

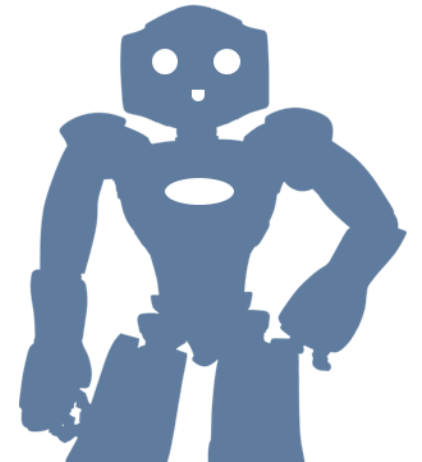
# MATLAB EXPO 2019

## Developing Battery Management System using Simulink

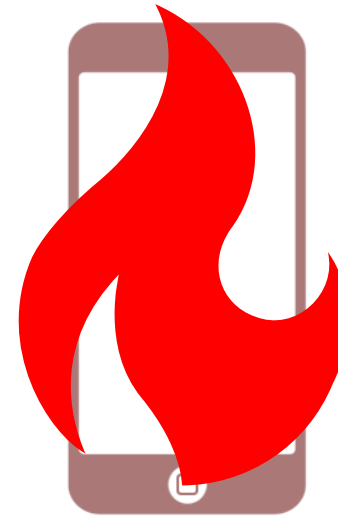
Romain LACHAUX



Battery: a good answer to energy storage across industries...



... with some risks to keep under control



**At Least Three Electric Vehicles Have  
Spontaneously Exploded in China This Week**

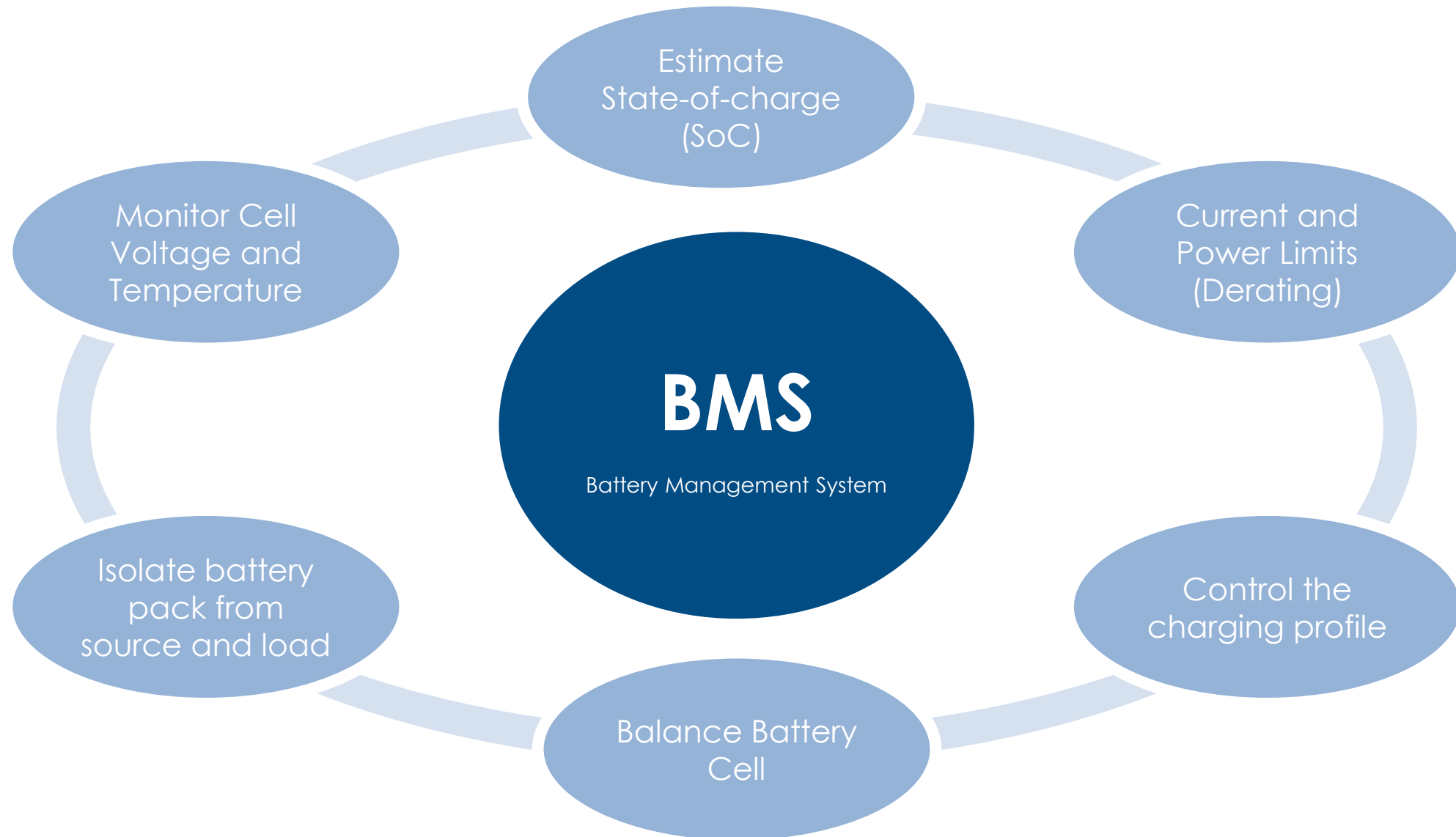
By Zhao Runhua / Apr 25, 2019 08:28 PM / Business & Tech

