

BMS (バッテリーマネジメントシステム) のモデルベースデザイン

MathWorks Japan

アプリケーションエンジニアリング部 (制御)

新帯 俊信

はじめに

- 本Webinarでは、MATLABバージョンはR2020aを用いています。
- 本Webinarでは、非常に広範囲のツールボックスに跨る紹介を行います。どの機能がどのツールボックスのものなのか、については、スライド末尾のツール一覧の資料をご参照ください。

背景

背景

近年、大容量のバッテリーが様々な所で使われている。
特に、航空機や車両など、一つの不具合で大事故に発展しかねない機器にも搭載されている。



品質要求

JIS 8715-2: 2012

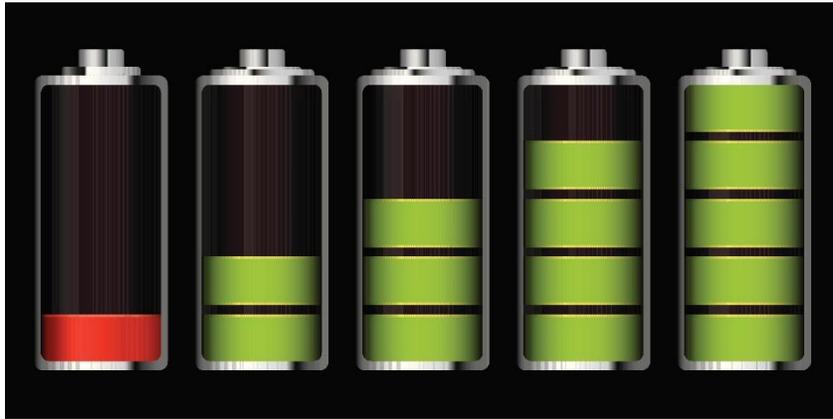
- 製品安全性(非機能)
 - 耐外部短絡特性、耐衝突特性、耐加熱特性、...
- 機能安全性
 - 過充電電圧制御特性、過大充電電流制御特性、充電時加熱制御特性

各社独自規格、限界試験

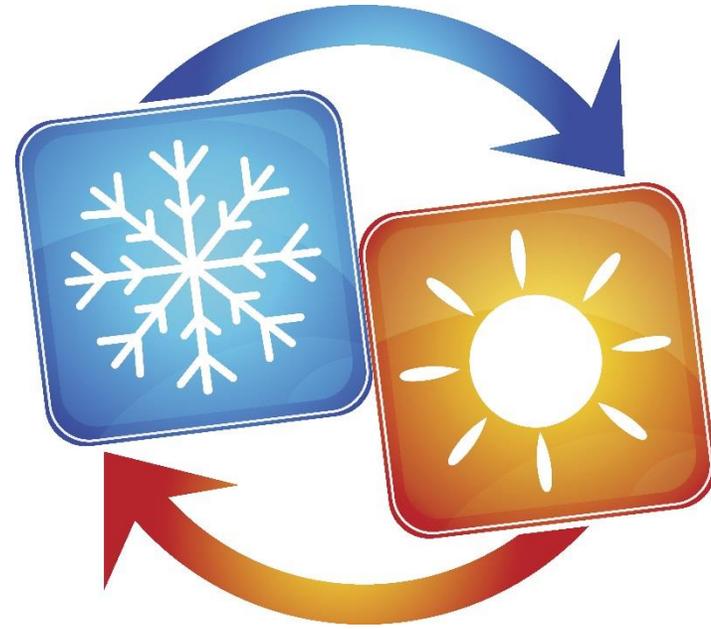
- ○○安全性
 - △ △特性、...

技術課題

非線形特性



溫度依存性



經年劣化



必要な機能

多種類、複雑な機能構成

- SOC/SOHの推定機能
 - カルマンフィルタ、分類/回帰機械学習、ディープラーニング...
- 過電流、過電圧、温度異常時の保護機能
 - 過充電、過放電、過剰電流、短絡検知、SOC上下限、温度異常...
- 状態遷移機能
 - 保護モード、検査モード、チャージモード...
- セルバランス制御機能

バッテリーマネジメントシステム(BMS)の全体像

Software

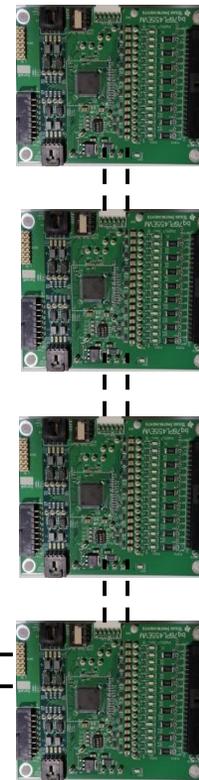
```

if (((uint32_T)State_Machine_DW.temporalCounter_i3) < 15U) {
    State_Machine_DW.temporalCounter_i3 = (uint8_T)((int32_T)((int32_T)
        State_Machine_DW.temporalCounter_i3) + 1));
}

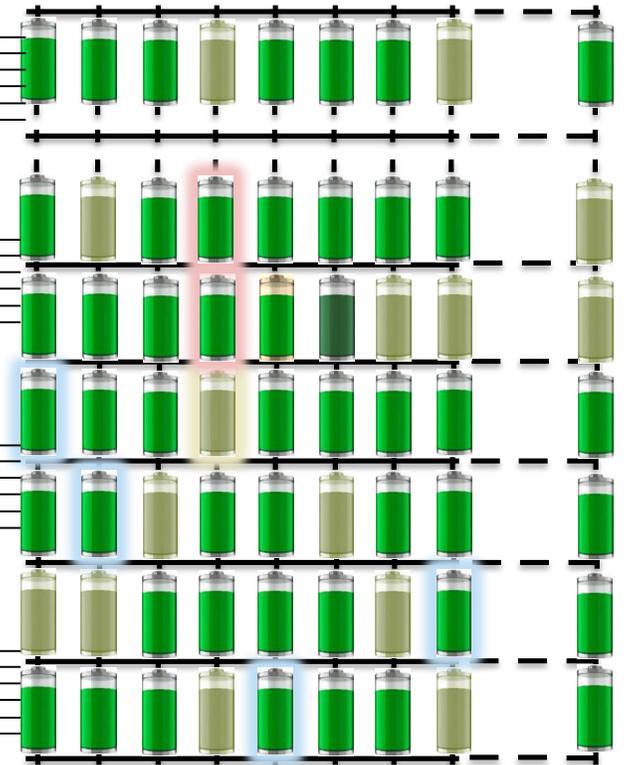
if (((uint32_T)State_Machine_DW.is_active_c2_State_Machine) == 0U) {
    State_Machine_DW.is_active_c2_State_Machine = 1U;
    State_Machine_DW.is_MainStateMachine = State_Machine_IN_Standby;
    *rtu_BMS_State = 0;
    State_Machine_DW.MonitorCurrLimMode = MonitorCurrLimModeType_NoCurrLimFaul
    State_Machine_DW.MonitorCellVoltageMode =
        MonitorCellVoltageModeType_NoCellVoltFault;
    State_Machine_DW.Delta = (real32_T)fabs((real_T)((real32_T)
        ((*rtu_Pack_Voltage) - sum_gyOCKAG3(rtu_Cell_Voltages))));
    State_Machine_DW.FaultPresent = false;
}
    
```



