の早い段階で行うことができました。その結果、ハードウェアの最初の試作でエラーの根本的な原因を早急に突き止めることができました。」

— Alexander Reiss, ITK Engineering

センサーなしのブラシレス DC (BLDC) モーターは、ブラシ付きモーターの動作に比べて摩耗が抑えられ、信頼性や静粛性に優れ、メンテナンスや滅菌のしやすさなどが向上しています。また、センサーなしのブラシレス DC (BLDC) モーターは、センサーありの BLDC モーターと比較して、安価で軽量な点が特徴です。ただし、センサーなしの制御に求められる複雑なアルゴリズムを開発するには、技術的な手間がかかります。

ITK のエンジニアは、ローター位置推定器のほか、医療機器ソフトウェアの IEC 62304 規格に準拠する歯科用ドリル モーターの高度なカスケード制御を設計および最適化する必要がありました。

プロジェクト開始時には、プロトタイプのコントローラーはありませんでした。クライアント指定のプロジェクト納期に間に合わせるために、ITK はモーターのハードウェアと並行して制御器のソフトウェアを開発する必要がありました。

ITK のエンジニアは、モデルベースデザインを使用して、センサーなしのブラシレス DC (BLDC) モーター コントローラーの設計、最適化、実装、テストを行いました。既存のコントローラーのデータシートや顧客からの情報をもとに、電気的および機械的なコンポーネントを備えた BLDC モーターのモデル化を Simulink で行いました。

浮動小数点コントローラーの設計を固定小数点に変換した後、シミュレーションを繰り返して固定小数点モデルを検証しました。また、MATLAB スクリプトも作成し、個々のモデルコンポーネントのバッチユニットテストを行いました。

このコントローラーとセンサーなしの BLDC モーターは、現在、歯科用ドリルで連続生産されています。



## II. モデルベースデザインの始め方

モデルベースデザインに移行する利点を理解する一方で、それにまつわる組織的、ロジスティック的、および技術的なリスクと課題の発生を懸念する声がチーム内からあがるかもしれません。このセクションでは、モデルベースデザインの導入を検討中のエンジニアリング チームからよく寄せられる質問にお答えし、チームが移行を管理するのに役立つヒントとベストプラクティスをご紹介します。

### Q. モデルベースデザインの導入はエンジニアリングの役割にどのように影響しますか？

A. モデルベースデザインは、制御設計とソフトウェア アーキテクチャにおけるエンジニアリングの専門知識を置き換えるものではありません。モデルベースデザインにより、制御エンジニアの役割は紙での要件を提供することから、モデルとコードの形式で実行可能な要件を提供することへと広がります。ソフトウェア エンジニアがアプリケーション ソフトウェアのコードを手書きする時間は短縮され、アーキテクチャのモデリング、OS、デバイスドライバー、その他のプラットフォームのソフトウェアのコーディング、およびシステム統合の実施に費やす時間は増えることになります。制御エンジニアとソフトウェア エンジニアは、ともに開発プロセスの初期段階からシステムレベルの設計に関与することになります。

### Q. モデルベースデザインに移行すると、既存のコードはどうなりますか？

A. 設計の一部となります。つまり、システムモデルに最初からモデリングされたコンポーネントと、レガシー コンポーネントの両方を含めることができます。そのため、システム シミュレーション、検証、およびコード生成を継続しながら、レガシー コンポーネントを段階的に導入することができます。

### Q. 推奨されているモデルベースデザインの導入方法がありますか？

A. 新しい手法や設計ツールを試す際には、常にリスクとなる要素が伴います。成功するチームはモデルベースデザインを段階的に導入し、プロジェクトを遅延させることなく、プロジェクトに役立つ可能性のある焦点を絞った手順を行うことで、このリスクを軽減してきました。どのような規模の企業でも、小さなグループのレベルからモデルベースデザインの導入を始めます。通常、1つのプロジェクトから始め、短期で成功させて、それを基礎とします。経験を積んだ後、モデルベースデザインを部門レベルに展開し、モデルがグループの組み込みシステム開発の中心となるようにします。