**Newsweek**

**RYANAIR FLIGHT EVACUATED AFTER PHONE  
CATCHES FIRE ON PLANE**

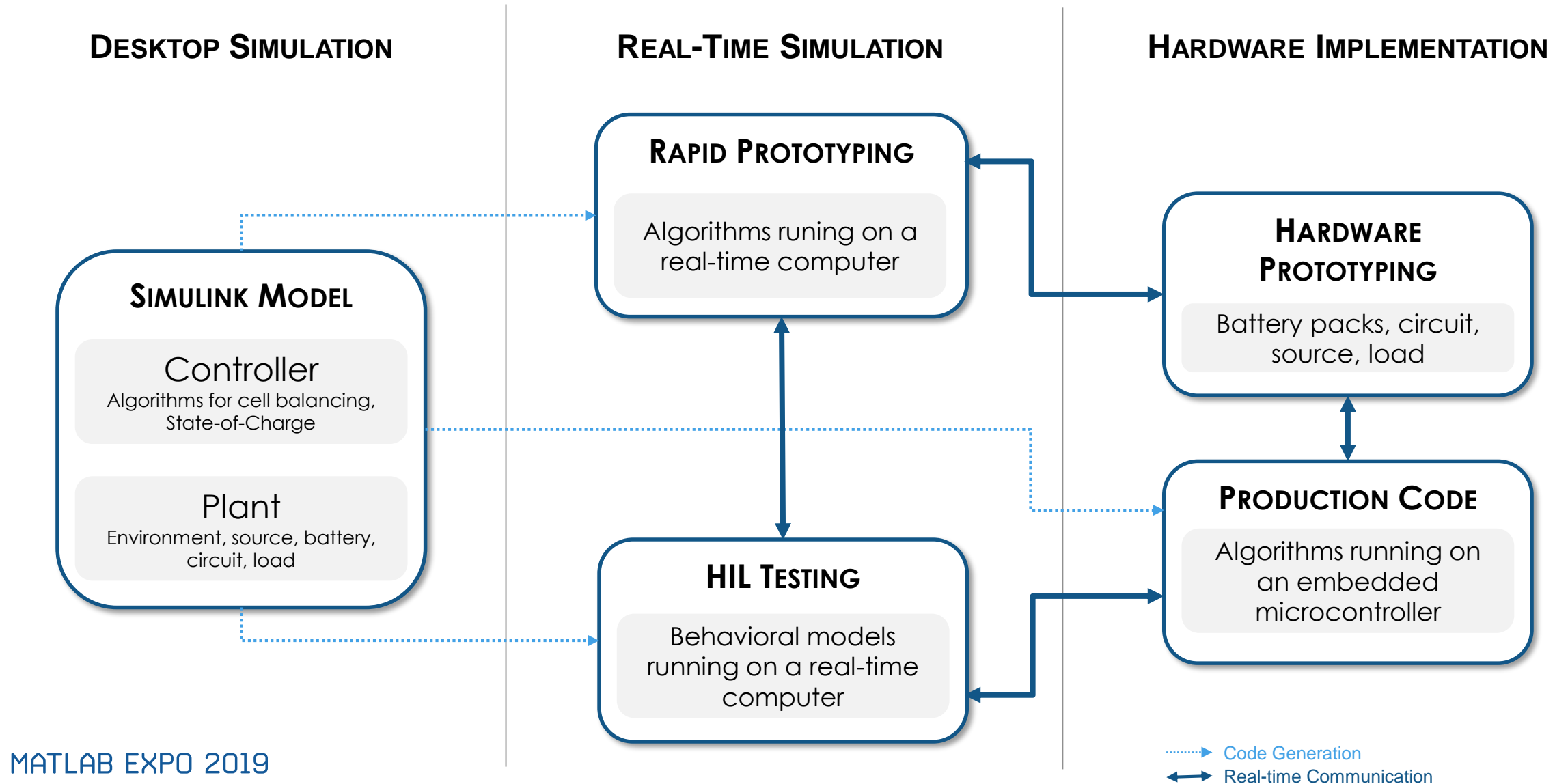
BY JENNI FINK ON 8/1/18 AT 11:00 AM EDT

CHALLENGE: Design and verify **battery management functions**



SOLUTION: Perform **system-level simulations** with **Simulink**

# BMS Development Workflow with Simulink and MBD



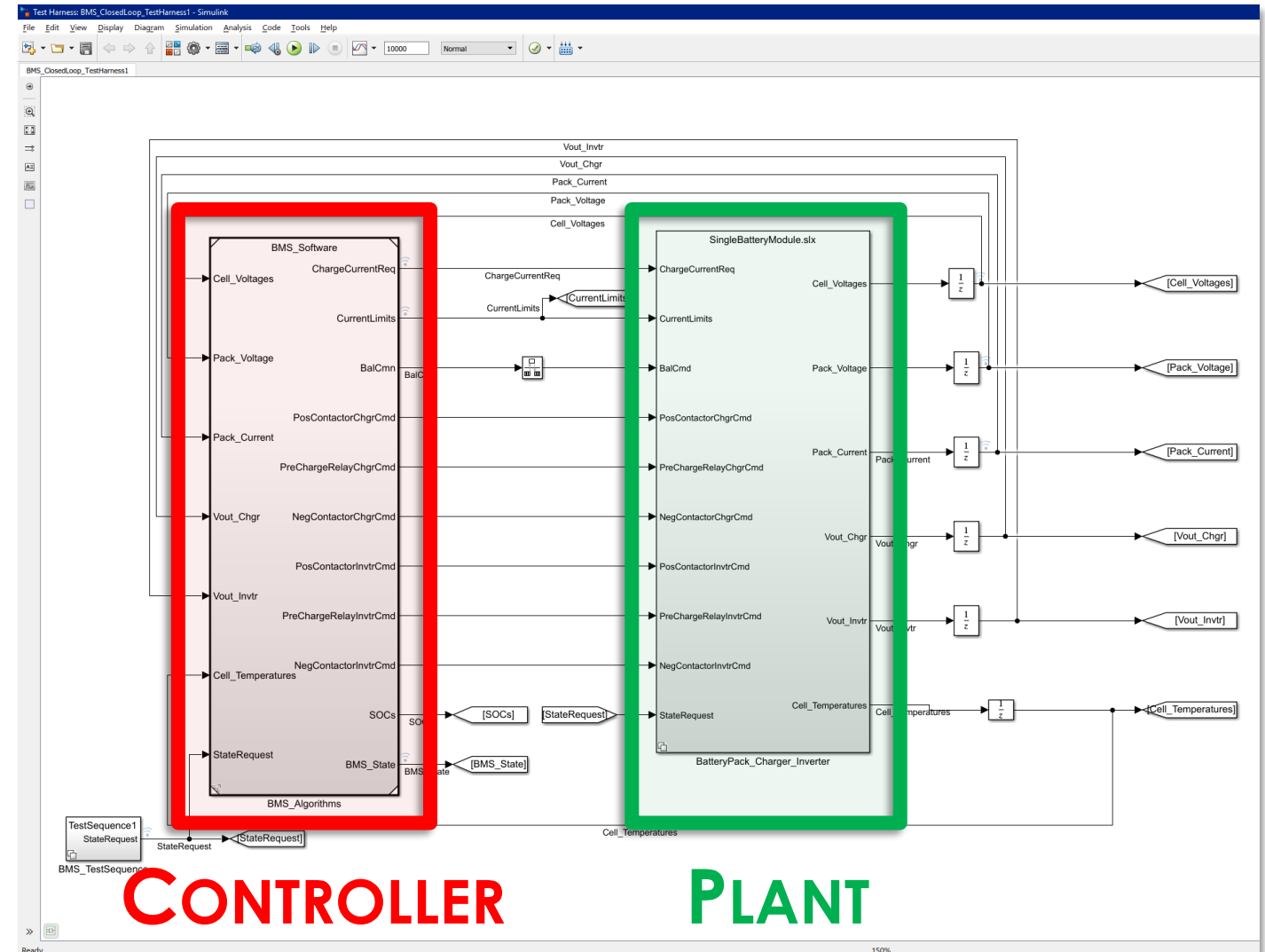
# Agenda

- ❑ **BMS Model Demo**
  - Physical Modeling
  - BMS Algorithms
  
- ❑ Deployment on Hardware
  - Code Generation
  - Real-Time Testing



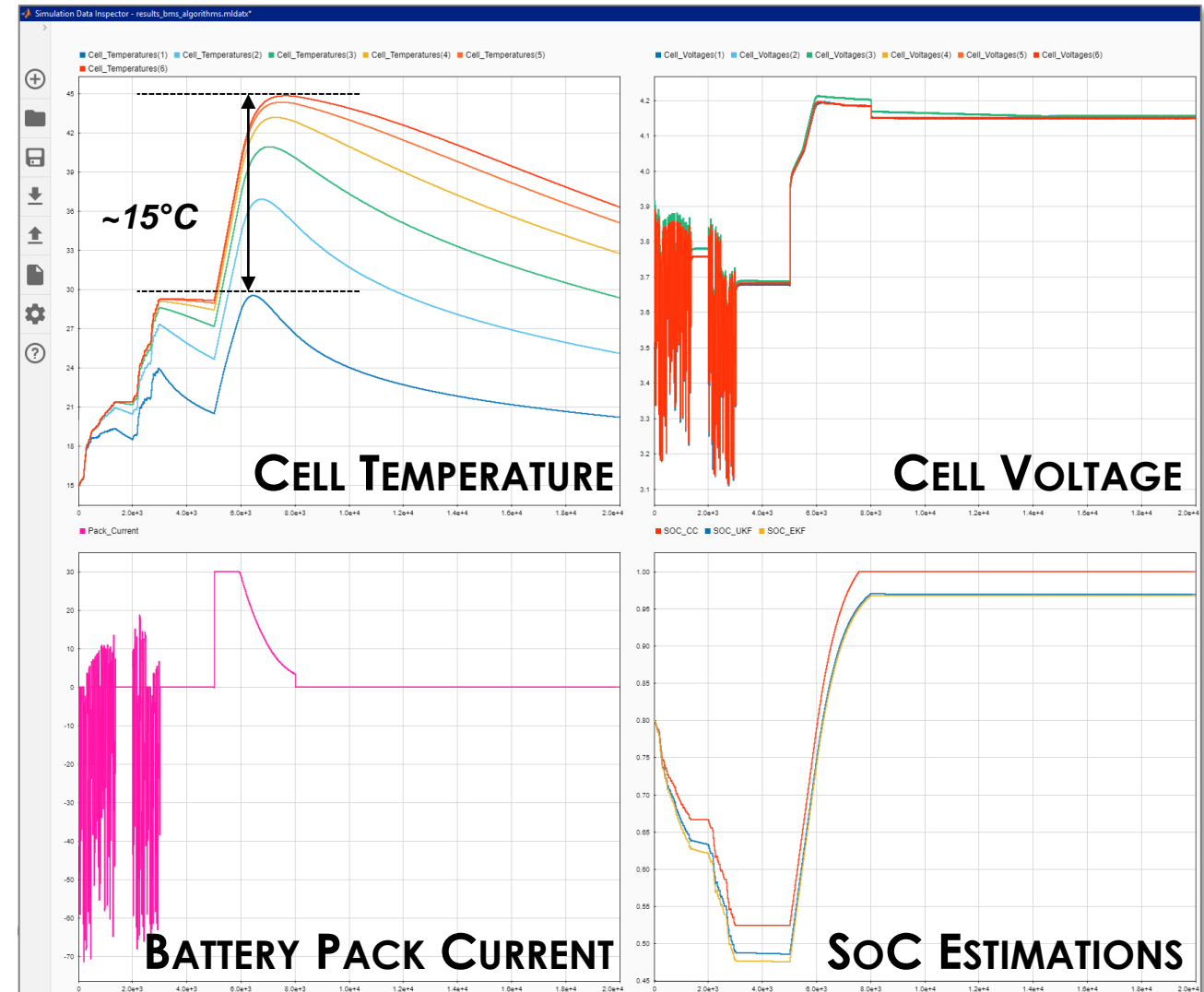
# BMS Model Overview

- System Model