Electronics

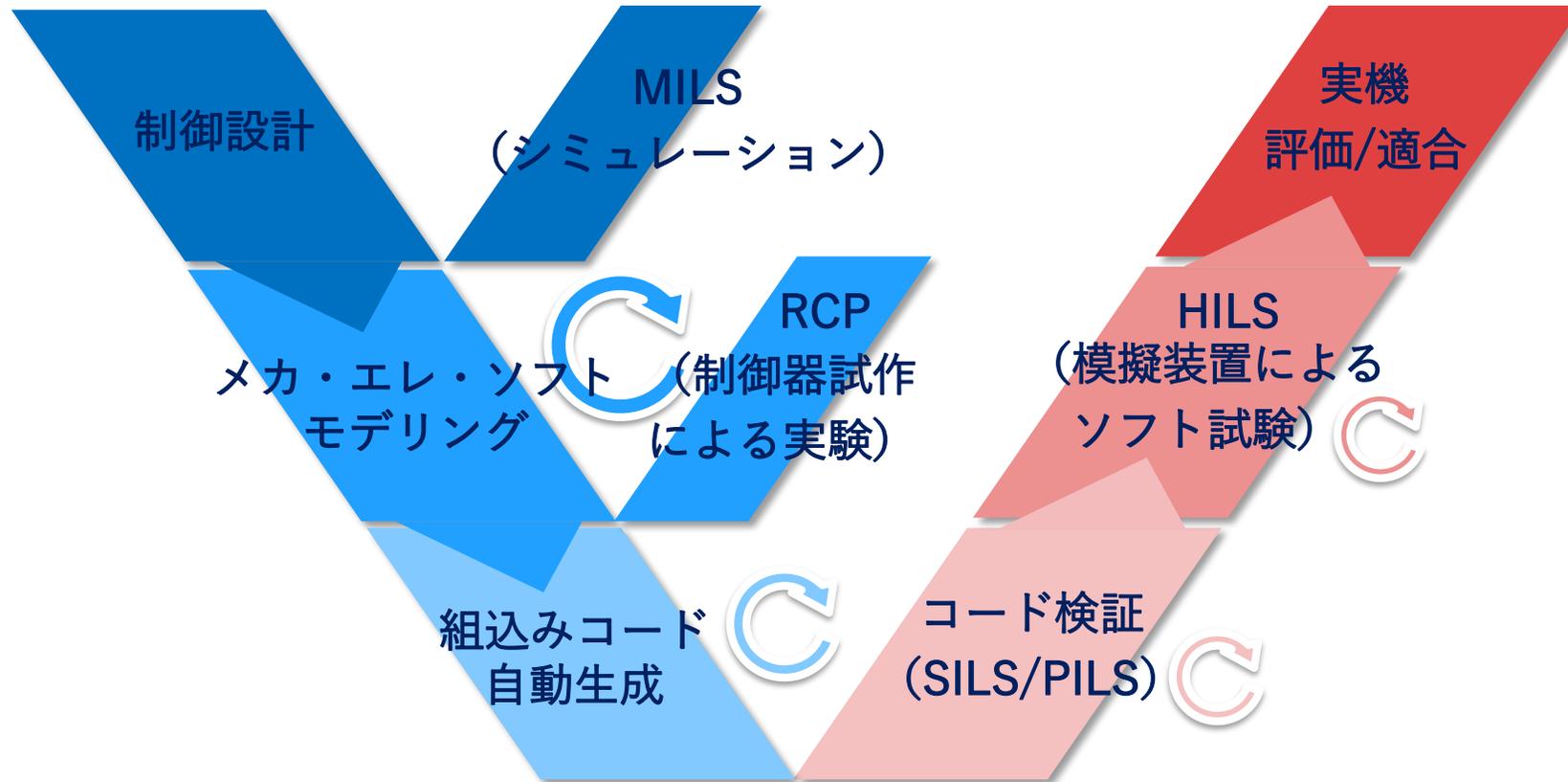


Battery Pack



BMS開発にはMBDが有効

→ MATLAB/Simulinkを使いましょう！



MILS: Model In the Loop Simulation

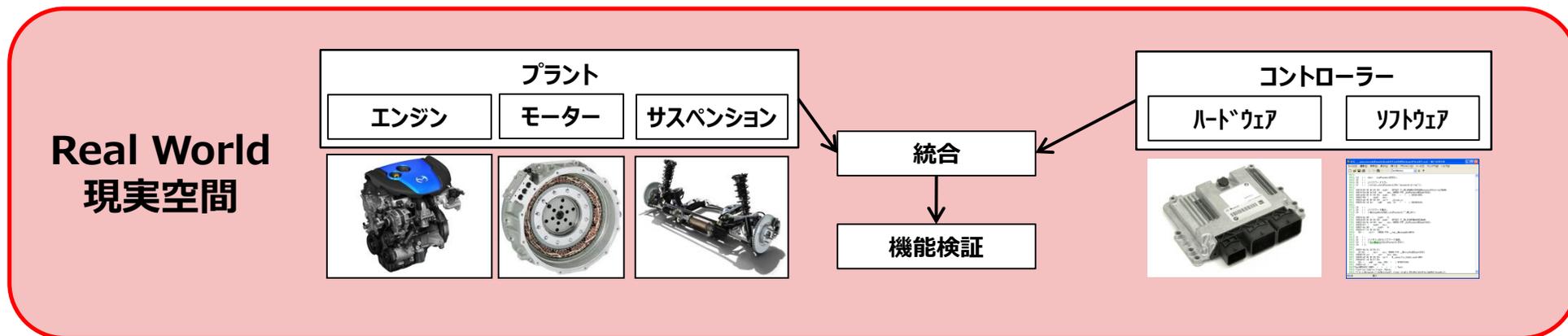
SILS: Software In the Loop Simulation

PILS: Processor In the Loop Simulation

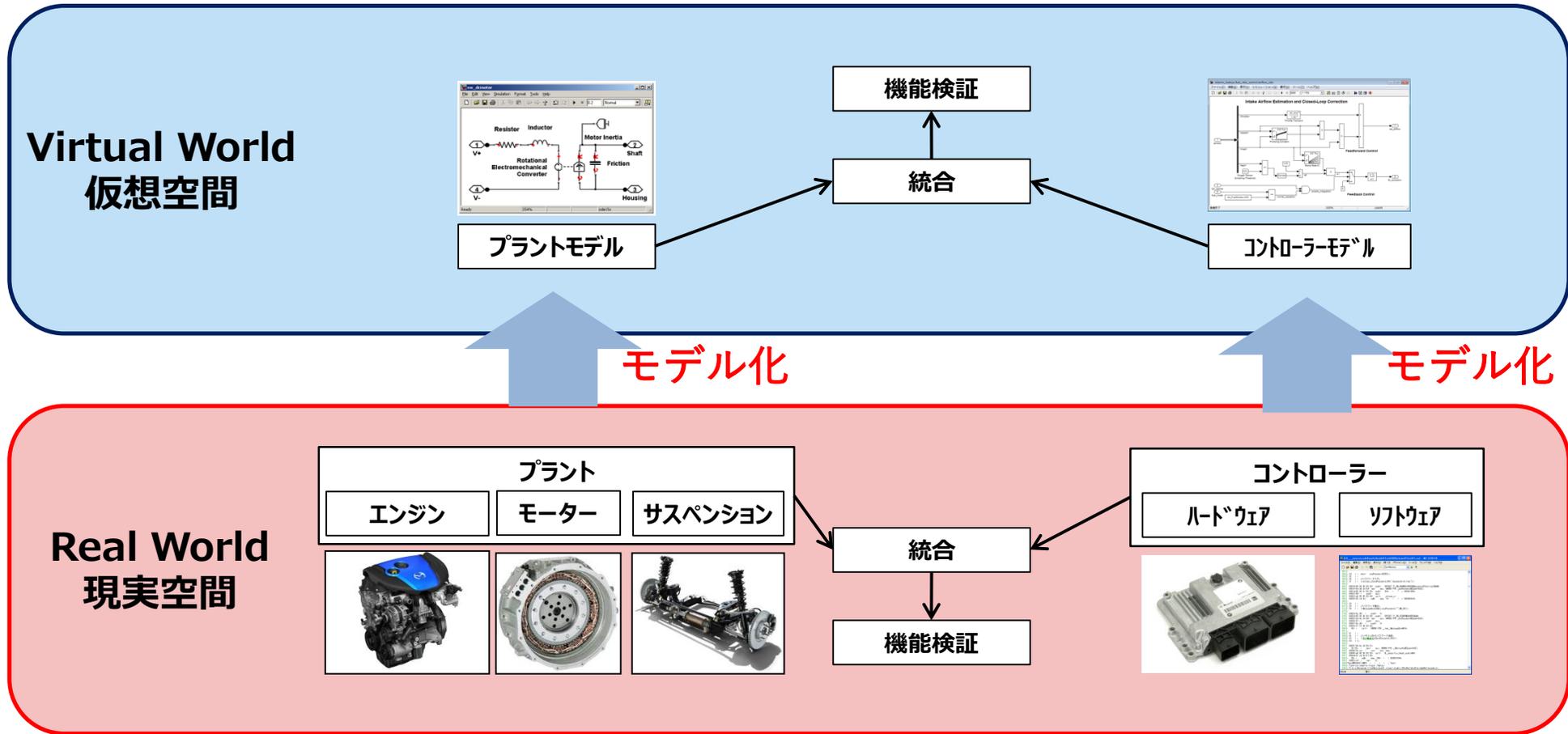
RCP: Rapid Control Prototyping

HILS: Hardware In the Loop Simulation

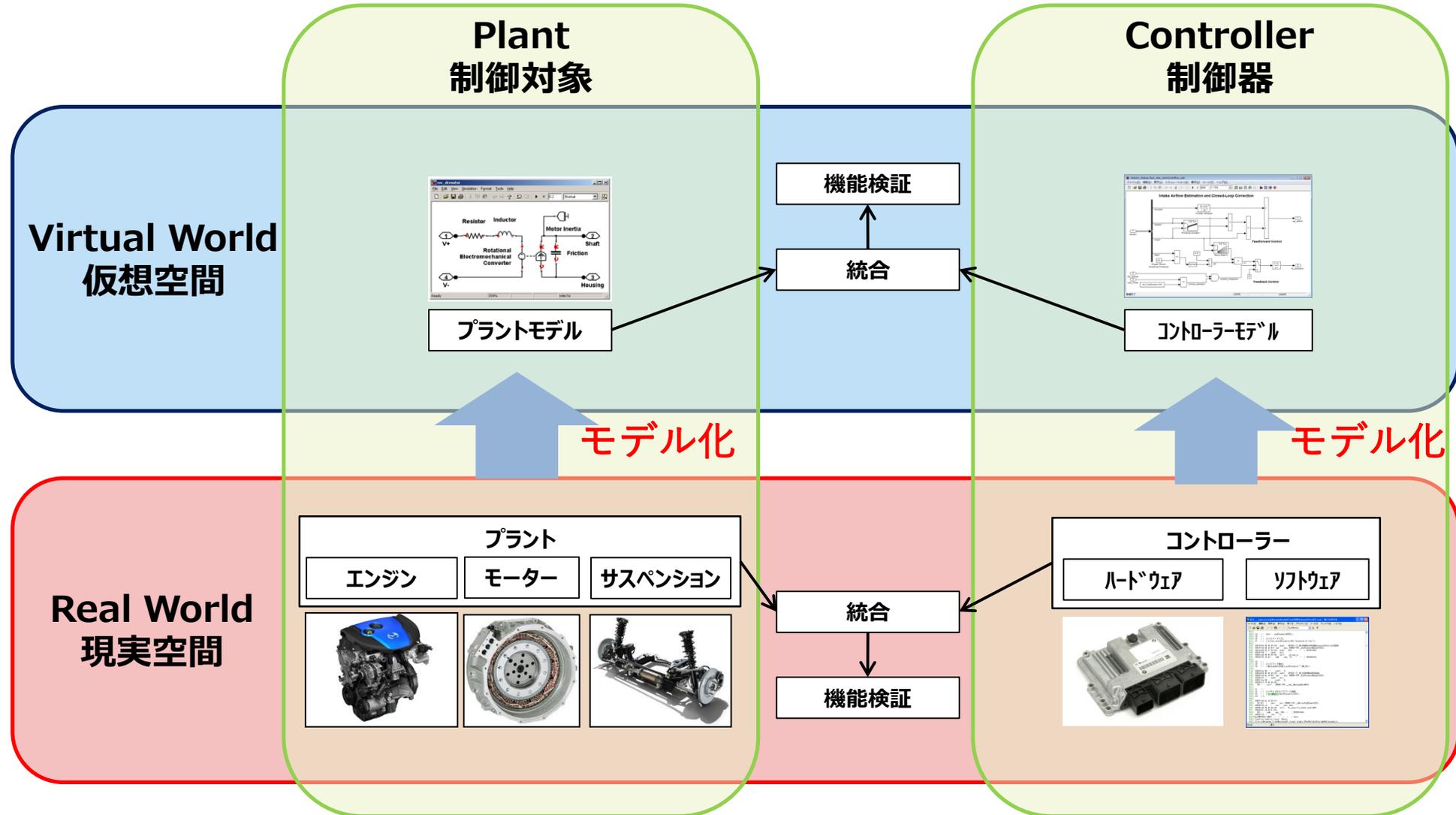
Model-Based Designとは？



Model-Based Designとは？

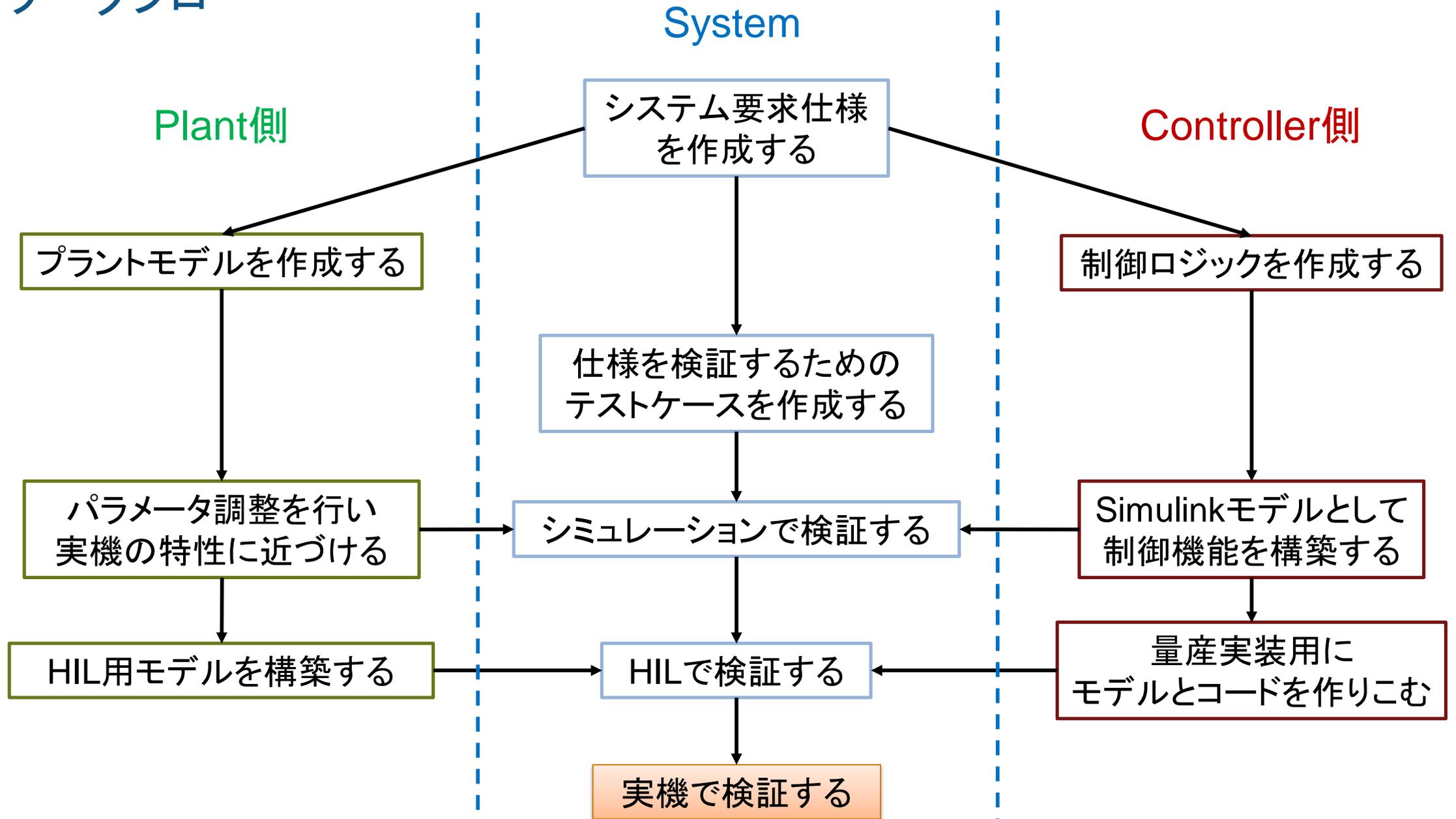


Model-Based Designとは？

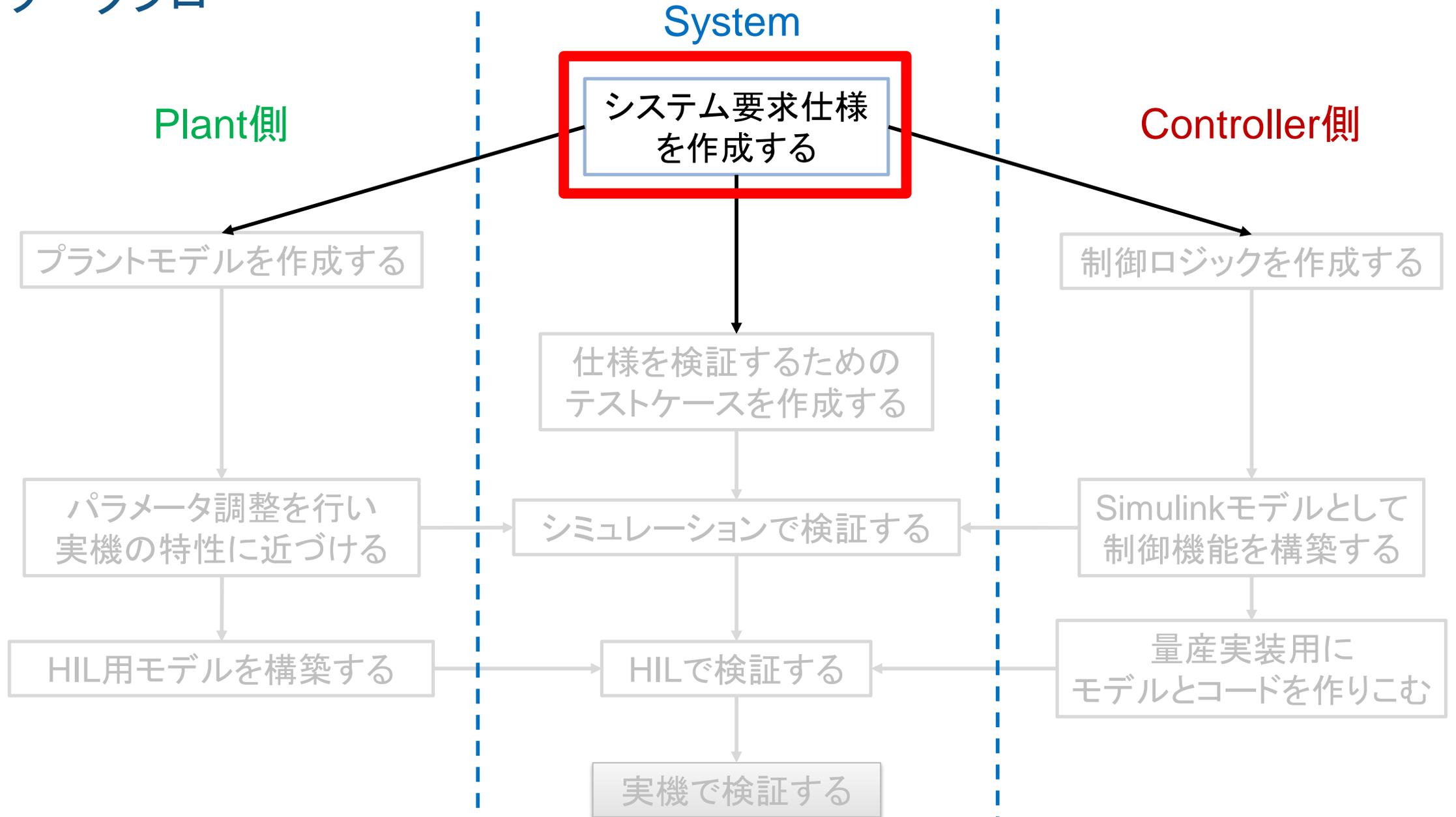


開発ワークフロー

ワークフロー



ワークフロー



システム要求仕様を作成する

要求

要件定義書

JIS 8715-2: 2012

⋮



- 最初は製品のイメージが定まらないので、絵を描いて考えてみる
- 試しにモデルを作ってみて動作を見てみる
- 試しに実機で制御してみる(RCP)

仕様書

BMS制御機能仕様書

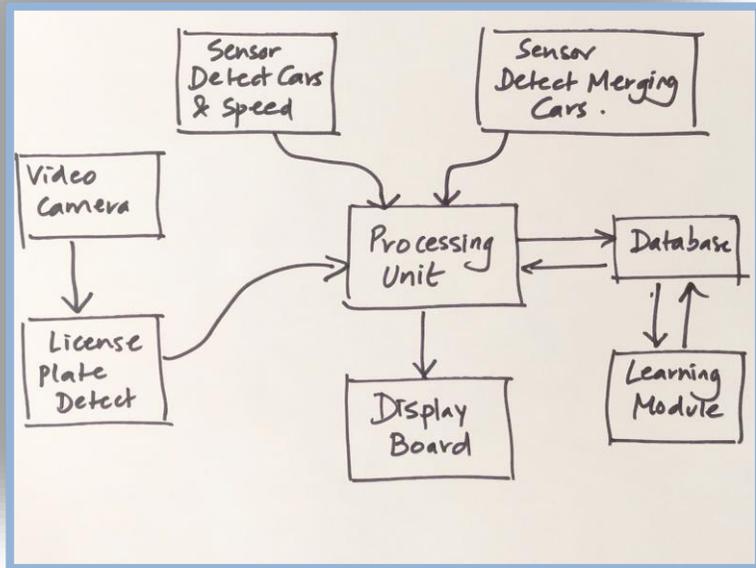
モジュール回路仕様書

⋮



ここは、後の開発工程でかかる工数や製品品質に大きく影響するフェーズ

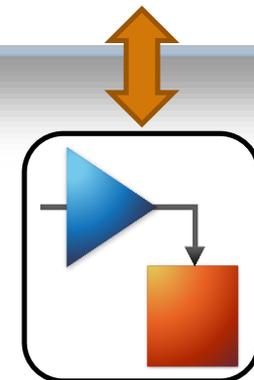
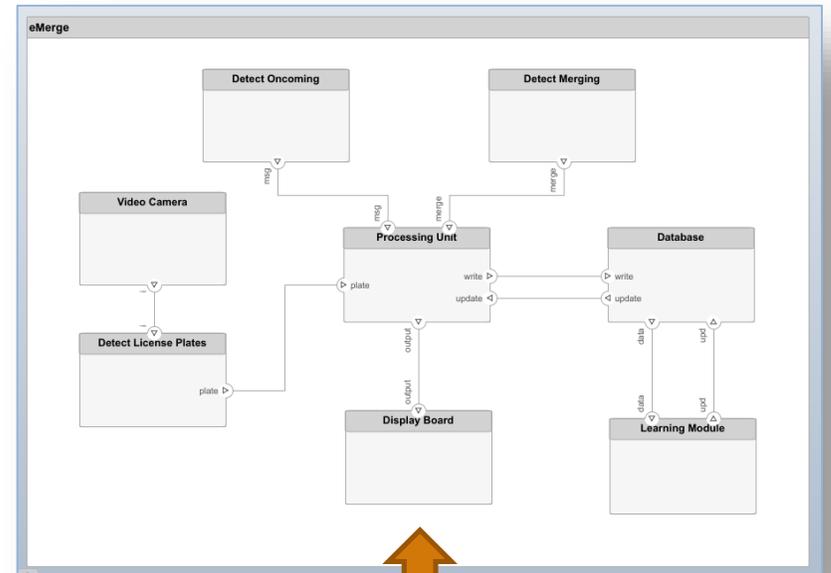
システム要求仕様を作成する システムの絵を描いて考える