### Q. モデルベースデザインは、IEC 62304 ソフトウェア開発プロセスに準拠していますか？

A. モデルベースデザインを活用することで、IEC 62304 に準拠した医療機器向け組み込みソフトウェアを開発できます。モデルベースデザインは検証と妥当性確認をワークフローに組み込んでいるため、ソフトウェアを医療機器に統合する前に、テストおよび検証を徹底的に行うことができます。また、IEC 62304 で要求される文書の一部も自動生成されます。モデルベースデザインで 사용되는ほとんどの Simulink ツールは、IEC 62304 に準拠した開発ワークフローに対応する TÜV SÜV 認証を受けています。

次の4つのベストプラクティスは多くのチームを成功に導いています。

- **プロジェクトの小さい部分で実験する。**はじめて導入する際は、組み込みシステムの新しい分野を選び、ソフトウェア挙動のモデルを構築し、モデルからコードを生成することを推奨します。チームメンバーは新しいツールと手法の学習に投資する時間を最小限にして、小規模な変更を行うことができます。この結果で、モデルベースデザインの次の主な利点を実際に確認できます。
  - 手書きでコーディングしなくても高品質なコードが作成できる。
  - コードがモデルの挙動と一致する。
  - モデルをシミュレーションすることで、アルゴリズムの欠陥をより簡単に発見できる。また、デスクトップでCコードを動的にテストするよりも、詳細を確認しやすい。
- **システムレベルシミュレーションを追加して最初のモデル化における成功を基礎とする。**前のセクションで述べたように、システムシミュレーションを用いて要件を検証し、設計の疑問点を調査し、早期にテストおよび検証を行うことができます。システムモデルは高忠実度である必要はありません。インターフェイスの信号が正しい単位で正しいチャンネルに接続され、システムの動的挙動を取得していることを確認できるだけの情報があれば十分です。シミュレーション結果により、プラントとコントローラーがどのように動作するかを早期に確認することができます。
- **モデルを使用して特定の設計上の問題を解決する。**チームは、プラント、環境、およびアルゴリズムのモデルをフルスケールで開発することなく、希望する機能を利用することができます。たとえば、チームは動作に使用されるソレノイドのパラメーターを選択する必要があります。この場合、ソレノイドの周りに概念的な「制御ボリューム」を描く単純なモデルを、その駆動力と、それが動作する環境を含めて開発することができます。チームはさまざまな極限の運用条件をテストし、方程式を算出することなく基本的なパラメーターを算出できます。このモデルはその後、異なる設計の問題に、または将来のプロジェクトで使用するために保存しておくことができます。
- **モデルベースデザインのコア要素から始める。**モデルベースデザインの導入当初に得られる利点として、コンポーネントモデルやシステムモデルの作成、シミュレーションを使用した設計のテストと検証、プロトタイプ作成とテストのためのCコードの自動的生成などの機能を上げることができます。その後、高度なツールや実践を検討し、モデリングガイドライン、コンプライアンスチェックの自動化、要件のトレーサビリティ、およびソフトウェアビルドの自動化を導入することができます。

## ケーススタディ: Khawaja Medical Technology



Simulink でモデル化された ECG  
信号解析アルゴリズム。

「モデルベースデザインに対する投資からは、品質の向上、開発期間の短縮、ISO や IEC 認証取得の迅速化など、非常に大きなリターンが得られました。」

— Dr. Antoun Khawaja,  
Khawaja Medical Technology

心電図 (ECG) データの解析は、心疾患の診断や治療に不可欠なものです。前臨床、臨床、外来、在宅など、各種環境での診断や、新薬の臨床試験でも活用されています。

製薬会社には、心筋梗塞治療薬の開発および承認プロセスの一環として、新薬が心臓に及ぼす影響の調査が義務づけられています。この調査には、ECG 信号の解析による異常の特定と、心疾患治療薬の安全性確認が含まれます。