  - Controller: BMS Algorithms
  - Plant : Physical Modeling
- Advantages of System-Level Simulation:
  - Quick design iterations
  - Early results in the development workflow
  - Possible to test each part alone or together in the same model (**Closed-loop testing**)



# Simulation Results Overview

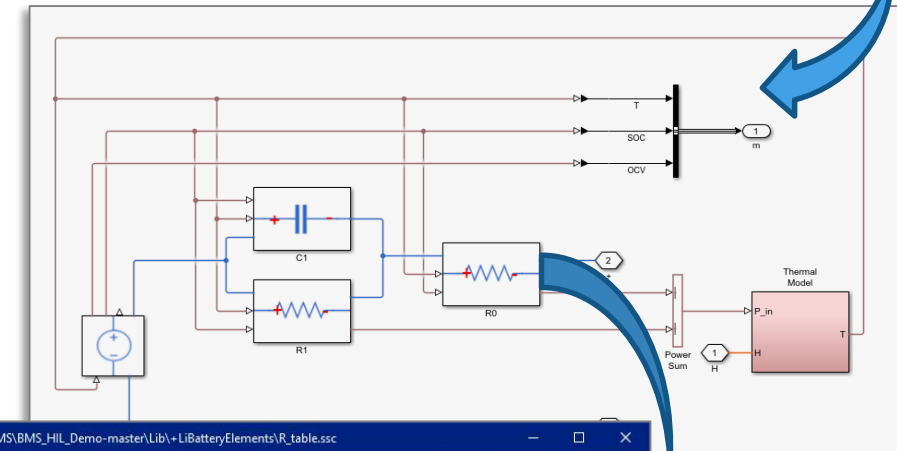
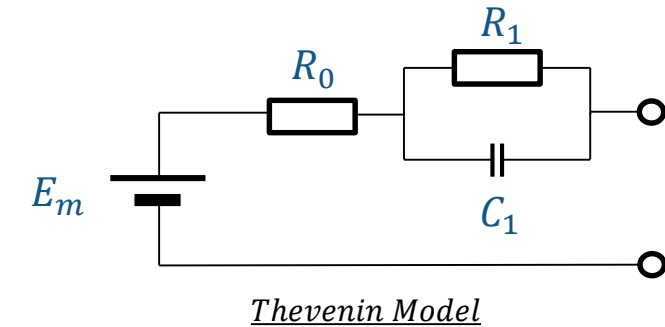
- **Early results** during design process
  - ➔ Possible to refine or add missing requirements
- Example:
  - Temperature differences
  - Potential impact on cells ageing
  - ➔ Need of a cooling system?