システム設計のラフスケッチ

描画
==
アーキテクチャ

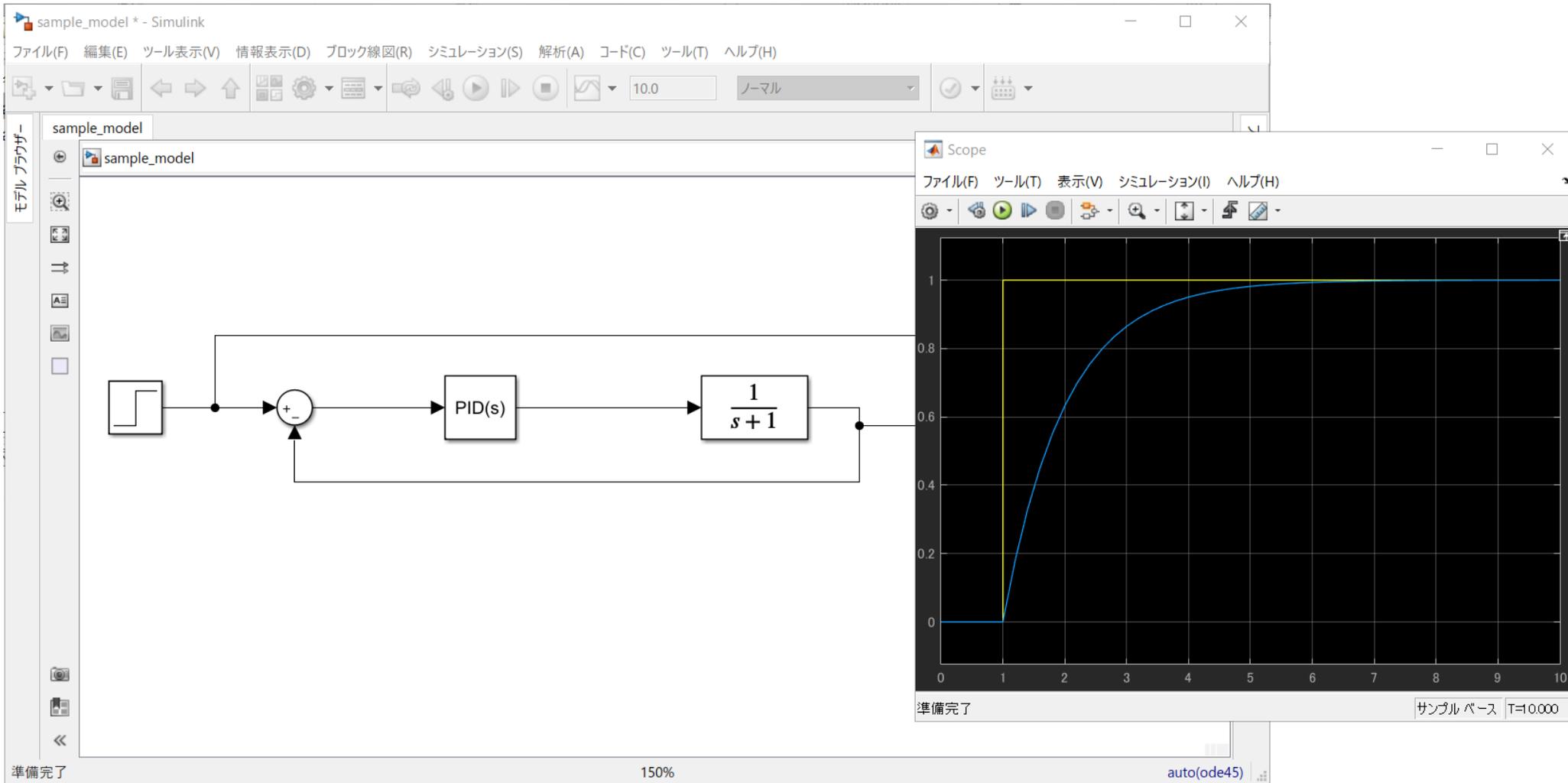
アーキテクチャ モデル



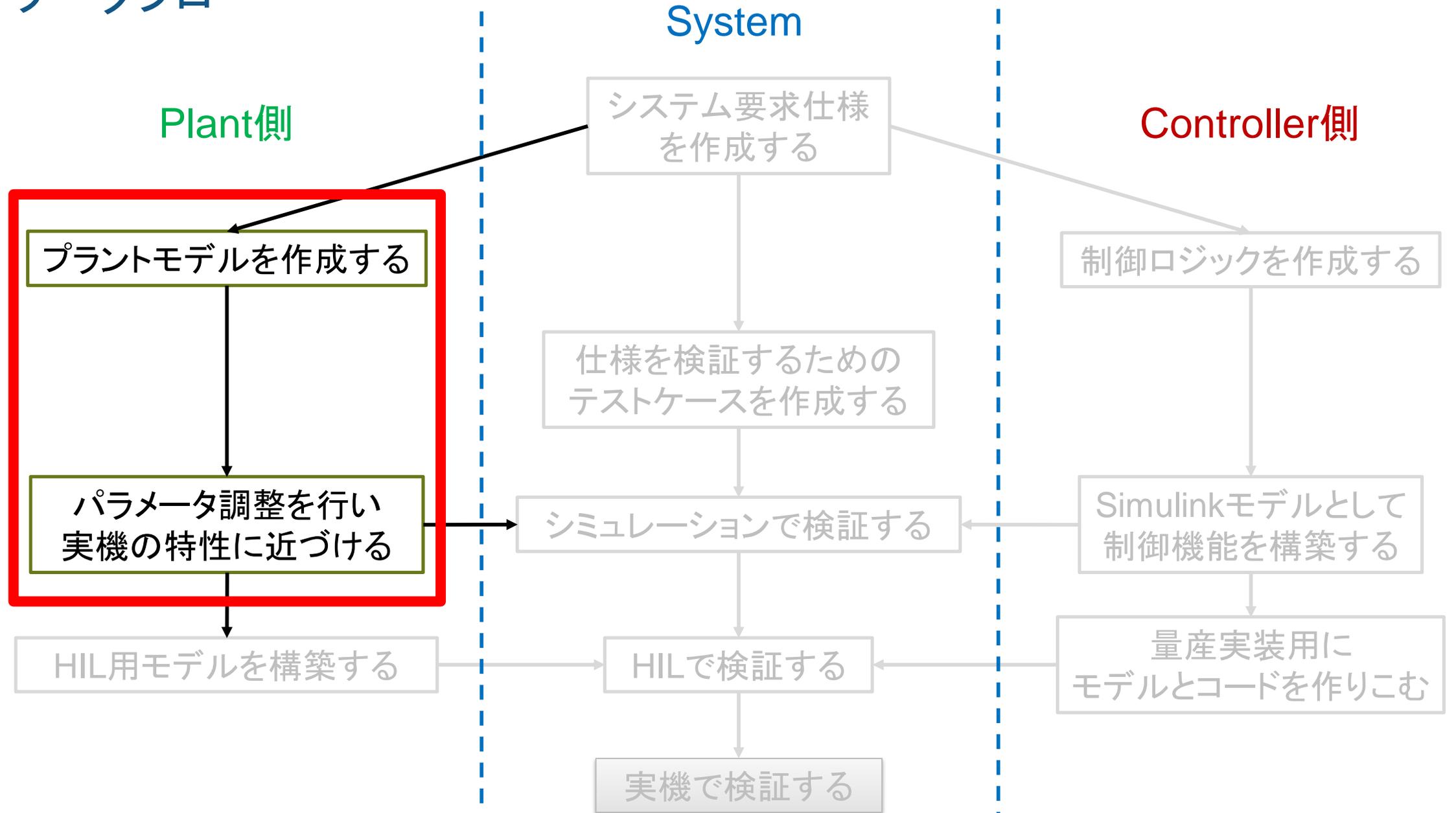
実装モデルと
接続可能

システム要求仕様を作成する 試しにモデルを作ってみる、実機で動かしてみる

すぐに作れる！動かせる！

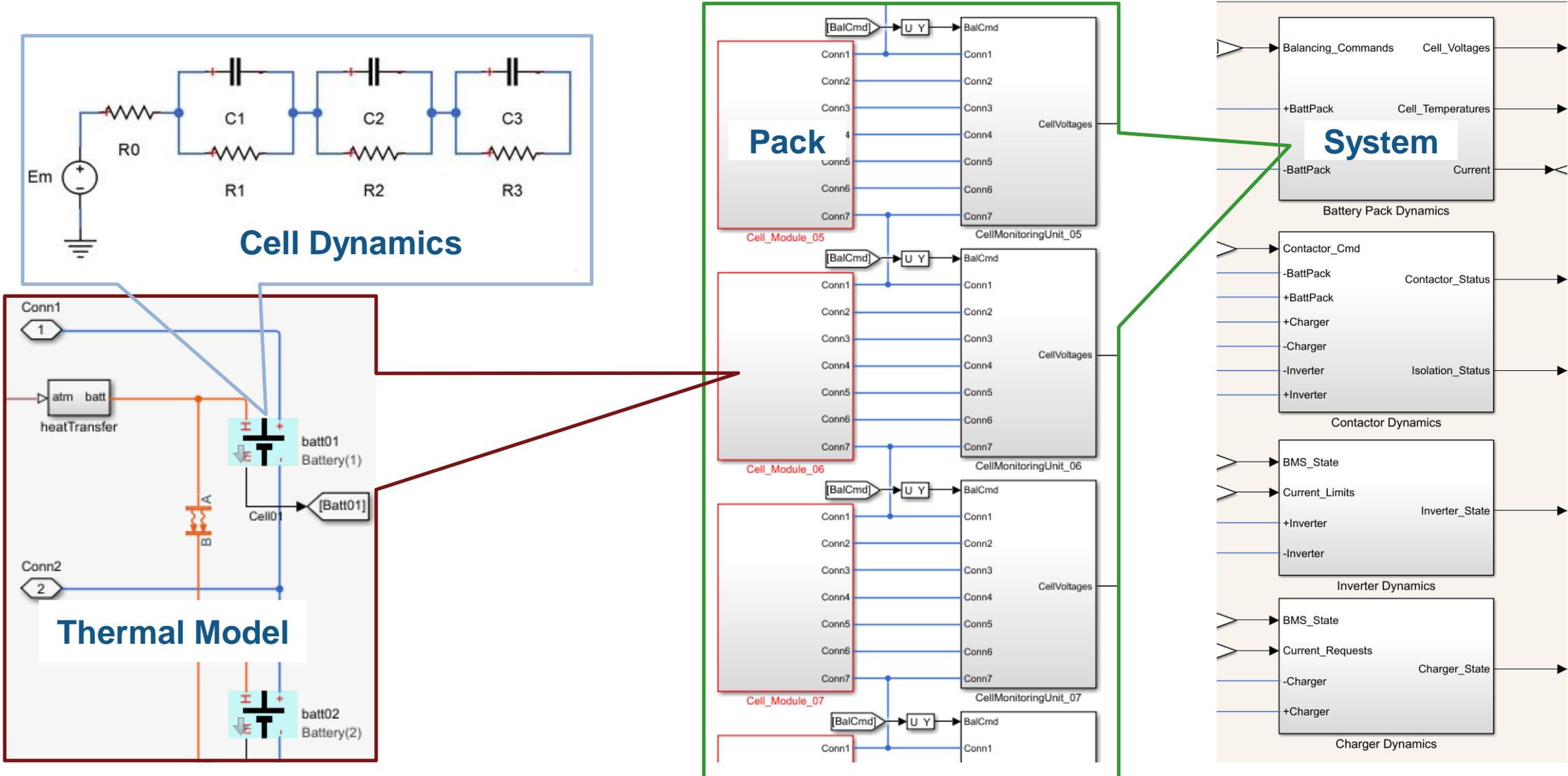


ワークフロー



プラントモデルを作成する

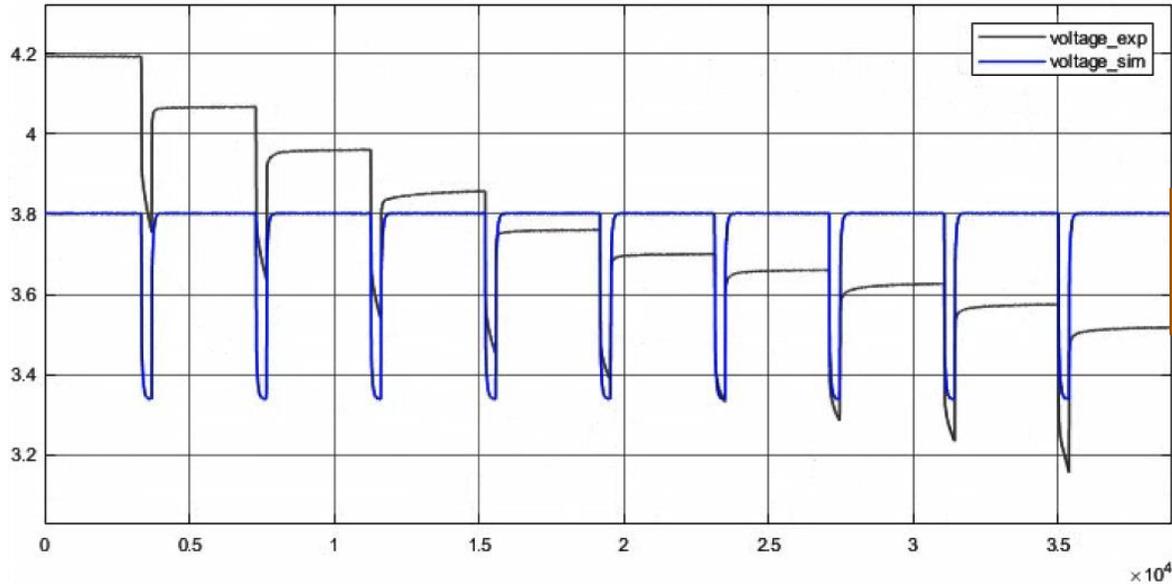
非線形特性も容易に再現、モジュール化をして階層構造化



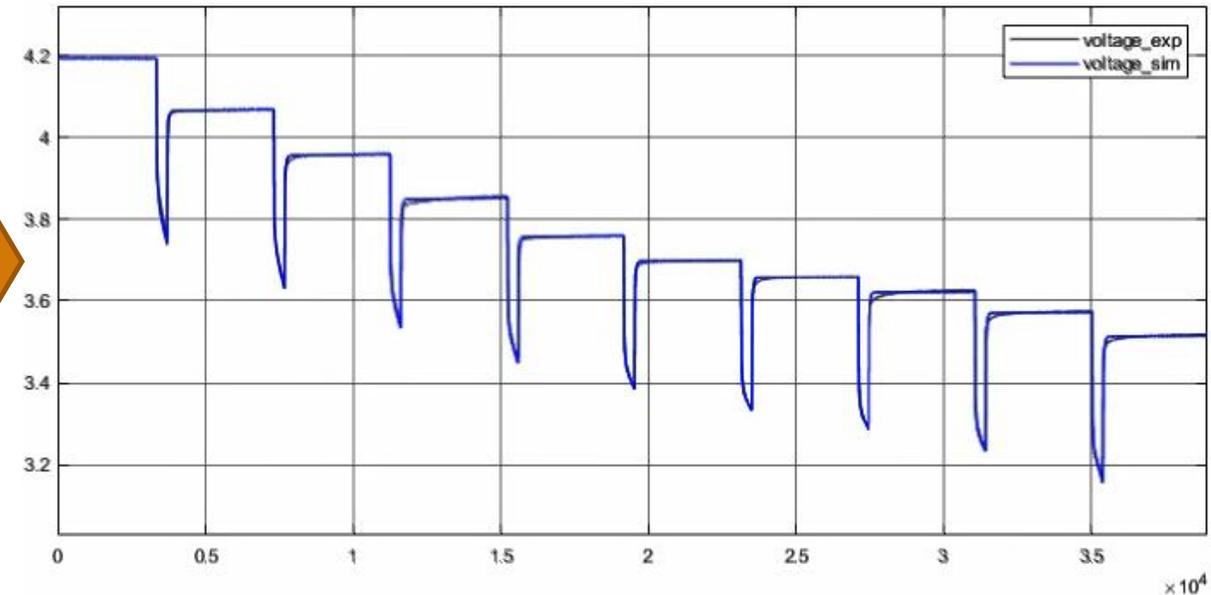
パラメータ調整を行い、実機の特徴に近づける

- 最適化機能を用いて実機データと一致するように調整する

調整前

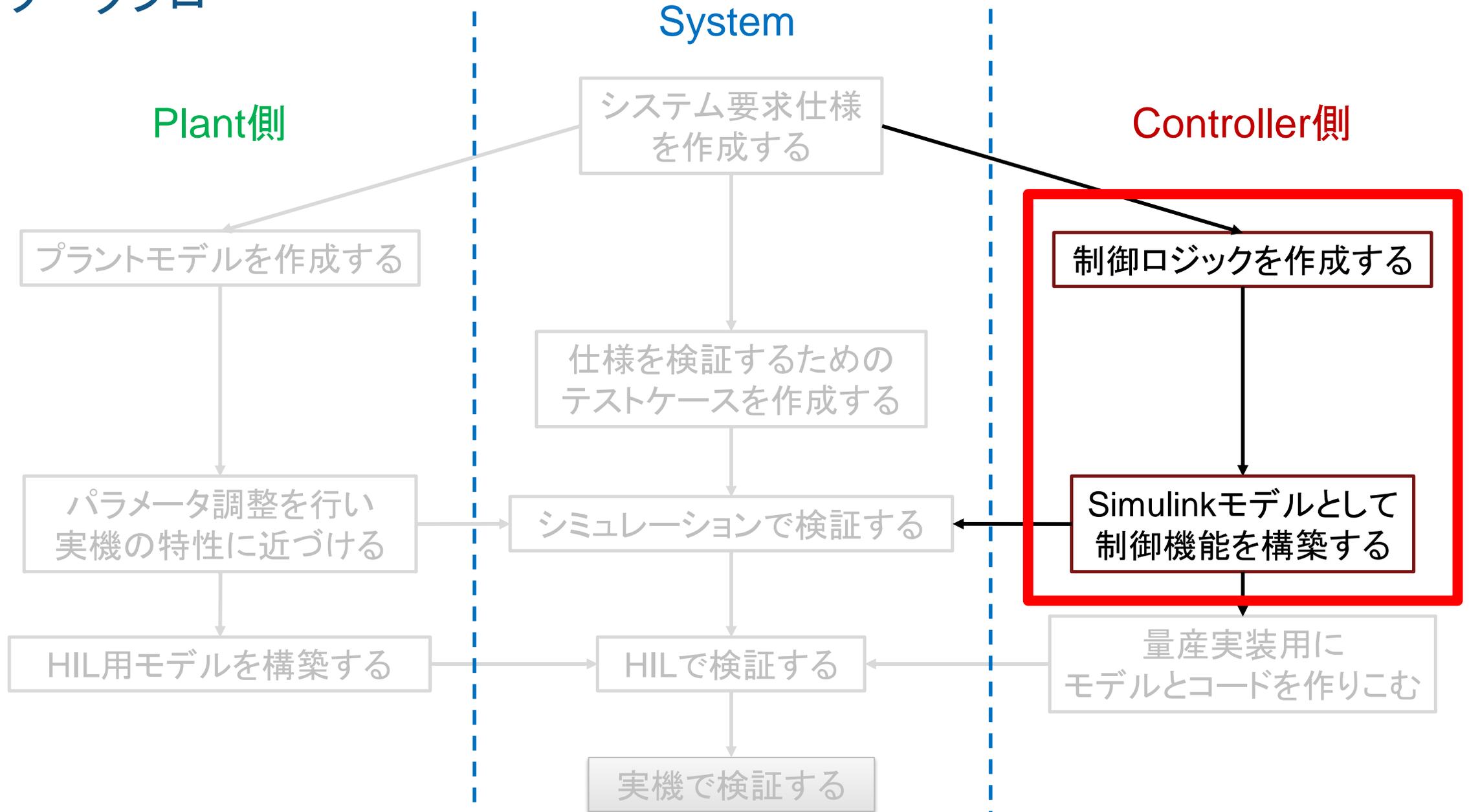


調整後



青線: シミュレーション結果
黒線: 実機の計測結果

ワークフロー

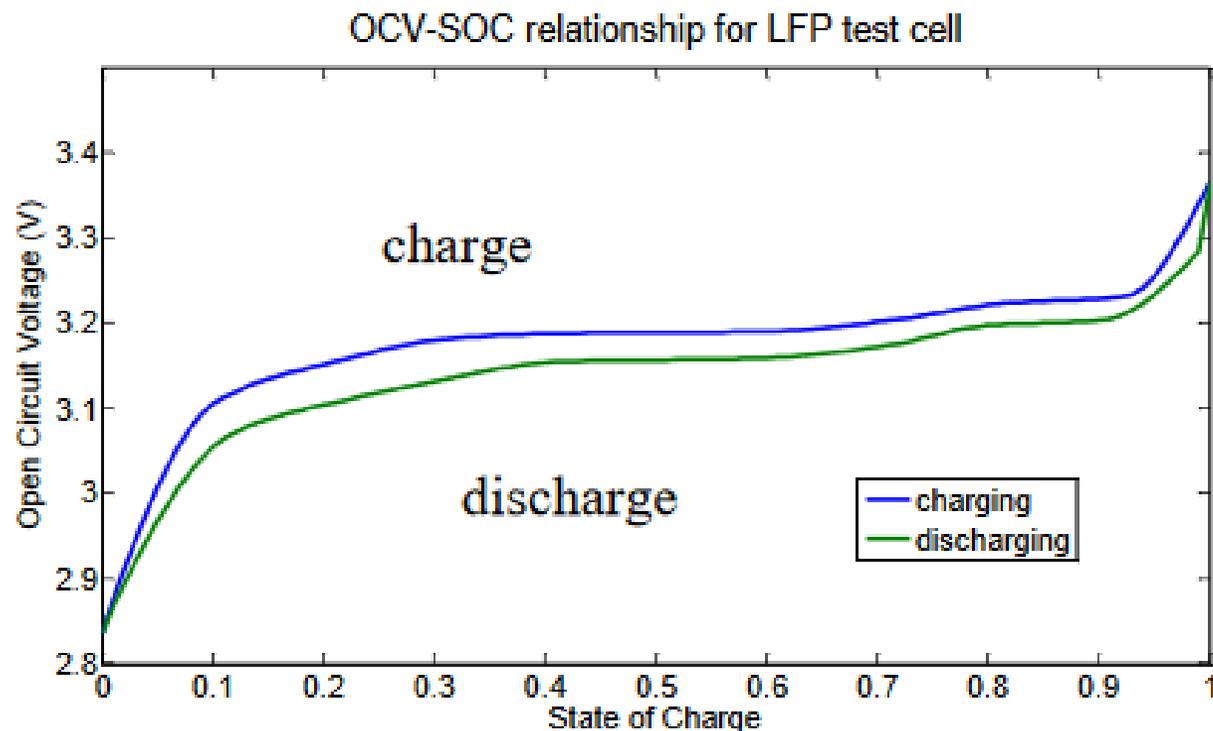


制御ロジックを作成する

(例) SOC推定のためのカルマンフィルタを設計する