Khawaja Medical Technology のエンジニアは、ECG 信号の解析を完全に自動化する斬新かつ高度なアルゴリズムを開発しました。このアルゴリズムにより、安静時、運動時、ホルターモニター装着時の ECG 信号をリアルタイムでモニタリングし、解析することが可能になります。エンジニアリング チームは、MATLAB と Simulink によるモデルベースデザインを使用して、ECG 自動解析用アルゴリズムを開発および展開しました。

また、Simulink Check™ を使用して、モデルが IEC 62304 などのモデリング ガイドラインや規格に準拠しているかどうかをチェックしました。Simulink Test™ でシミュレーション ベースのテストを作成および実行し、テストを要件にトレースし、Simulink Coverage™ でテストカバレッジを測定しました。

さらに、MATLAB クラス群も開発し、信号処理層および分類層で再利用可能な System object™ の作成に使用しました。これらの System object は、ECG 信号のピークの検出、信号特性の測定、不整脈の分類、心室肥大や心筋梗塞などの心疾患の診断など、各種のタスクを実行します。

これらのツールを活用することで、Khawaja Medical Technology は、開発期間を 40% 短縮し、ISO 13485 および IEC 62304 への準拠を加速させ、数年ではなく数か月でプロトタイプを構築できました。

## モデルベースデザインのツール

### 基本製品

#### *MATLAB*

データの解析、アルゴリズムの開発、および数学的モデルの作成

#### *Simulink*

組み込みシステムのモデル化とシミュレーション

### 要件の記録および管理

#### *Simulink Requirements*

要件の作成、管理、およびモデル、生成コード、テストケースへのトレース

#### *System Composer*

システムおよびソフトウェア アーキテクチャの設計と解析

### 設計

#### *Simulink Control Design*

モデルの線形化と制御システムの設計

#### *Stateflow*

ステートマシンおよびフローチャートを使用した判定ロジックのモデル化とシミュレーション

#### *Simscape*

マルチドメイン物理システムのモデル化およびシミュレーション

### コード生成

#### *Simulink Coder*

Simulink モデルと Stateflow モデルからの C および C++ コードの生成

#### *Embedded Coder*

組み込みシステム用に最適化された C および C++ コードの生成

#### *HDL Coder*

FPGA と ASIC 設計用 VHDL<sup>®</sup> および Verilog<sup>®</sup> コードの生成

#### *GPU Coder*

NVIDIA GPU 向けの CUDA コードを生成

## テストと検証

### [Simulink Test](#)

シミュレーションベースのテストの開発、管理、実行

### [Simulink Check](#)

設計品質を測定し、検証作業を追跡して、規格への準拠を検証

### [Simulink Coverage](#)

モデルおよび生成コードのテストカバレッジを測定

### [Polyspace® 製品](#)

C/C++ コードに重大なランタイムエラーがないことの証明

### [Simulink Design Verifier](#)

設計エラーの特定、要件準拠の証明、テストの生成

### [IEC Certification Kit](#)

IEC 62304 認証のためのコード生成ツールと検証ツールの適格性確認

## 関連情報

以下のリソースは、ご所属のチームがモデルベースデザインを短期間で学習するのに役立ちます。

### インタラクティブ チュートリアル

#### [MATLAB 入門](#)

#### [Simulink 入門](#)

#### [Stateflow 入門](#)

#### [Simscape 入門](#)

### ビデオ

[Simulink とは](#) (2:15)

[MATLAB と Simulink によるモデルベースデザイン](#) (2:08)

[制御向け Simulink 入門](#) (11:30)

### オンサイトまたは自己学習形式のトレーニングコース

[MATLAB 基礎](#)

[Simulink 基礎](#)

[MATLAB と Simulink による制御設計](#)

### その他のリソース

[技術コンサルティング サービス](#)

[医療機器向け MATLAB および Simulink](#)