# PLANT: Battery Physical Modeling with **Simulink** and **Simscape**

# Battery Cell Modeling

- **Thevenin Model** (1st Order) to represent electrical behavior of battery cell
- Model based on Simscape Foundation Library components...
- ... with dependance upon SoC and temperature by **modifying source code**



```
equations
    if v > Vf
        i == (v - Vf*
    else
        i == v*Goff;
```

**Custom Component**

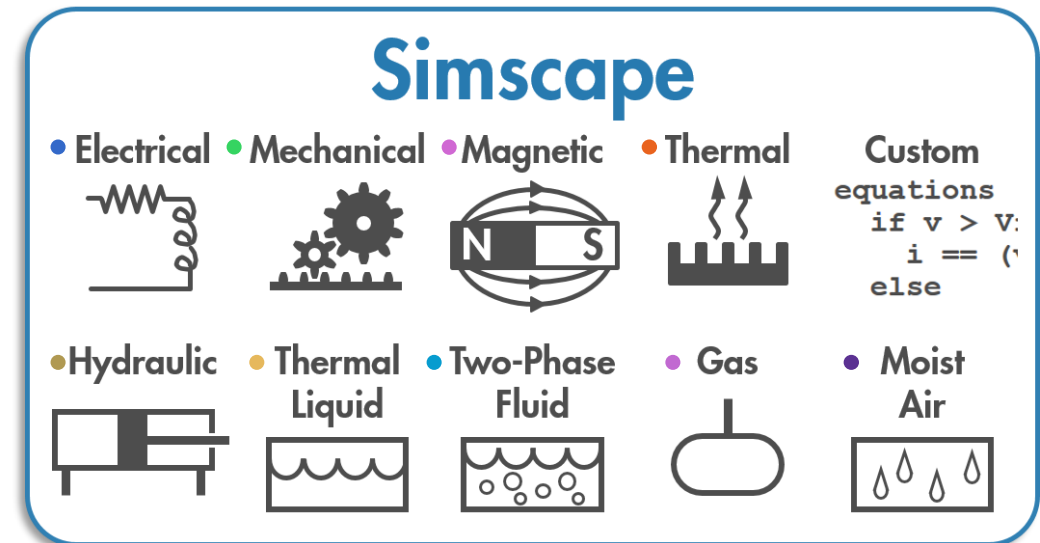
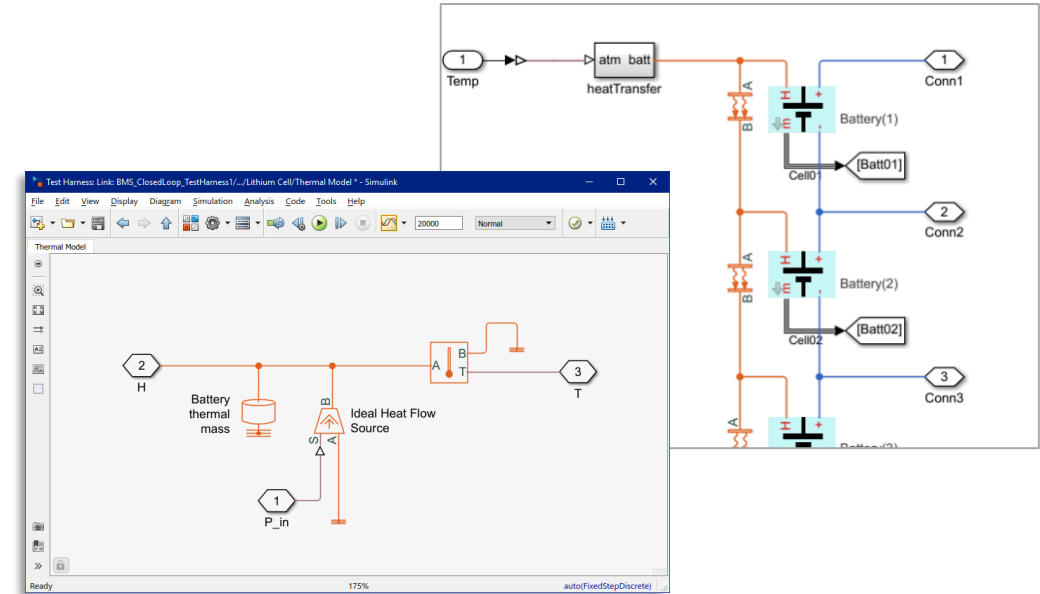
```

C:\MATLAB\09_Webinar\01_BMS\BMS_HIL_Demo-master\Lib\LiBatteryElements\R_table.ssc
EDITOR  VIEW
New Open Save Find Files Compare Go To Comment % fx Breakpoints
Find Indent EDIT BREAKPOINTS
38 equations
39 let
40     % Perform the table lookup
41     R = tablelookup(SOC_Table,Temp_Table,R_Table,SOC,T,...
42         interpolation=linear,extrapolation=nearest)
43 in
44     % Electrical Equations
45     v == i*R;
46     pow == v*i;
47 end
  
```

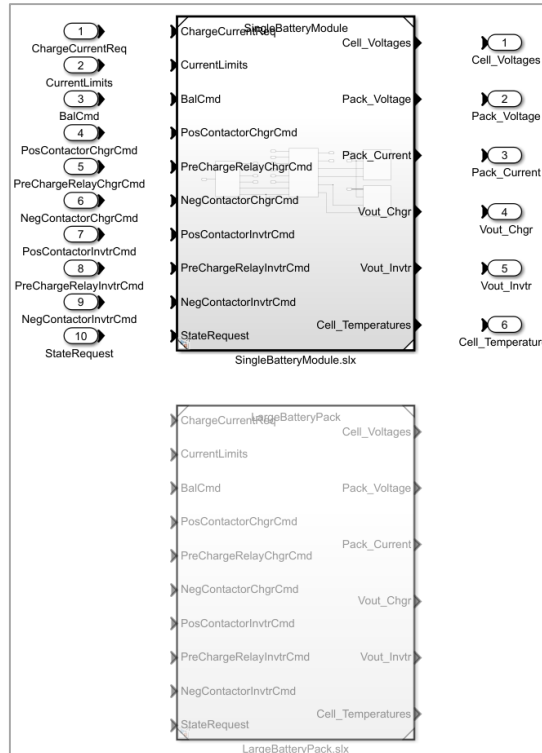
# Battery Cell Modeling

- What about **thermal behavior**?
  - Cell heat up under load
  - Convection heat flux between cells
  - Thermal exchange with environment