$$SOC = \frac{C_r}{C_f} \times 100[\%]$$

C_r は残容量[Ah]、 C_f は満充電容量[Ah]

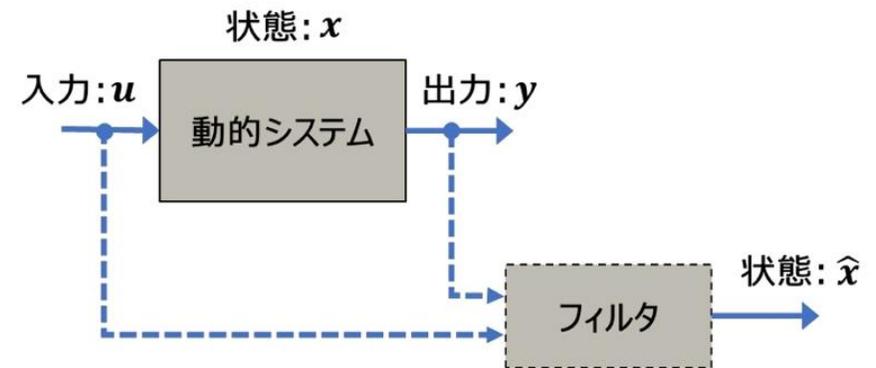
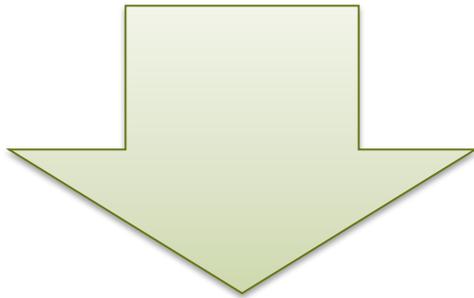


SOCの特性は非線形
かつ
直接測れない

制御ロジックを作成する

(例) SOC推定のためのカルマンフィルタを設計する

- 電圧、電流、温度をリアルタイム計測し、バッテリーモデルを使って精度よく推定したい



- カルマンフィルタによる推定
 - 逐次ベイズフィルタの一種であり、測定データからシステムの状態を実時間で推定するアルゴリズム
 - コンピュータービジョン、誘導・航法システム、計量経済学、信号処理などで広く使われており、多くの実績がある

制御ロジックを作成する

(例) SOC推定のためのカルマンフィルタを設計する

1. 温度とSOCに依存する非線形数式モデルを構築

$$\dot{x} = f(x), \quad y = h(x)$$

2. SimulinkでUKFモデルを構築・実装

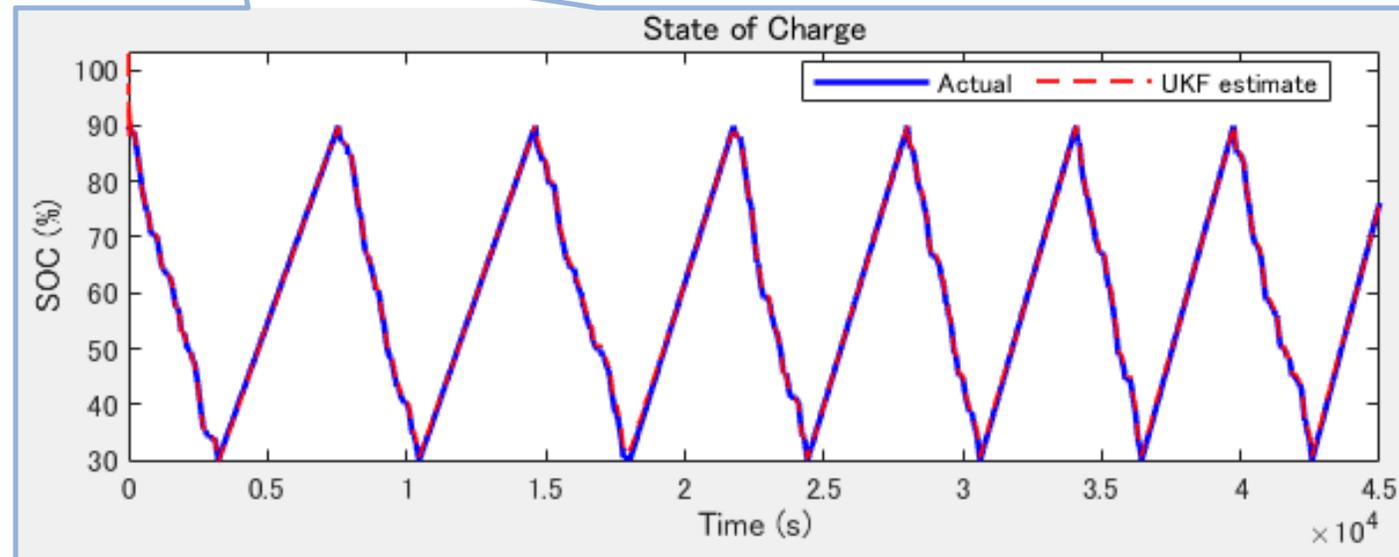
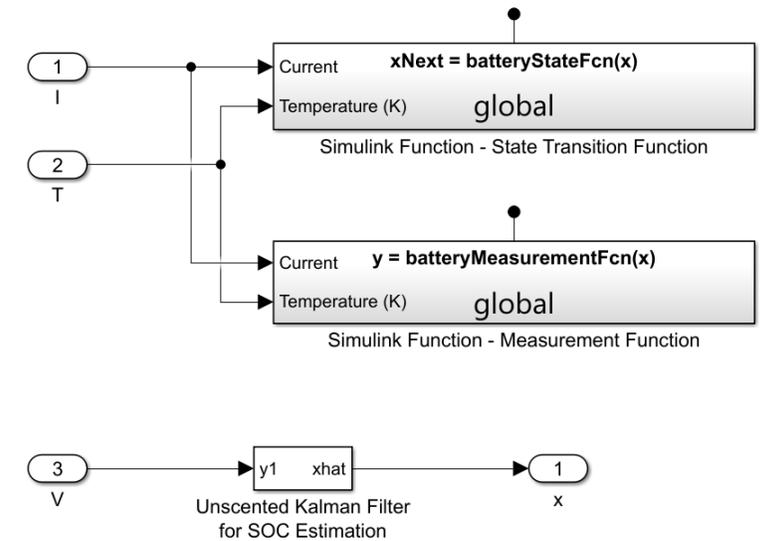
センサー
計測値