→ **Thermal component** from Foundation Library



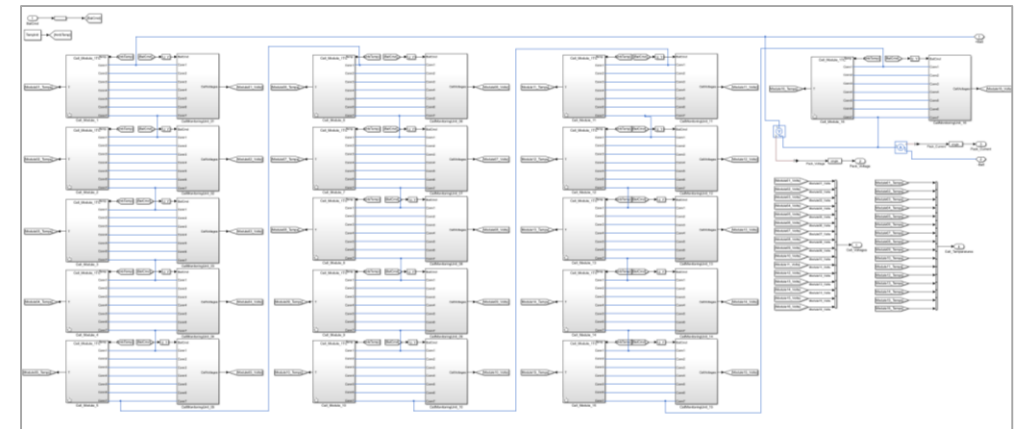
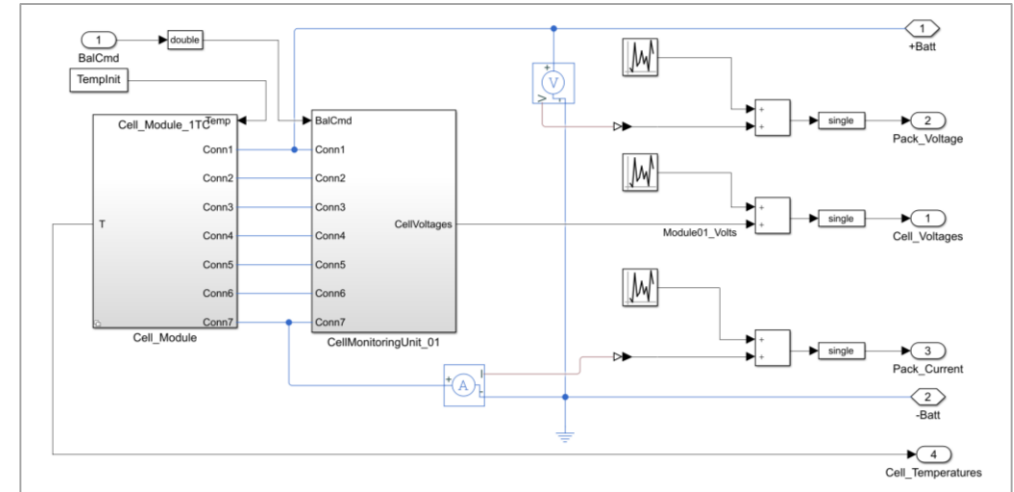
# Battery Pack



1 Module of 6 cells

OR

16 Module of 6 cells

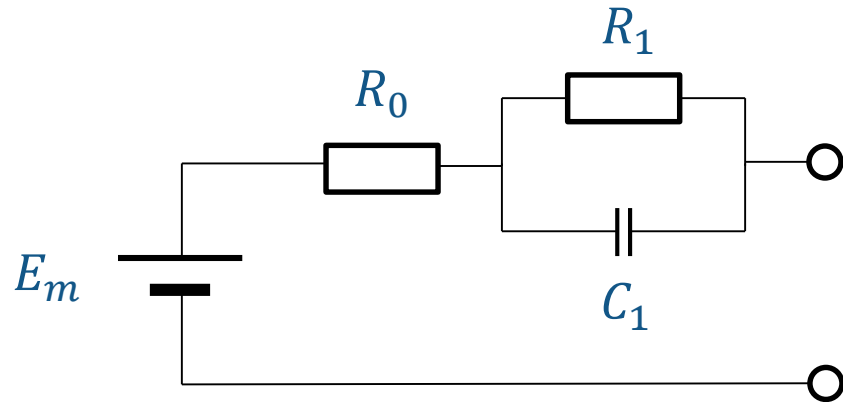


Variant Model

Possible to model **different architecture or technological choice** of a subsystem in the **same model**

# Tuning a Lithium Battery Model to Match Measured Data

## Cell Physical Model:



Thevenin Model

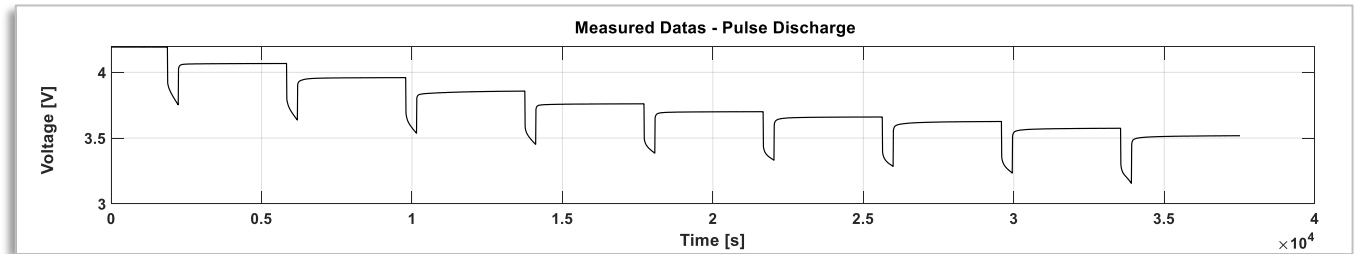
## Lithium Cell Characteristic Measurement:

$E_m$	SOC 1	SOC 0.9	SOC 0.8	...	SOC 0
5°C	4.20 V	4.12 V	4.05 V	...	3.09 V
20°C	4.18 V	4.09 V	4.01 V	...	3.05 V
40°C	4.15 V	4.02 V	3.97 V	...	3.01 V

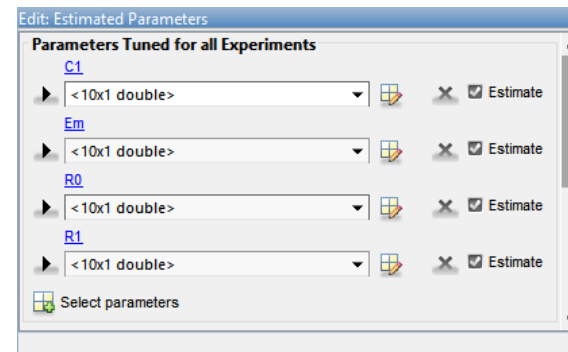
OBJECTIVE: Match model behavior to tests measurements

# Estimating Parameters Using Measured Data

1. **Import** measurement datas

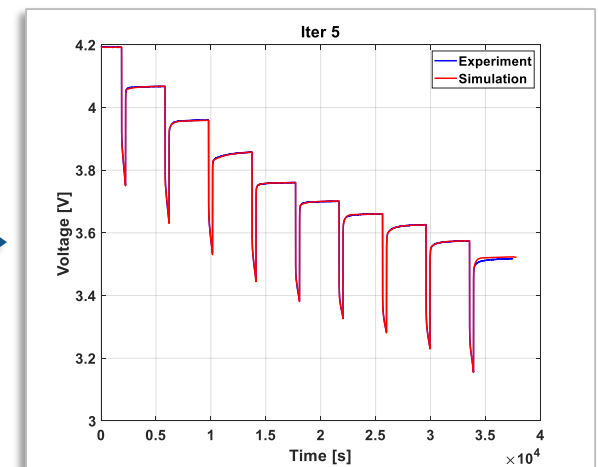
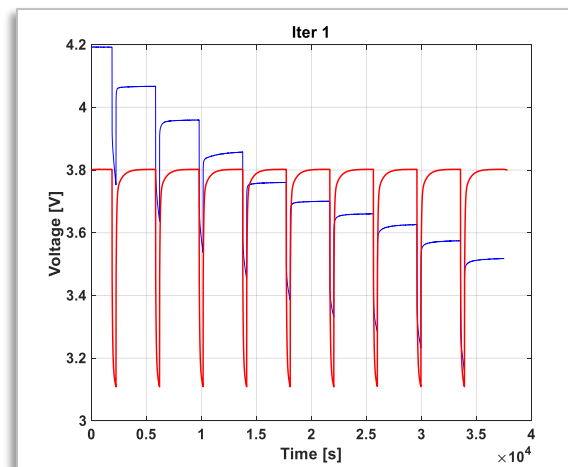


2. **Identify** parameters and set range



Name:	Em
Value:	[4.1;4.1;4.1...
Initial guess:	Em
Minimum:	3
Maximum:	5

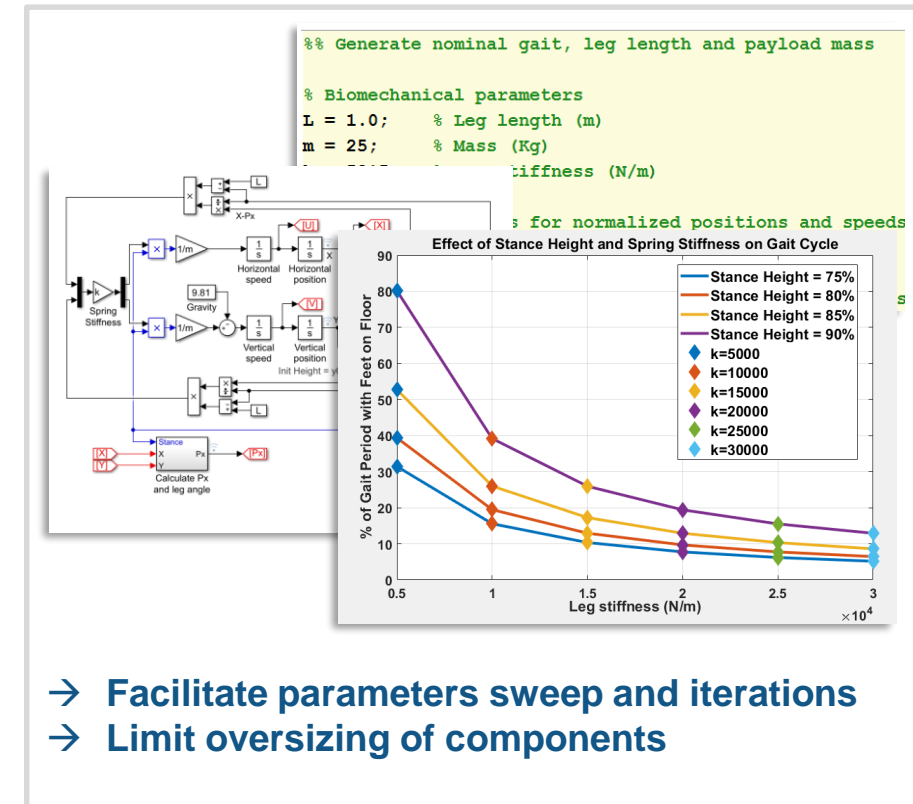
3. **Perform estimation**





# Advantages of Physical Modeling

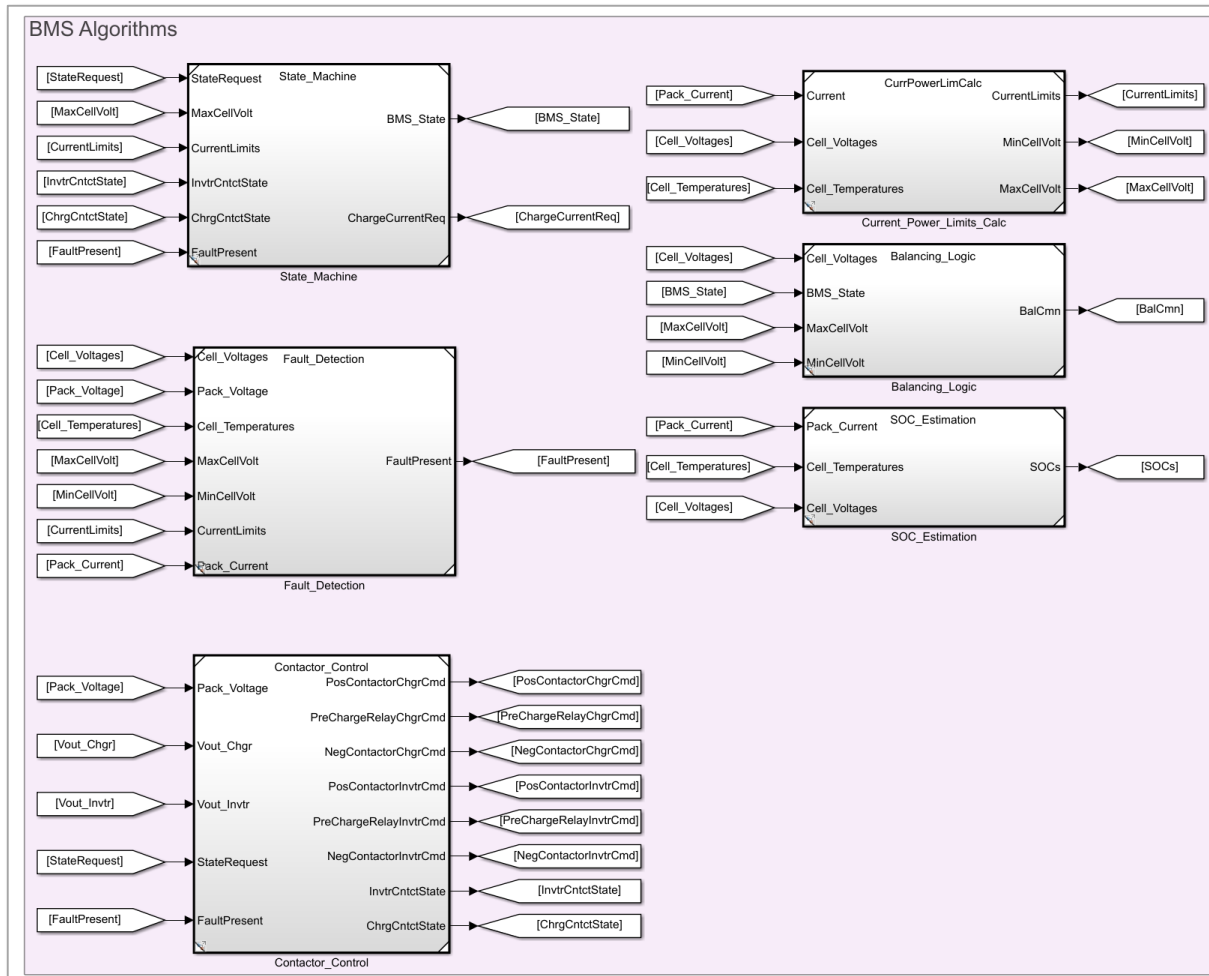
- With this physical model, you can
  - **Evaluate** your architecture
  - **Optimize** your design
  - **Refine** and **Validate** your requirements