Unscented Kalman Filter

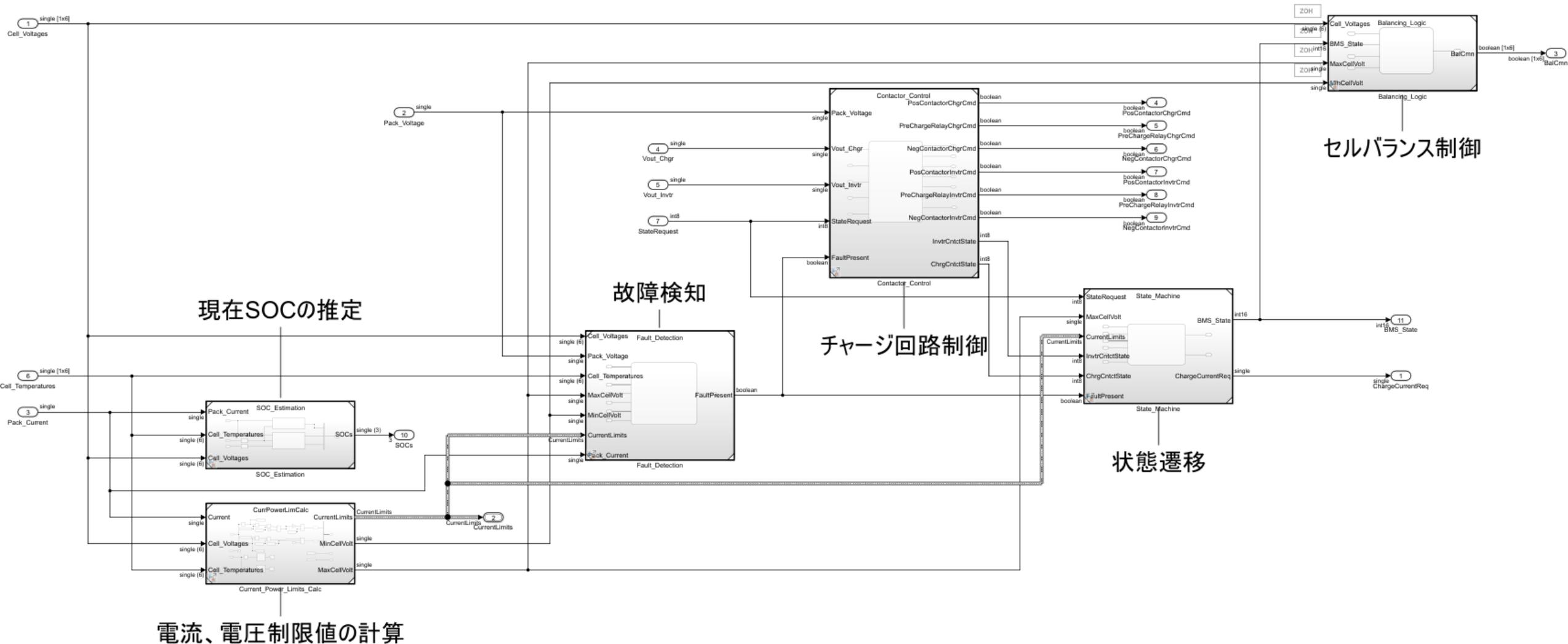
SOC

3. 実測値と予測値を比較しながらチューニング

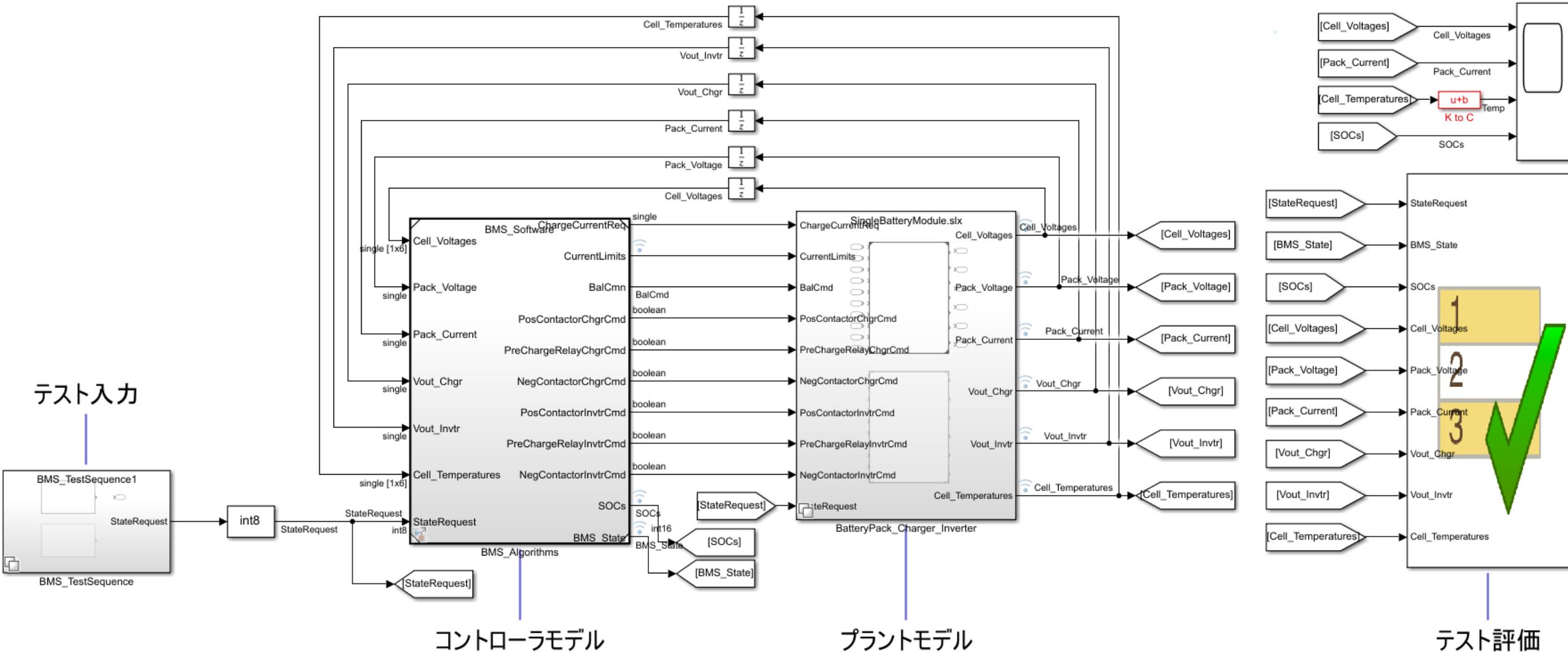
複雑なアルゴリズムも容易に
調整・可視化が可能！



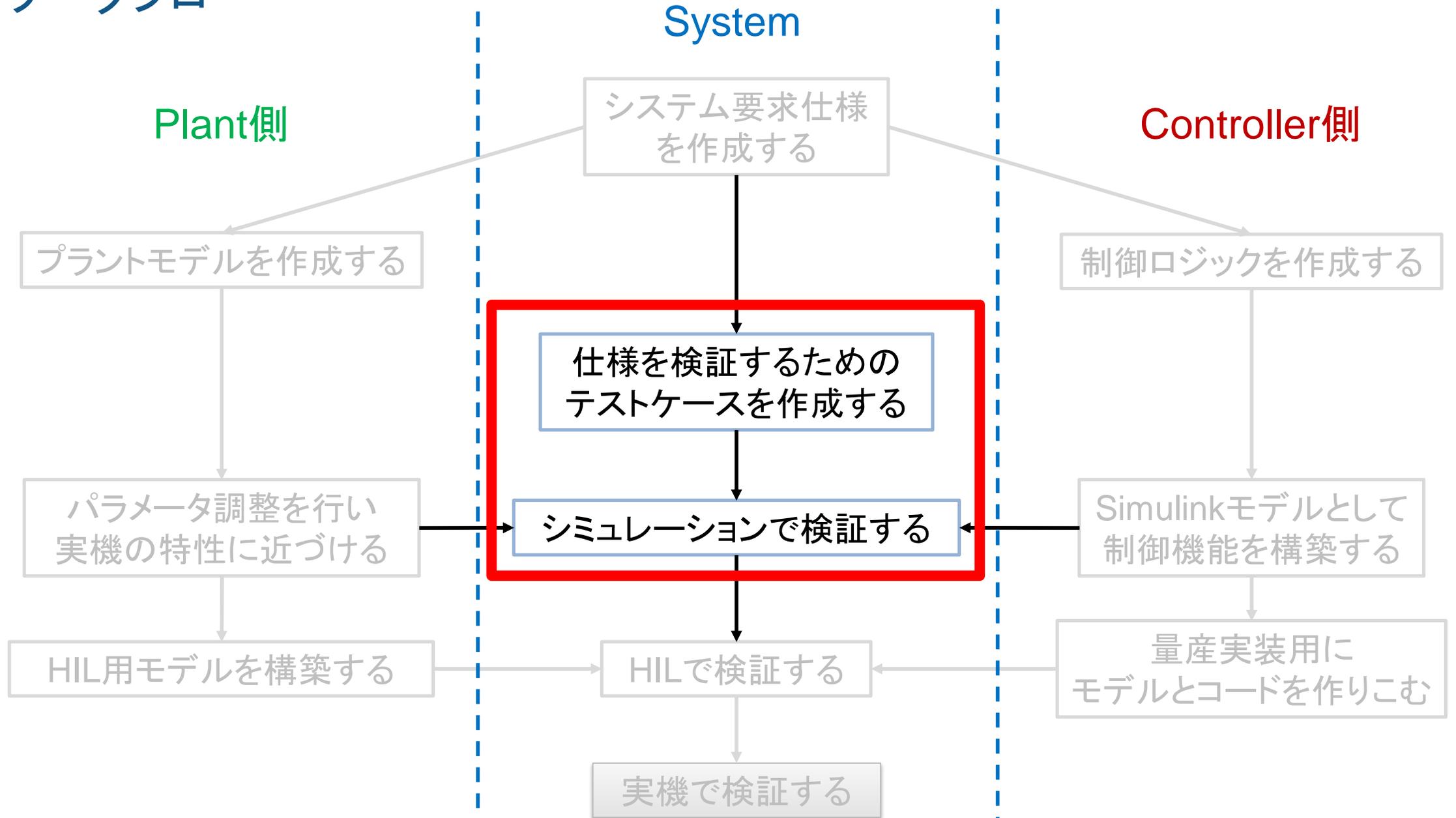
Simulinkモデルとして制御機能を実装する



システムレベルのシミュレーション



ワークフロー



テストケースを作成する

- テスト管理機能で作業の効率化、自動化ができる

テスト入力データ

time	Cell_Voltages(1)	Cell_Voltages(2)	Cell_Voltages(3)	Cell_Voltages(4)	Cell_Voltages(5)
0	3.7	3.7	3.7	3.7	3.7
0.1	3.7	3.7	3.7	3.7	3.7
0.2	3.7	3.7	3.7	3.7	3.7
0.3	3.7	3.7	3.7	3.7	3.7
0.4	3.7	3.7	3.7	3.7	3.7
0.5	3.7	3.7	3.7	3.7	3.7
0.6	3.7	3.7	3.7	3.7	3.7
0.7	3.7	3.7	3.7	3.7	3.7
0.8	3.7	3.7	3.7	3.7	3.7
0.9	3.7	3.7	3.7	3.7	3.7
1	3.7	3.7	3.7	3.7	3.7

テスト マネージャー

テスト

テストブラウザー 結果とアーティファクト

名前またはタグでテストをフィルター (たとえば、tags: ラ

▼ BMS_Test_man

▼ 機能検証

セル過大電圧保護

セル過大電圧保護

BMS_Test_man » 機能検証 » セル過大電圧保護

ベースラインテスト

シミュレーションのリリースを選択: Current

テストケースを外部ファイルから作成

▶ タグ

▶ 説明

▶ 要件

▶ テスト対象システム*

▶ パラメーターのオーバーライド

▶ コールバック

▶ 入力*

▶ シミュレーション出力

▶ コンフィギュレーション設定のオーバーライド

▶ ベースライン基準

▶ 回復

▶ 論理的評価と時間的评价*

▶ カスタム基準

▶ カバレッジの設定

テストケースを分類ごとに管理

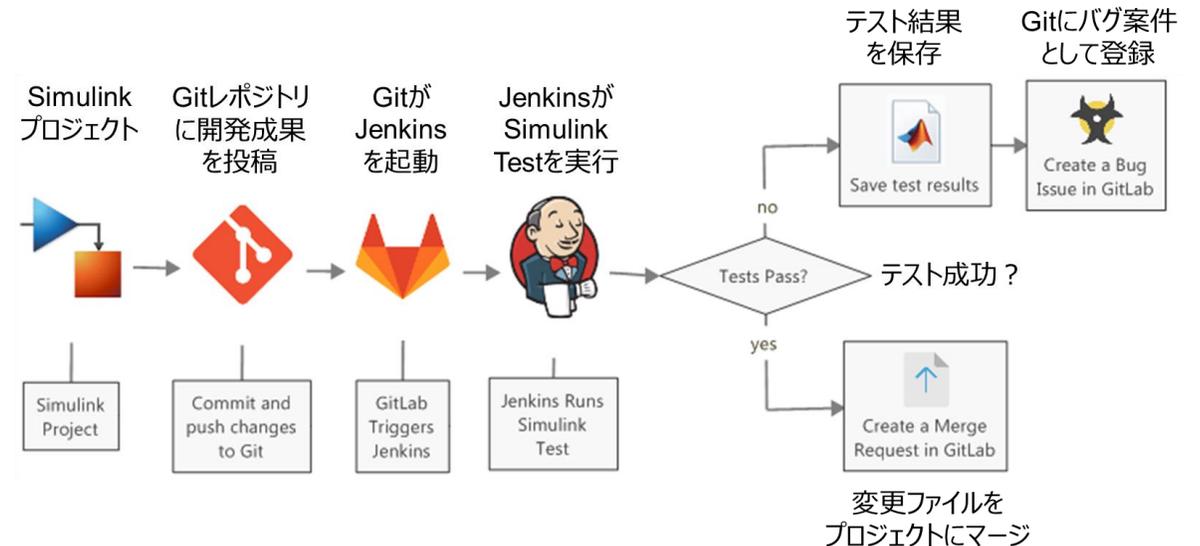
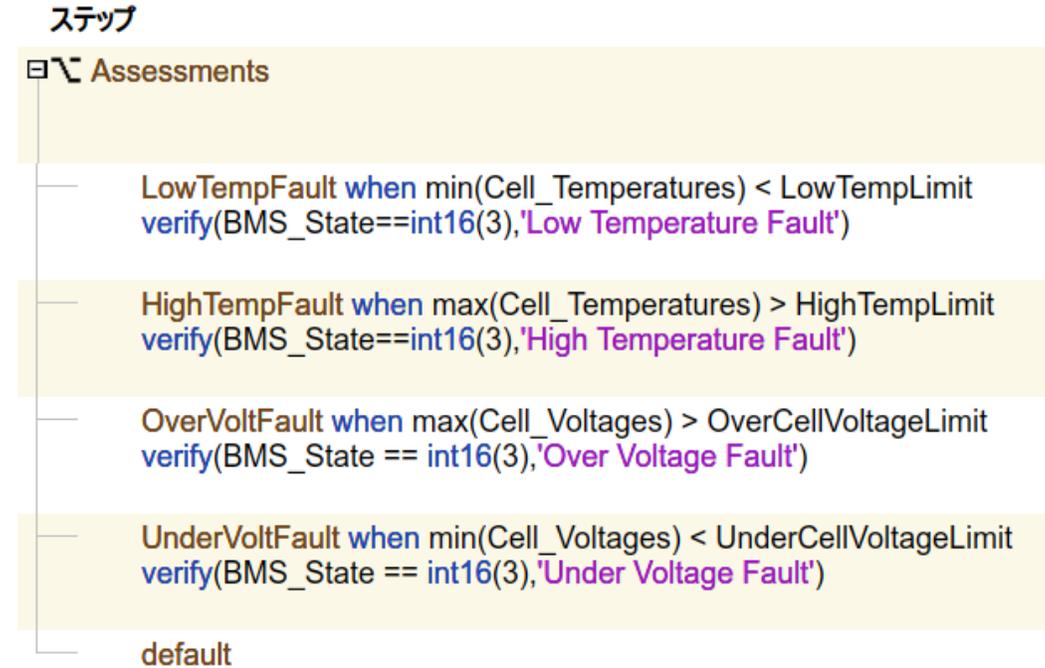
テスト結果の可視化

クリックひとつですべてのテストを実行する

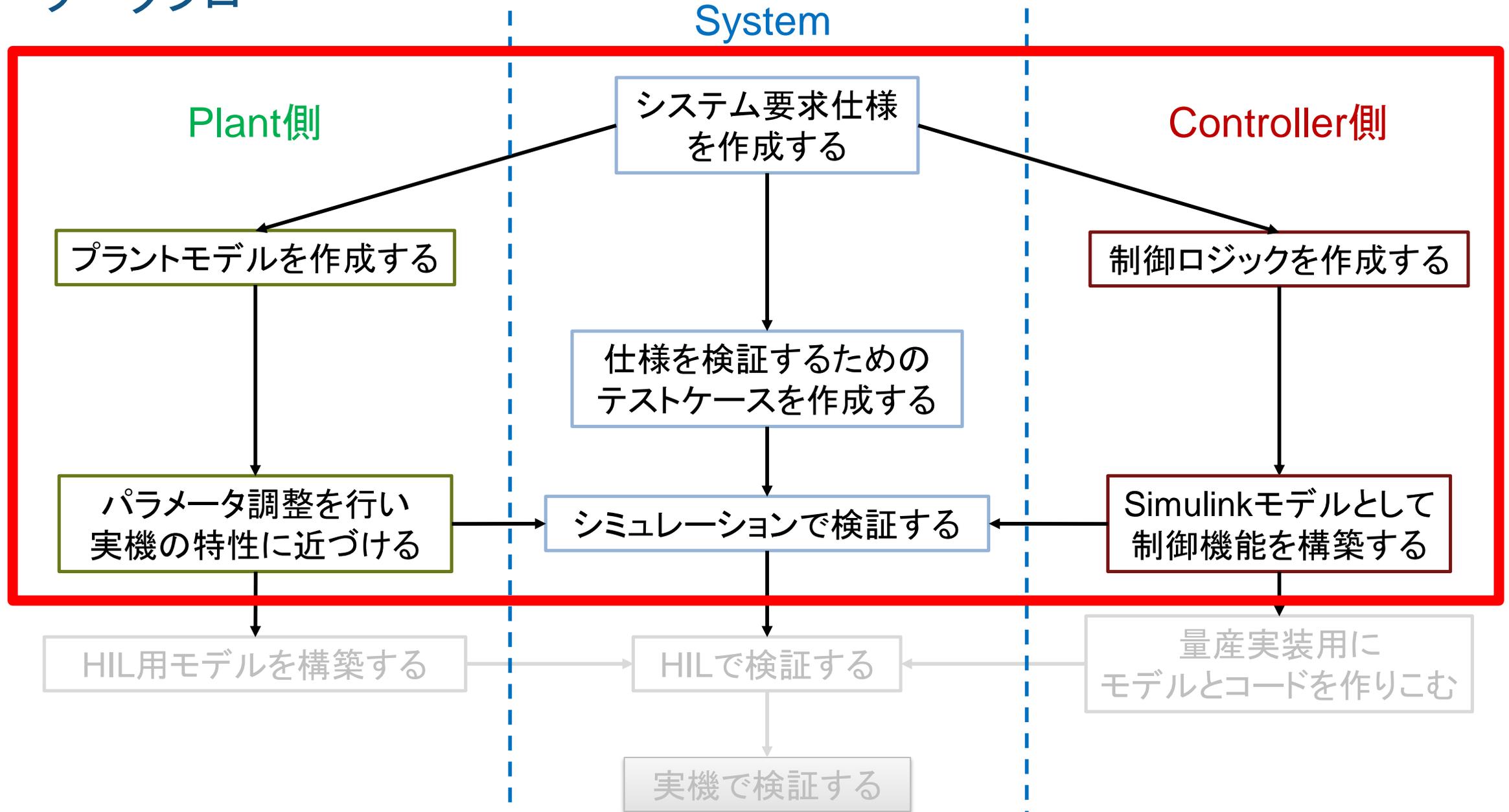
シミュレーション条件
合否判定基準
などの設定

シミュレーションで検証する 結果の判定も自動化！

- 結果の判定方法
 - 期待値との一致性
 - 信号、式の最大最小判定
 - 論理的評価
 - モデル変更前後の一致性
- 可否結果を自動でレポート出力
- CIツール連携



ワークフロー



要求とモデルとテストを繋ぐ 要求のインポートと可視化

仕様書



インポート

Summary: Range

Description Rationale

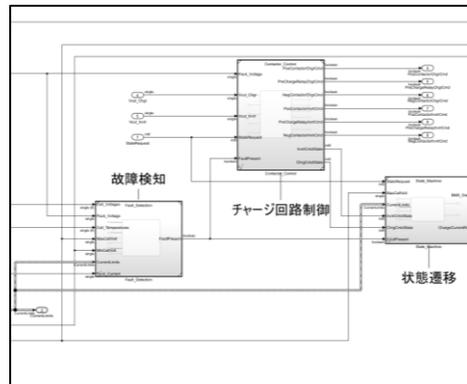
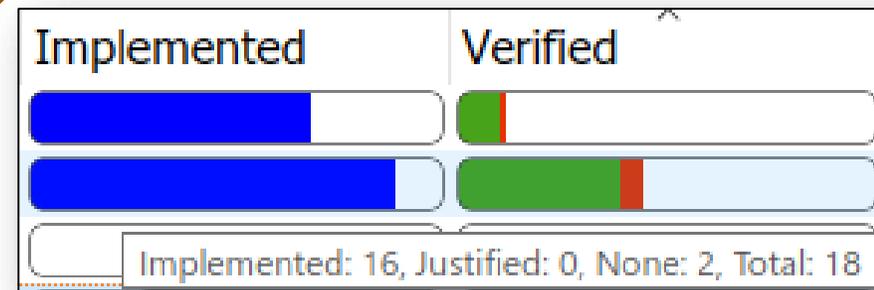
The aircraft shall be controllable for all distances within line-of-sight