+ Enable **Closed-loop testing of your control algorithms** to verify and validate it

# CONTROLLER: BMS Algorithms with **Simulink** and **Stateflow**

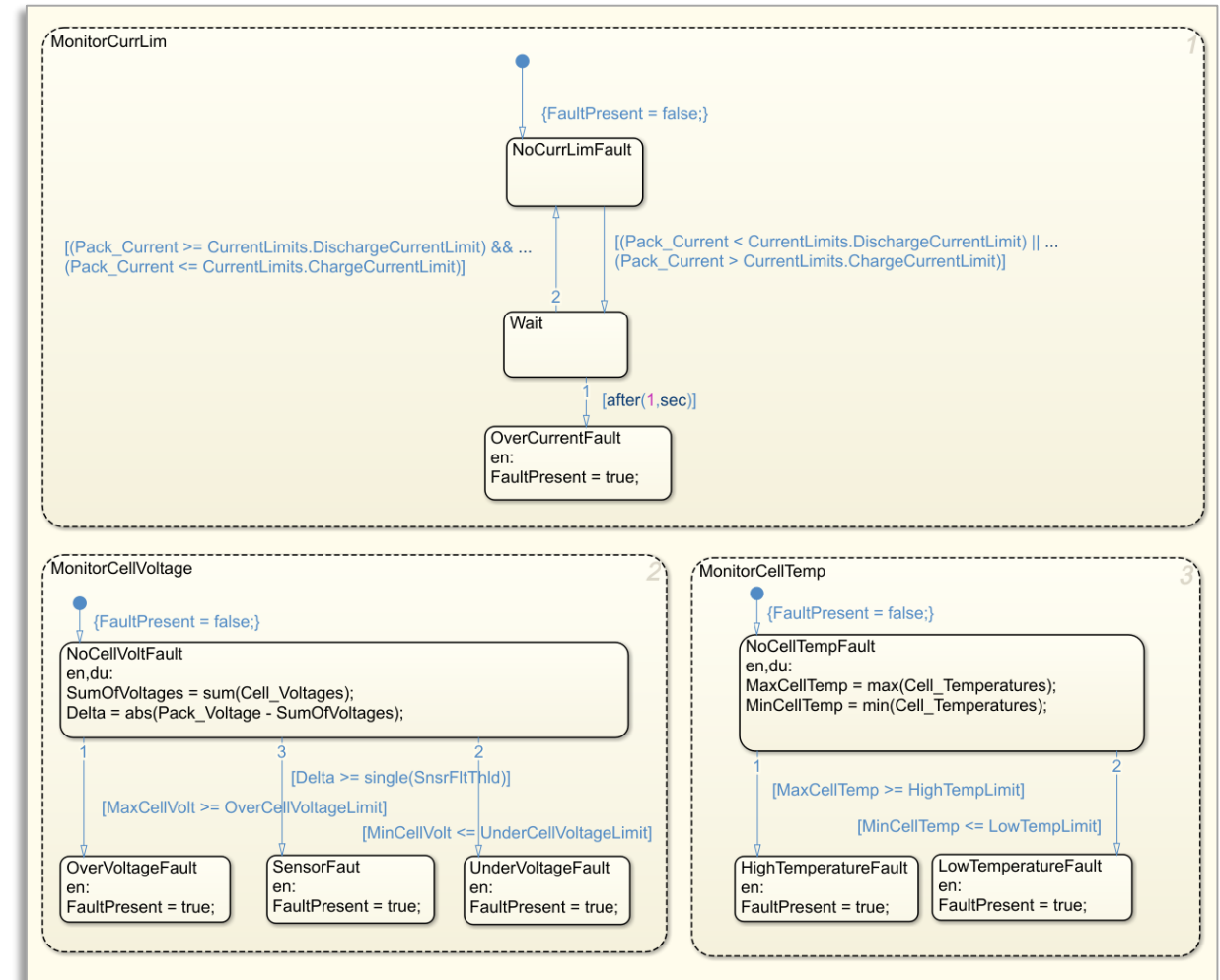
# Battery Management System Functions



- **Battery State**
- **Fault Management**
- **Battery isolation control**
- **Derating Calculations**
- **State-of-Charge Estimation**