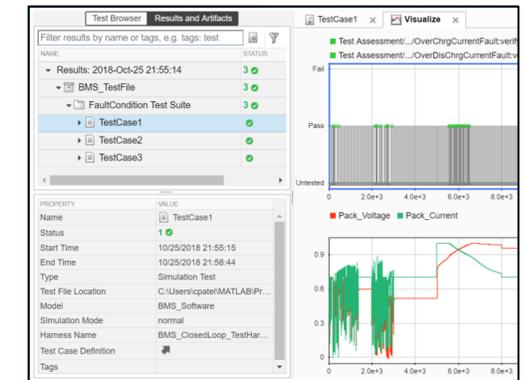
要件エディター

要件の項目に対するテストケースを
多対多で関連付け

要件の項目に対するモデルを
多対多で関連付け



Simulinkモデル



テストケース

サブシステムごとにテストケース作成

要求とモデルとテストを繋ぐ 仕様変更があった場合

仕様書



変更点
インポート

Summary: Range

Description Rationale

The aircraft shall be controllable for all distances within line-of-sight



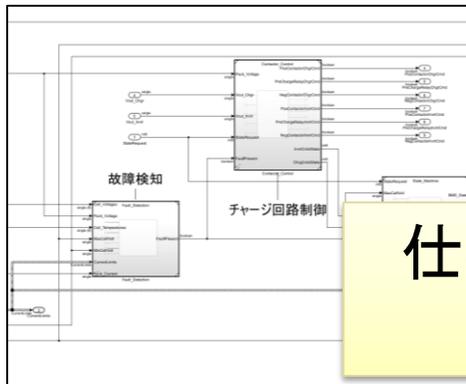
要件エディター

該当する仕様変更箇所と
関連付けられているモデルを表示

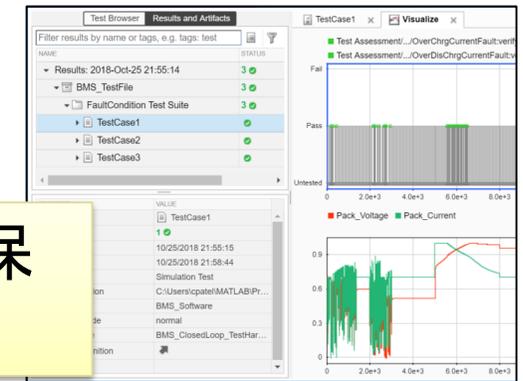
該当する仕様変更箇所と
関連付けられているテストケースを表示

修正
完了報告

修正&検証
完了報告



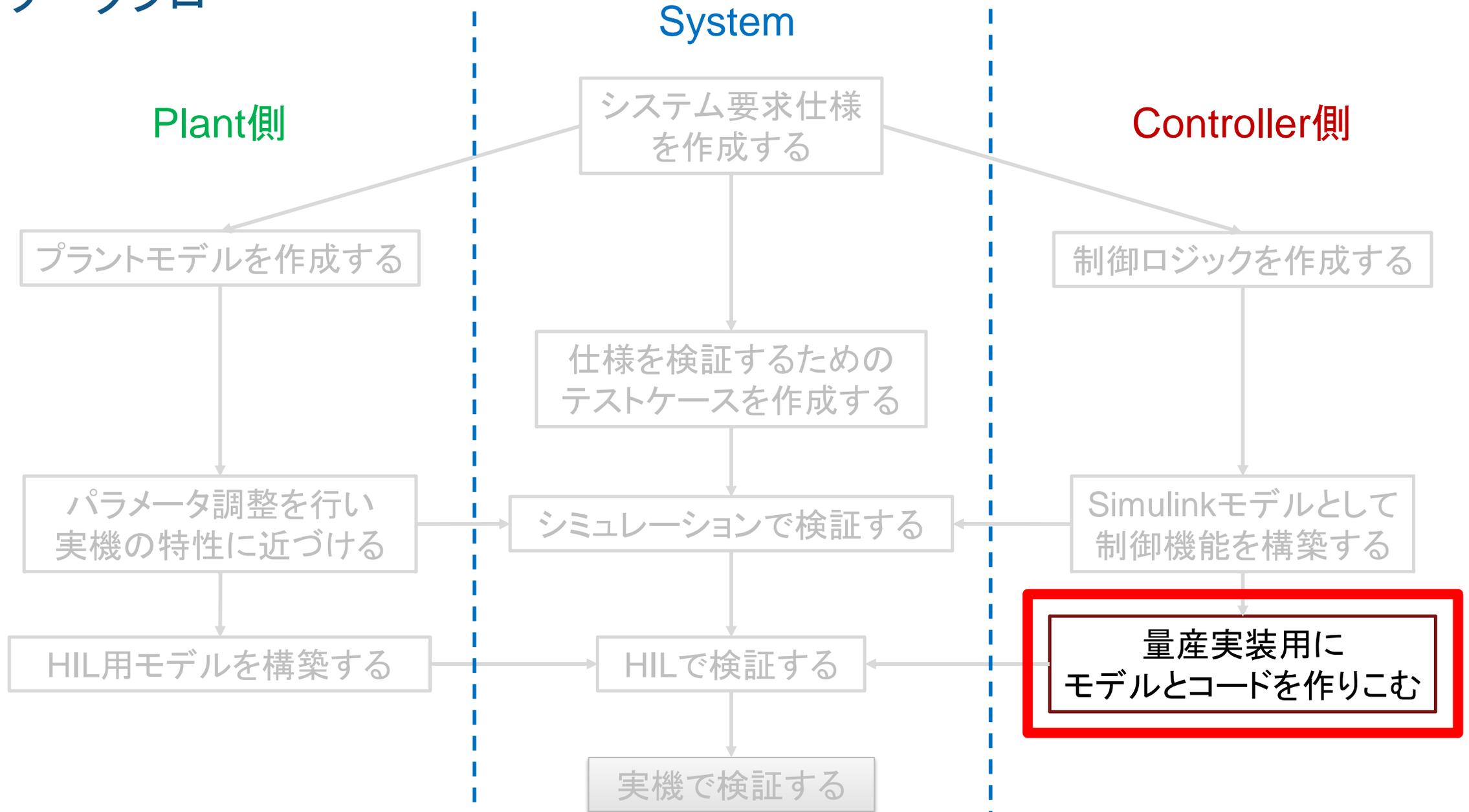
Simulinkモデル



テストケース

仕様の「抜け漏れ」をなくし、トレーサビリティを確保
自動化と可視化により修正作業の効率アップ

ワークフロー



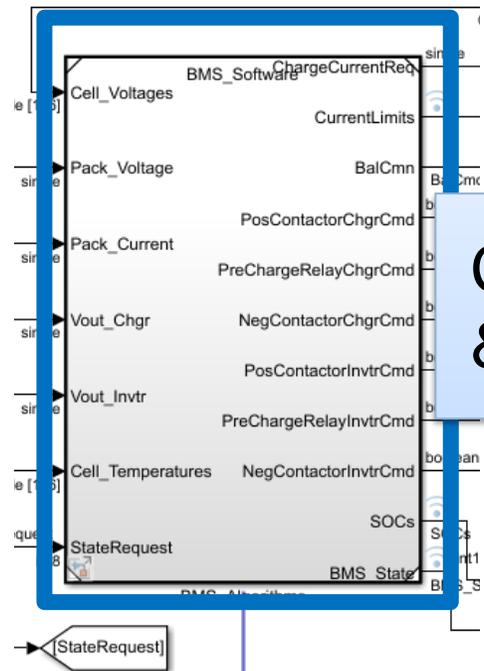
量産実装用にモデルとコードを作りこむ

- モデリング基準に対する準拠性確認
- 条件分岐や判断に関するカバレッジ測定
- 量産コード生成

量産実装用にモデルとコードを作りこむ

- モデリング基準に対する準拠性確認
- 条件分岐や判断に関するカバレッジ測定
- 量産コード生成

量産コード生成



コントローラモデル

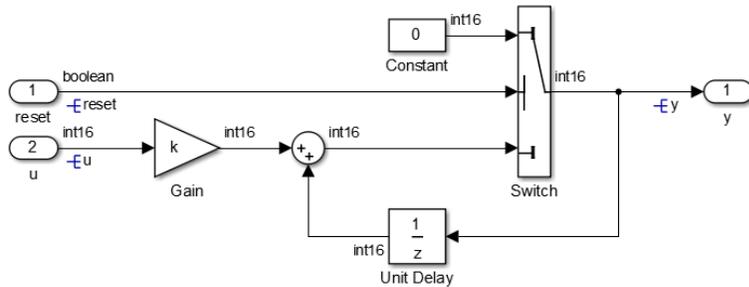
C/C++コード生成
& ビルド



Embedded Coder® :

モデル/M-ファイルから量産/組込み用C/C++コードを自動生成

モデル (ロジック)



コード
自動生成

```
void rst_cntr_step(void)
{
  if (reset) {
    y = 0;
  } else {
    y += (int16_T)(k * u);
  }
}
```



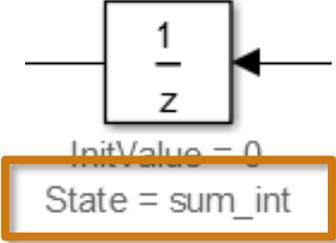
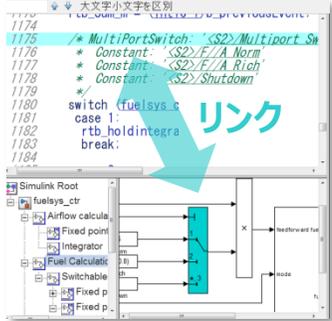
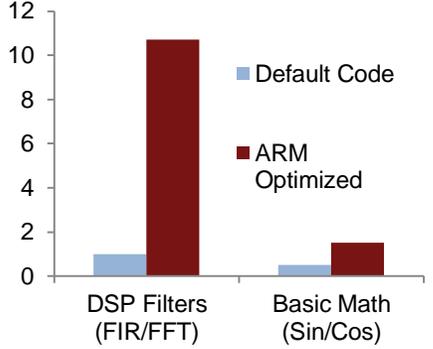
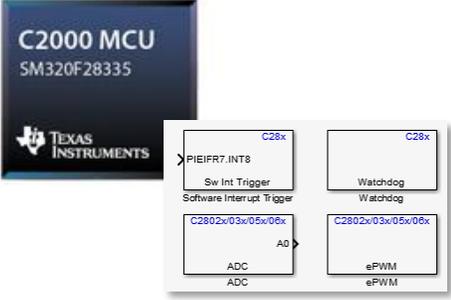
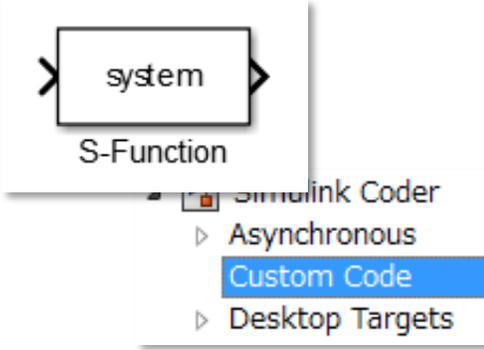
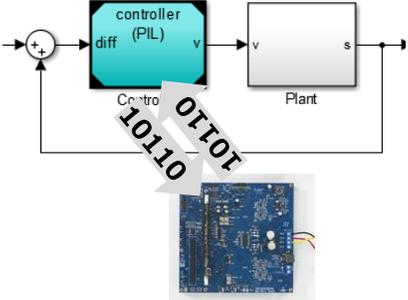
ソフト仕様



- 関数
 - 名前、引数
- 変数/定数属性
 - 名前、記憶クラス、データ型
- コード書式

- コーディングレス
ただし、モデリングやコード生成準備作業は必要
- モデルの誤解釈混入リスクを解消
- モデル・コード間の乖離を防止
- ソフト屋さん以外の開発者も記述・理解しやすい

Embedded Coder 主な機能

効率的なCコード生成	変数・定数設定	モデル・生成コード間リンク	最適化・カスタマイズ									
<p>量産・組込用途に最適です</p> <pre>if (reset) { y = 0; } else { y += k * u; }</pre> 	<p>利便性の高いコードを生成します</p>  <pre>static s16 sum_int;</pre>	<p>レビュー・トレーサビリティに便利</p> 	<p>高速化・内製ルール対応に有効</p>  <table border="1"> <caption>Performance Comparison</caption> <thead> <tr> <th>Category</th> <th>Default Code</th> <th>ARM Optimized</th> </tr> </thead> <tbody> <tr> <td>DSP Filters (FIR/FFT)</td> <td>~1.0</td> <td>~11.0</td> </tr> <tr> <td>Basic Math (Sin/Cos)</td> <td>~0.5</td> <td>~1.5</td> </tr> </tbody> </table>	Category	Default Code	ARM Optimized	DSP Filters (FIR/FFT)	~1.0	~11.0	Basic Math (Sin/Cos)	~0.5	~1.5
Category	Default Code	ARM Optimized										
DSP Filters (FIR/FFT)	~1.0	~11.0										
Basic Math (Sin/Cos)	~0.5	~1.5										
MCU/DSP専用ブロック	外部Cコード取り込み	モデル・コード等価性検証	各種規格に対応									
<p>実験・試作に便利です</p>  <p>※一部MCU/DSPで利用可能</p>	<p>既存ソフト資産の活用が可能</p> 	<p>生成コードの品質保証に貢献</p>  <p>SIL: Software In the Loop PIL: Processor In the Loop</p>	<p>多様なニーズに対応しています</p> 									

量産コード生成 ～コードの性能～

Delphi社HVモータ制御ソフト



Task / Module	Throughput (uSec)	
	Model	Hand-Code
Current Magnitude and Phase Process	1.42	1.31
ABC to dq0 Frame Transformation	0.76	0.52
Resolver Harmonic Learn	0.48	0.22
Angle Position Determination	0.93	0.84
PI-Current Regulator	7.62	7.51
Torque Mode	4.82	4.72
dq0 Rotating to Stationary Frame Transformation	0.94	0.82
Complete 100 uSec Task	65.37	63.83

※MathWorks Automotive Conference Michigan 2015

Visteon社パワートレイン制御ソフト

		Code Size
Hand Code		928
Auto Code	No overflow/underflow check	904
	Check OF/UF everywhere	1562
	Check only where necessary	934

*Based on Tasking Compiler for ST10

Table 2 ROM and RAM comparison between a floating-point hand code and auto code.

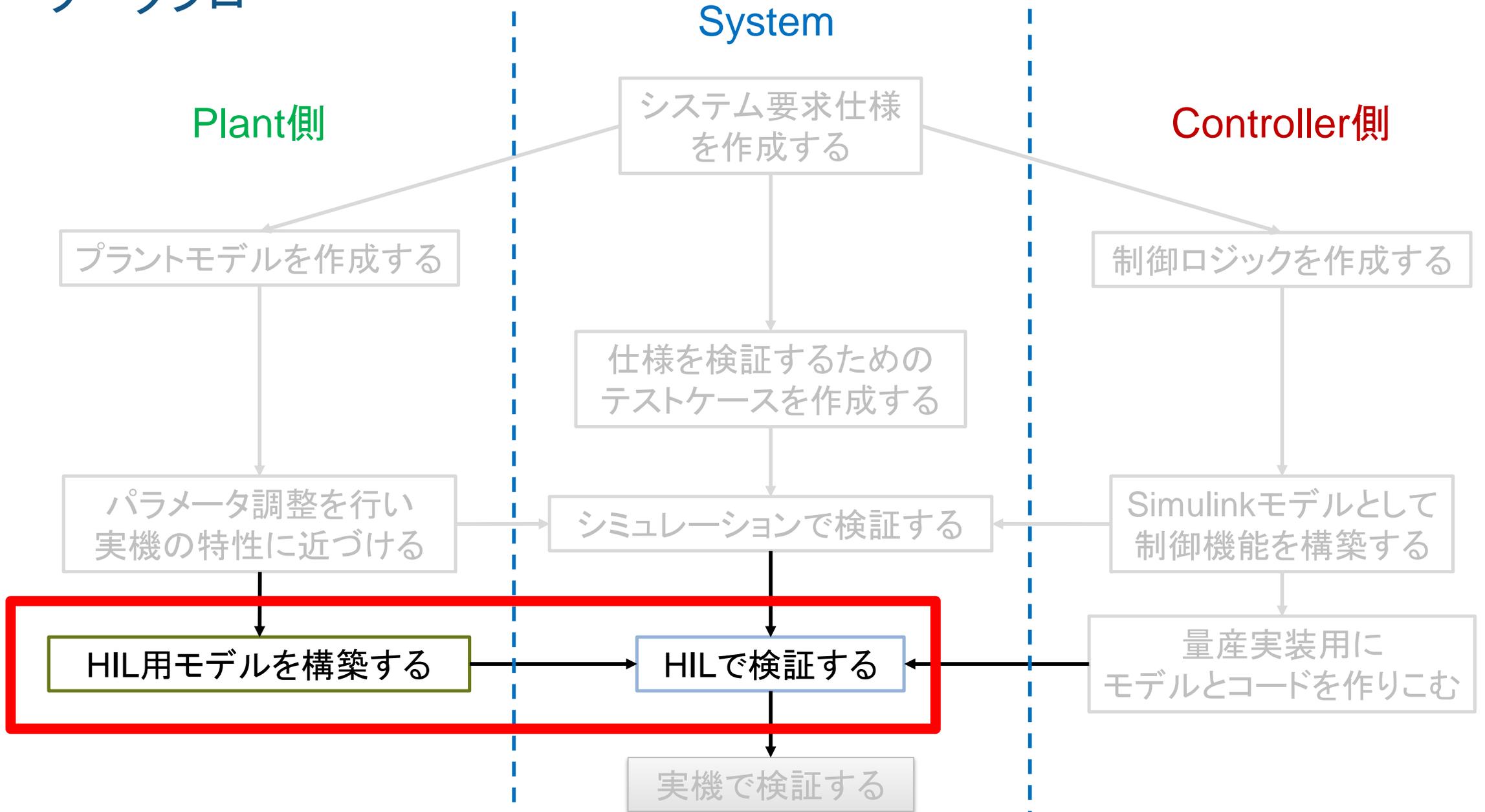
	Hand Code	Auto Code
ROM	6408	6192
RAM	132	112

※SAE Technical Paper 2004-01-0269, March 2004

ハンドコードと遜色ないCコードを生成可能

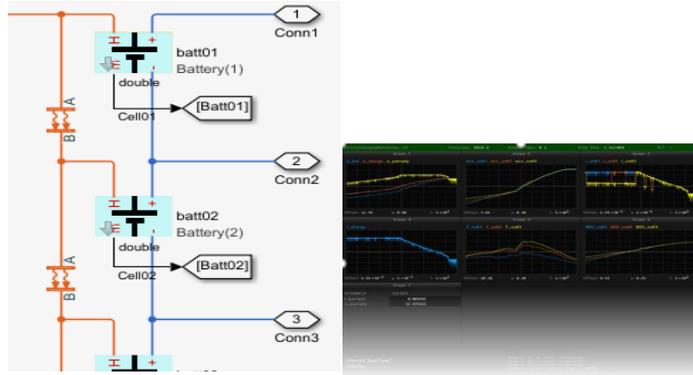
- 一般論として、最適化されたハンドコードと比べて計算速度・メモリ効率性は劣る可能性があります
- 高効率コード生成にはツール知識やモデリング上の工夫が必要

ワークフロー



HIL用モデルを構築し、検証する

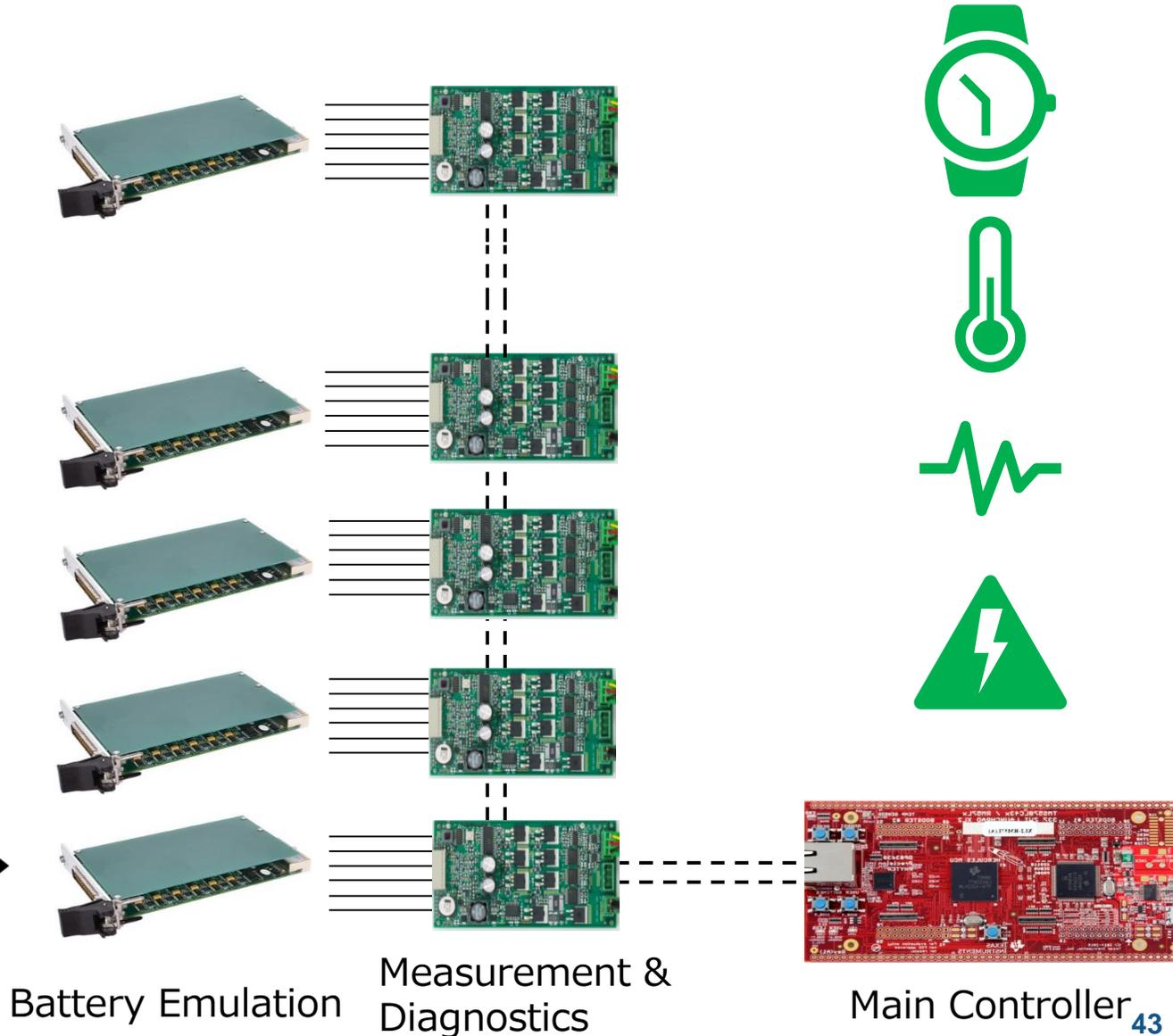
- ✓ 任意の試験条件の素早い再現
- ✓ 実機では危険な試験を安全に実施
- ✓ 24時間自動テスト可能



自動コード生成



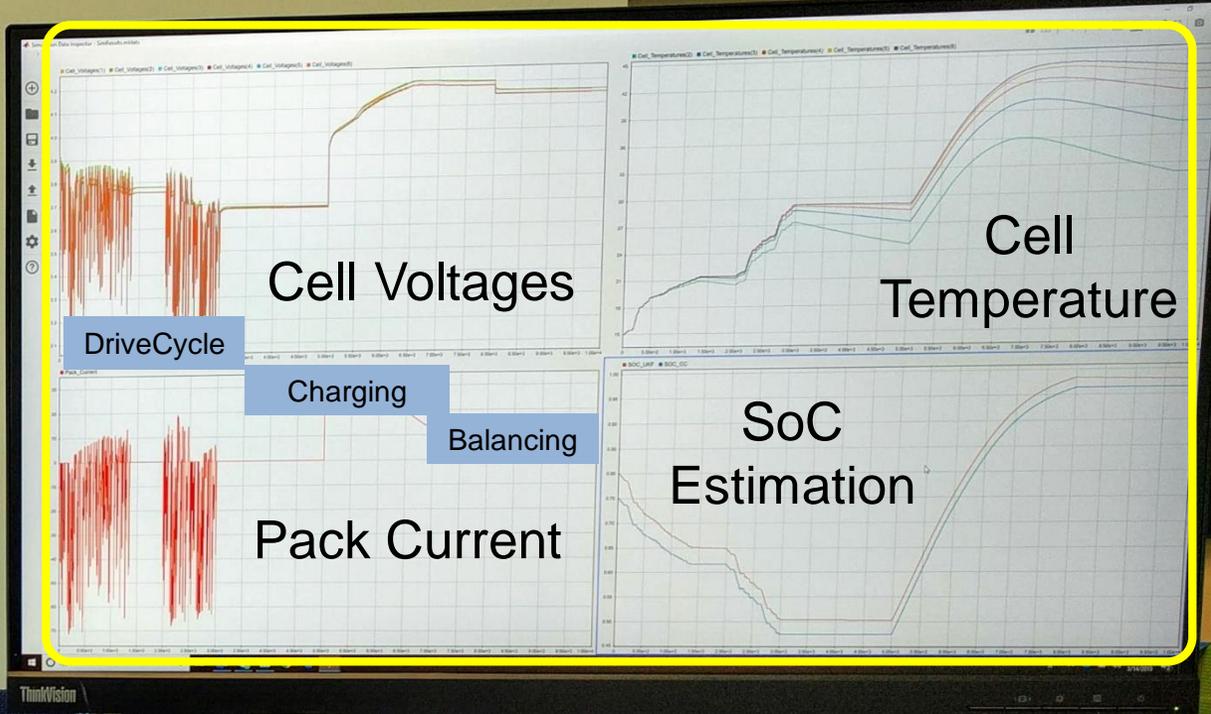
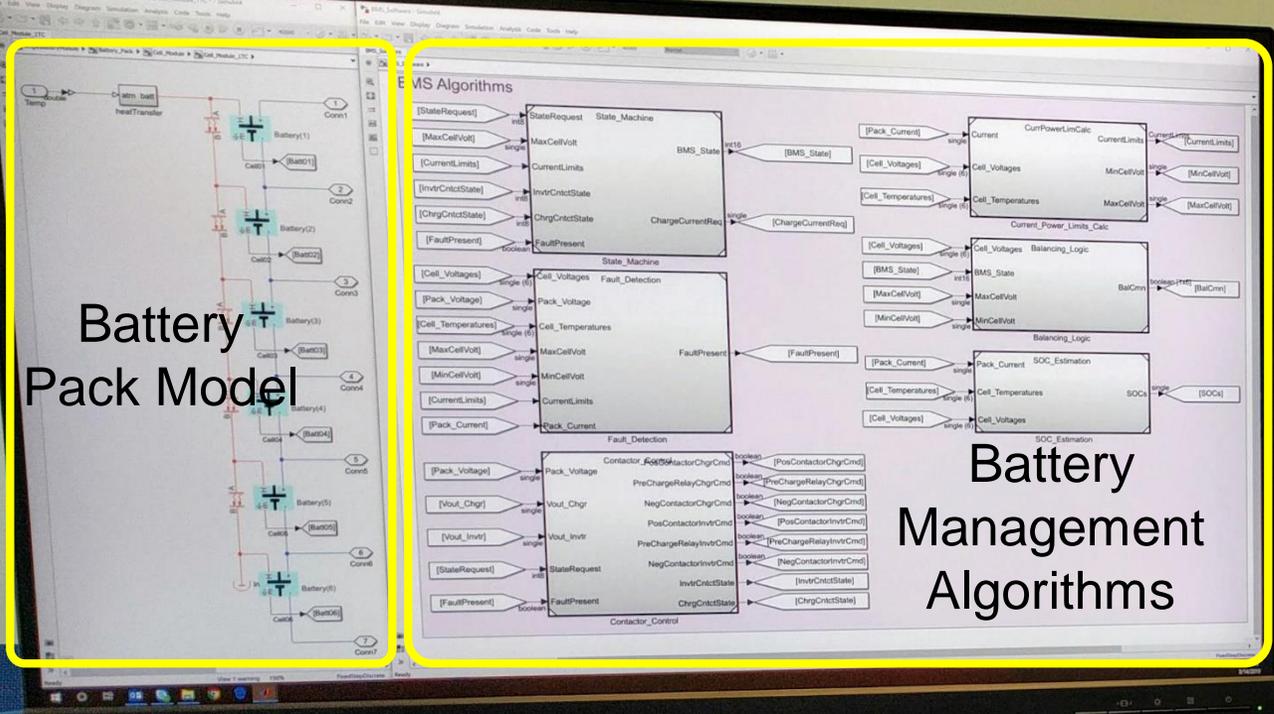
1001011100101
Wiring and Signal Conditioning



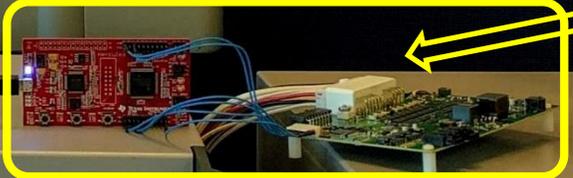
Battery Emulation

Measurement & Diagnostics

Main Controller ⁴³



Simulink Host Computer

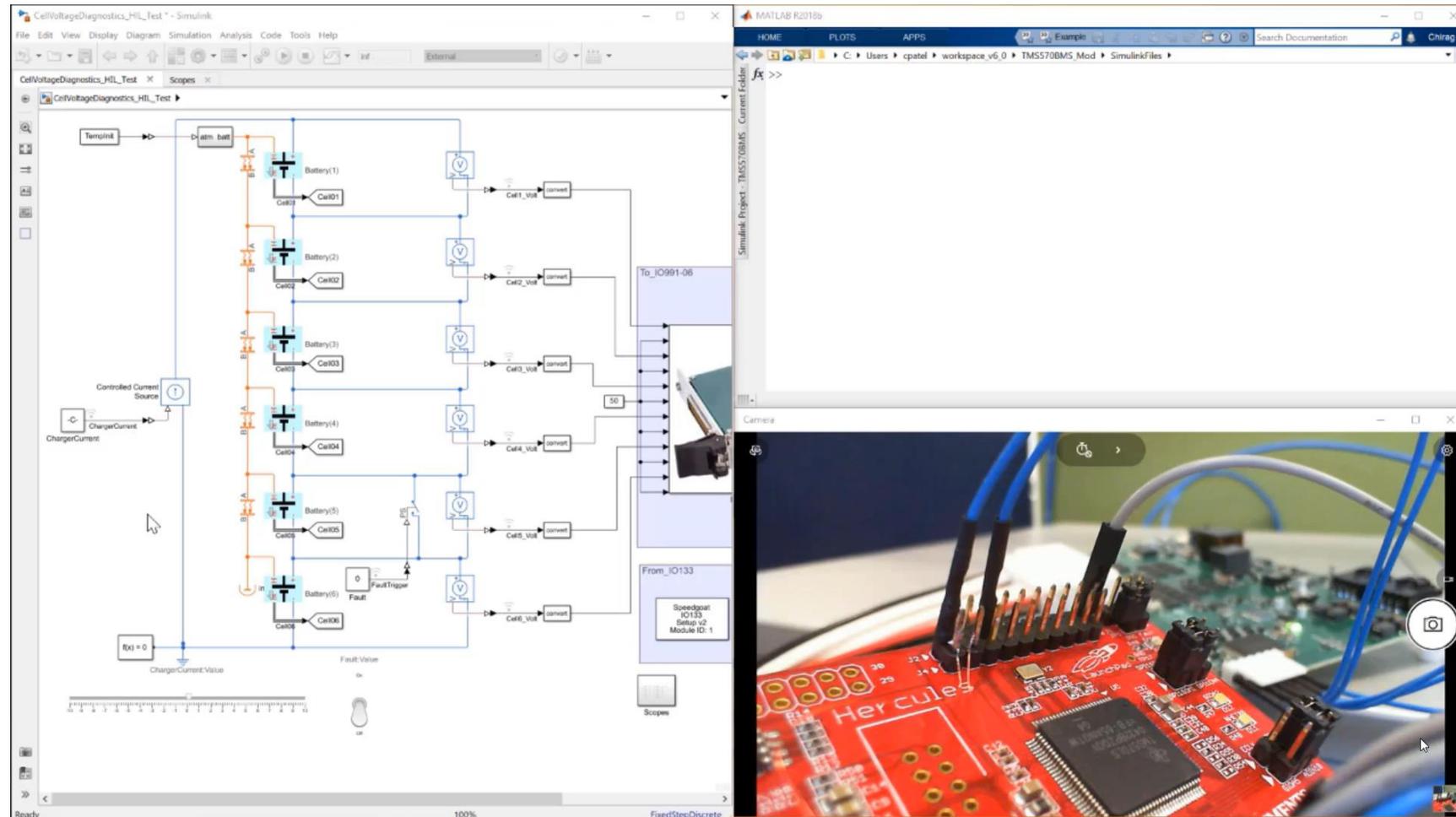


Speedgoat Target Computer with Battery Emulation Card – IO991-06

performance real-time target machine

speedgoat

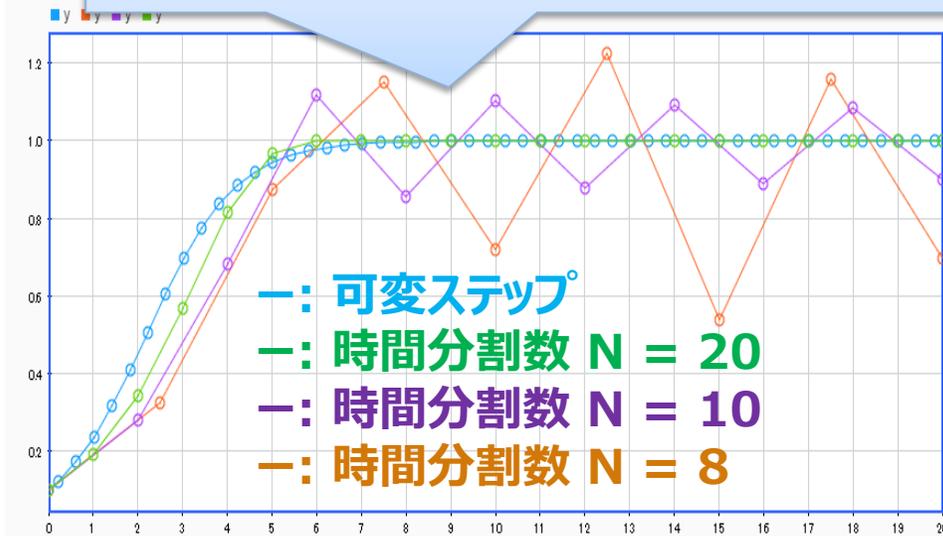
BMS-HILシミュレータ実行の様子



HILシミュレータでのプラントモデルのシミュレーション

- オンラインシミュレーション: 固定ステップソルバーで演算 (コード生成された形で実装)
 - 一定のタイムステップでモデルを実行
 - ステップサイズと精度はトレードオフの関係

演算精度と安定性を保てるステップサイズの検討



演算精度を確保するためのタイムステップ



HILシミュレータ

ユーザーが指定できるステップサイズ

設定可能なステップサイズ(CPUのオーバーラン限界)

プラントモデルの実行時間

モデルのシミュレーション
実行

入出力
処理/
その他

アイドル

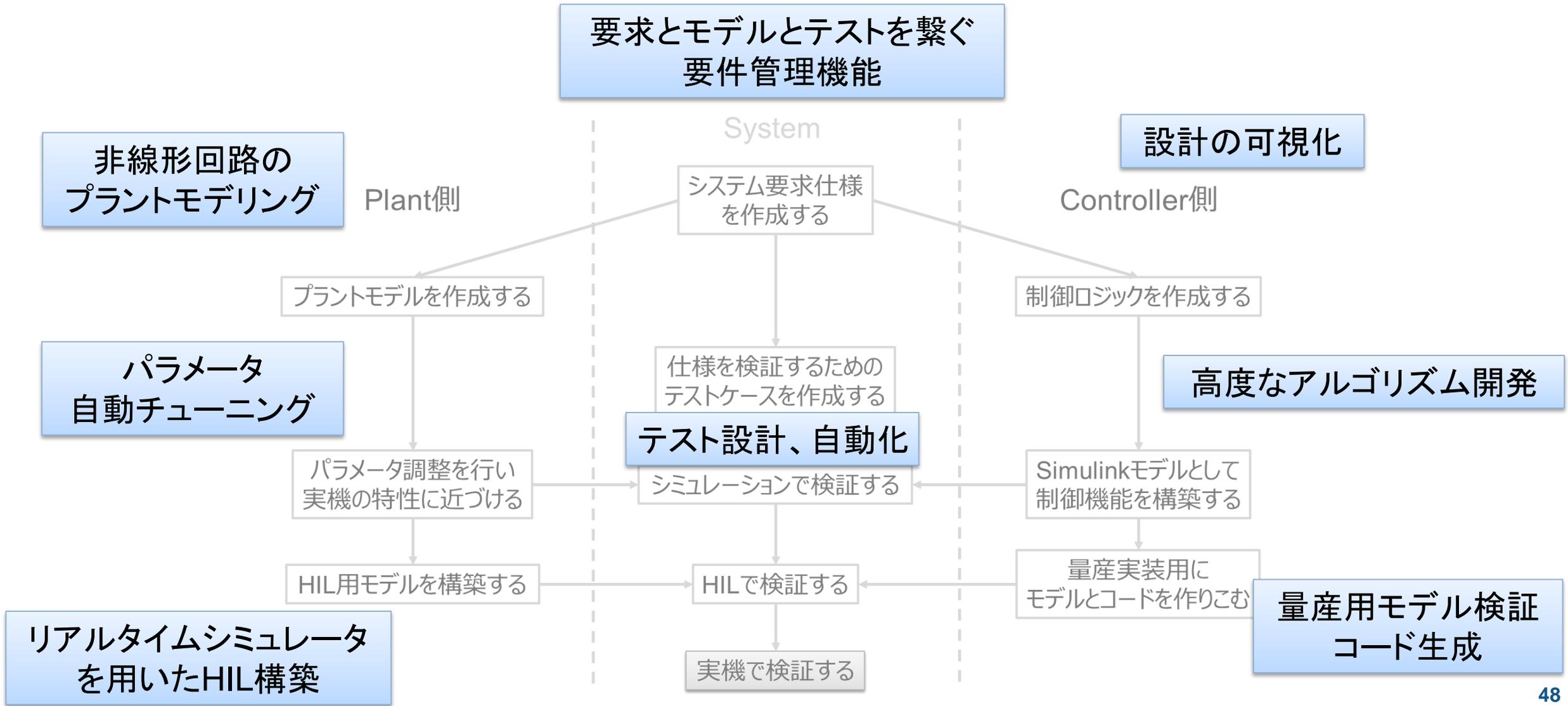
ステップサイズ

HILシミュレータの演算性能の制約

モデルの忠実度とリアルタイム性のバランスの検討が必要

まとめ

各ワークフローでプロジェクトを支えるMATLAB/Simulink



ツール一覧

分類	用途	機能	製品
机上検証 (シミュレーション)	組込み制御システムの モデリング・シミュレーション	数値解析、グラフ、プログラミング、GUI	MATLAB®
		ブロック線図モデリング	Simulink®
		フローチャート・状態遷移図/表モデリング	Stateflow®
	プラントモデルの作成	複合物理領域の物理モデリング	Simscape™
		電気系モデリング(パワエレ/電力系統/デジアナ/センサ)	Simscape Electrical™
		自動車のパワートレインライブラリ	Powertrain Blockset™
	プラントモデルのシステム同定	プラントの実験データから数式モデルを同定	System Identification Toolbox™
	フィードフォワード・フィードバック 補償器の設計	線形制御(古典/現代)の関数ライブラリ	Control System Toolbox™
		補償器設計のSimulink用GUI	Simulink Control Design™
	プラントモデルのパラメータ同定、 制御パラメータの調整	最適化の関数ライブラリ	Optimization Toolbox™
パラメータ推定、応答最適化のGUI		Simulink Design Optimization™	
リアルタイム シミュレーション (RCP/HIL)	RCP、HIL用途のCコード生成	MATLABプログラムからCコード生成	MATLAB Coder™
		SimulinkモデルからCコード生成	Simulink Coder™
	リアルタイムシミュレーション	Speedgoat社のリアルタイムHWと連携	Simulink Real-Time™
実装 (マイコン)	固定小数点の設計	Simulinkモデルに固定小数点のデータ型を追加 固定小数点設計の専用GUI	Fixed-Point Designer™
	MCU実装用途の組込みCコード生成	組込みCコード生成	Embedded Coder®

ツール一覧

分類	用途	機能	製品
検証	静的解析	デッドロジック、ゼロ割、配列の範囲外アクセス解析	Simulink Design Verifier™
	モデリング基準準拠性確認 ブロック依存性解析	ISO26262、MISRA C 準拠 依存モデルの切り出し	Simulink Check™
	カバレッジ測定	MCDC解析	Simulink Coverage™
	テスト効率化	テストケース作成、管理、自動実行	Simulink Test™
システムズエンジニアリング	要件管理	要件リンク、可視化機能	Simulink Requirements™
	アーキテクチャ設計	アーキテクチャモデルの設計、Simulinkモデルと連携	System Composer™
予知保全	予知保全、異常検知	統計解析、特徴量抽出	Predictive Maintenance Toolbox™
信号処理	データ解析	フィルター設計、解析、スペクトル推定	Signal Processing Toolbox™
機械学習	機械学習	ニューラルネットワークの作成、分析、学習	Deep Learning Toolbox™
		分類、回帰モデル構築、分析、学習	Statistics and Machine Learning Toolbox™
	並列計算	マルチコア、GPU計算、クラスター並列処理	Parallel Computing Toolbox™
	実装コード生成	NVIDIA GPU向けのコード生成	GPU Coder™

ご清聴ありがとうございました



Accelerating the pace of engineering and science

© 2020 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

ユーザー事例

#	ゲスト講演者	講演タイトル	開催年・場所
1	Austin Sendek, Stanford University	Designing a better battery with machine learning Video and Slide https://jp.mathworks.com/videos/building-a-better-battery-with-machine-learning-1521196340290.html	2017 United States
2	福井 正博 様 立命館大学	MATLAB 製品を用いたリチウム・イオン電池の実践 Slide https://www.matlabexpo.com/content/dam/mathworks/mathworks-dot-com/images/events/matlabexpo/jp/2017/e3-ritsumeikan-battery-modeling.pdf	2017 Japan
3	加納 潤一 様 パナソニック株式会社	モデル予測制御を用いた蓄電池エネルギーマネジメント制御開発 Slide https://www.matlabexpo.com/content/dam/mathworks/mathworks-dot-com/images/events/matlabexpo/jp/2017/e4-panasonic-hems.pdf	2017 Japan
4	伊藤 章 様 株式会社デンソー	MATLAB によるエネルギーマネジメントシステム統合開発	2015 Japan

BMSのサンプルモデル

- MathWorksのBMSサンプルモデルは以下からダウンロードすることができます。

<https://jp.mathworks.com/matlabcentral/fileexchange/72865-design-and-test-lithium-ion-battery-management-algorithms>

The screenshot shows the MathWorks File Exchange page for the project "Design and Test Lithium Ion Battery Management Algorithms". The page includes a search bar, navigation links, and a "Download" button highlighted with a red box. A red arrow points from the "Download" button to the text "ここをクリック" (Click here). Another red arrow points from the "Download" button to the text "ダウンロードするためにはログインが必要です" (Login is required to download).

MathWorks® Products Solutions Academia Support Community Events

File Exchange Search File Exchange

MATLAB Central ▾ Files Authors My File Exchange Contribute About

Design and Test Lithium Ion Battery Management Algorithms
version 1.0.1 (8.95 MB) by Chirag **STAFF**

This example project can be used as a reference design to get started with designing Battery Management System with MATLAB and Simulink.

Trial software
★★★★★ 0 Ratings
12 Downloads ⓘ
Updated 03 Oct 2019
View License

+ Follow **Download**

ダウンロードするためにはログインが必要です **ここをクリック**

Overview

This example project can be used as a reference design to get started with designing Lithium Ion Battery Management System (BMS) with MATLAB and Simulink.

Project includes Simulink models for BMS Algorithms such as:

1. State of Charge estimation using Extended Kalman Filter, Unscented Kalman Filter
2. Passive Battery Cell Balancing
3. State Machine for Pre-charging and Contactor Management
4. Fault Management - Over/Under Voltage, Over Current, Over Temperature etc.
5. Charge and Discharge Current Limit Calculations

To design and test these algorithms, project also includes files for

1. Lithium Battery Cell Parameter Estimation

MATLAB Release Compatibility
Created with R2019b
Compatible with any release

Platform Compatibility
 Windows macOS Linux

Tags
battery battery contactor...
battery fault man... battery managemen...

ソリューションページも是非ご覧ください！

MathWorks® 製品 ソリューション アカデミア サポート

Model-Based Design

お問い合わせ 購入方法 Miyagawaさん

Search MathWorks.com

What is Model-Based Design?

In Model-Based Design, a system model is at the center of the process, from requirements development, through design, implementation, and testing.

[Watch video](#)

Model-Based Design

Use Model-Based Design with MATLAB® and Simulink® to improve product quality and reduce development time by 50% or more.

See why and how teams adopt Model-Based Design.

- Integrate testing with design requirements
- Refine algorithms through multidomain simulation
- Automatically generate embedded software code and documentation
- Develop and reuse test suites

[Explore Model-Based Design with Simulink](#)

[Contact sales](#)

[Request a trial](#)

6,000 COMMUNICATIONS SYSTEMS

Model-Based Design of Safety-Critical Avionics Systems (Highlights) 7:12

Weinmann Develops Life-Saving Transport Ventilator

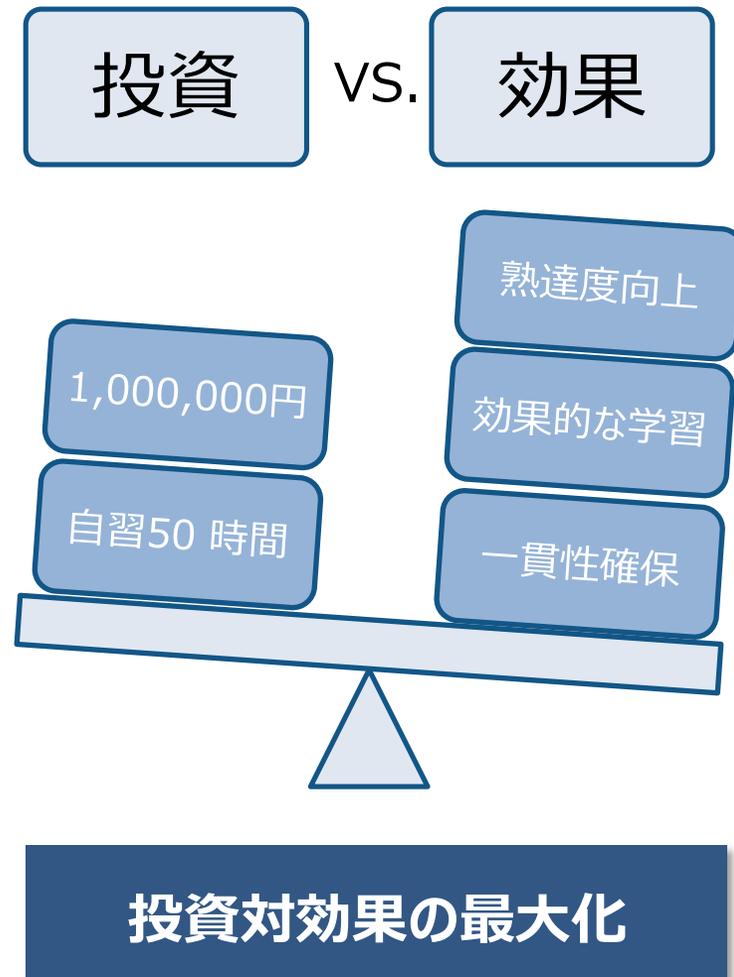
Alstom Grid Develops HVDC Transmission Control System

Model-Based Design with MATLAB and Simulink 3:17

Simulink Overview 2:04

Measuring Return on Investment of Model-Based Design 2:09

トレーニング・コンサルティングサービス



トレーニングサービス

MathWorks製品の機能の理解や使い方を、短期間で最大限に習得して頂けるような、多種多様な教育カリキュラムを提供します。

<http://jp.mathworks.com/services/training/>

コンサルティングサービス

お客様のプロジェクトを成功させるため、お客様のモデル・データを使った、コンサルティングサービスを提供します。

<http://jp.mathworks.com/services/consulting/>