# Fault Management

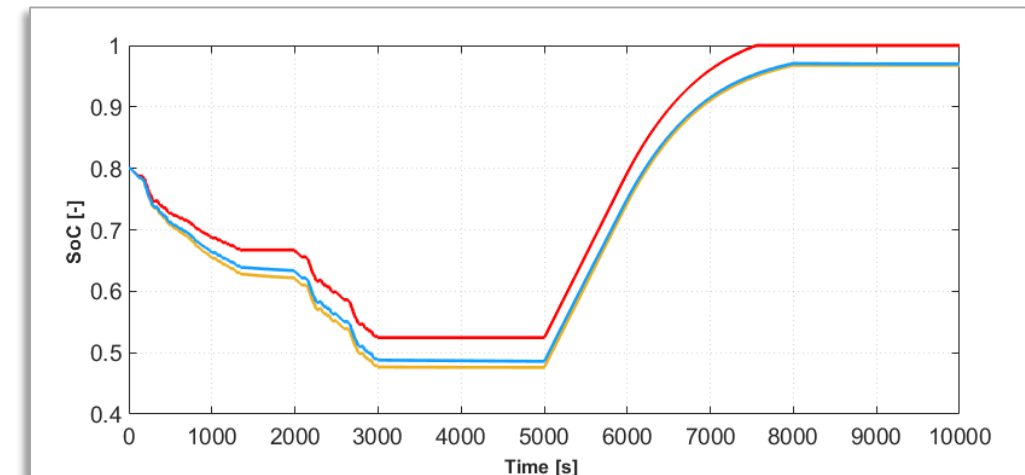
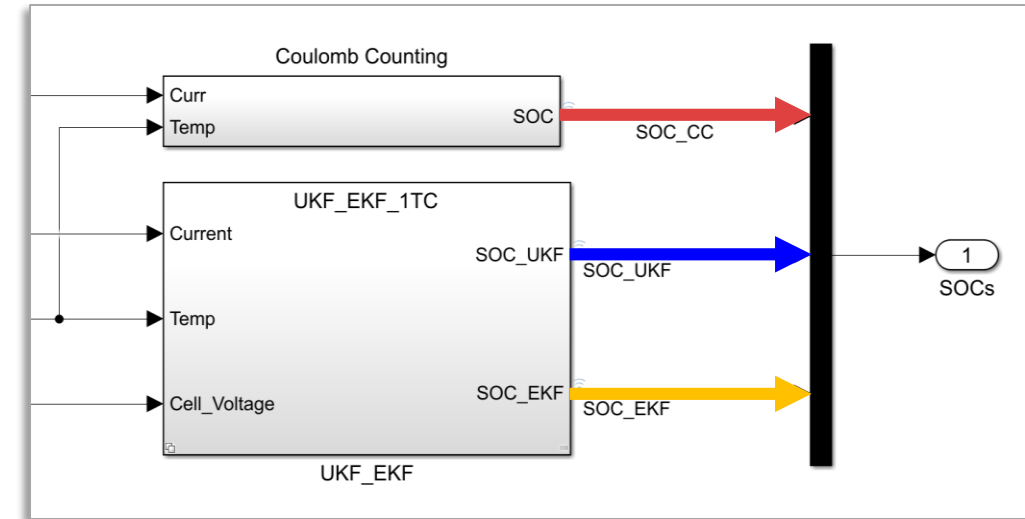
- Monitoring three physical channels:
  - Battery Pack **Current**
  - Cell **Voltage**
  - Cell **Temperature**
- Broadcasting Fault Presence to other BMS subsystems
  - Contactor Opening (SAFETY)
  - BMS\_State == FAULT



# State-Of-Charge Estimation

- Two methods:
  - Coulomb Counting**
    - ✓ : Simple to implement / **low computational needs**
    - ✗ : Accuracy and robustness
  - Kalman Filtering\***
    - ✓ : **High accuracy** by including a nonlinear battery model which uses current and voltage measurement
    - ✗ : Slightly higher computational effort

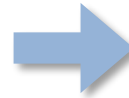
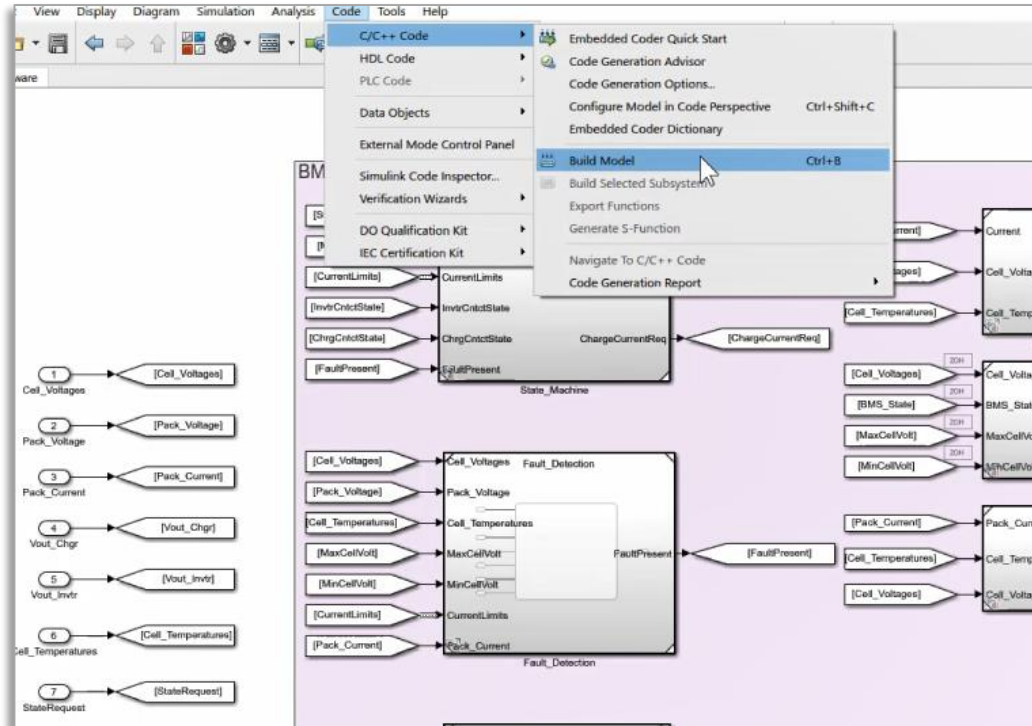
\* : ready to use block available in Control System Toolbox or System Identification Toolbox



# Agenda

- BMS Model Overview
  - Physical Modeling
  - BMS Algorithms
  
- **Deployment on Hardware**
  - **Code Generation**
  - **Real-Time Testing**

# Generate C/C++ Code From BMS Models



**Code Generation Report**

Find:  Match Case

**Contents**

- [Summary](#)
- [Subsystem Report](#)
- [Code Interface Report](#)
- [Traceability Report](#)
- [Static Code Metrics Report](#)
- [List of inserted blocks](#)
- [Code Replacements Report](#)
- [Coder Assumptions](#)

**Generated Code**

- Main file**
  - [ert\\_main.c](#)
- Model files**
  - [BMS\\_Software.c](#)
  - [BMS\\_Software.h](#)
  - [BMS\\_Software\\_private.h](#)
  - [BMS\\_Software\\_types.h](#)
- Shared files**
  - [rtwtypes.h](#)

**Referenced Models**

- [Balancing\\_Logic](#)
- [Contactor\\_Control](#)
- [CurrPowerLimCalc](#)
- [Fault\\_Detection](#)
- [SOC\\_Estimation](#)
- [State\\_Machine](#)

**File: BMS\_Software.c**

```

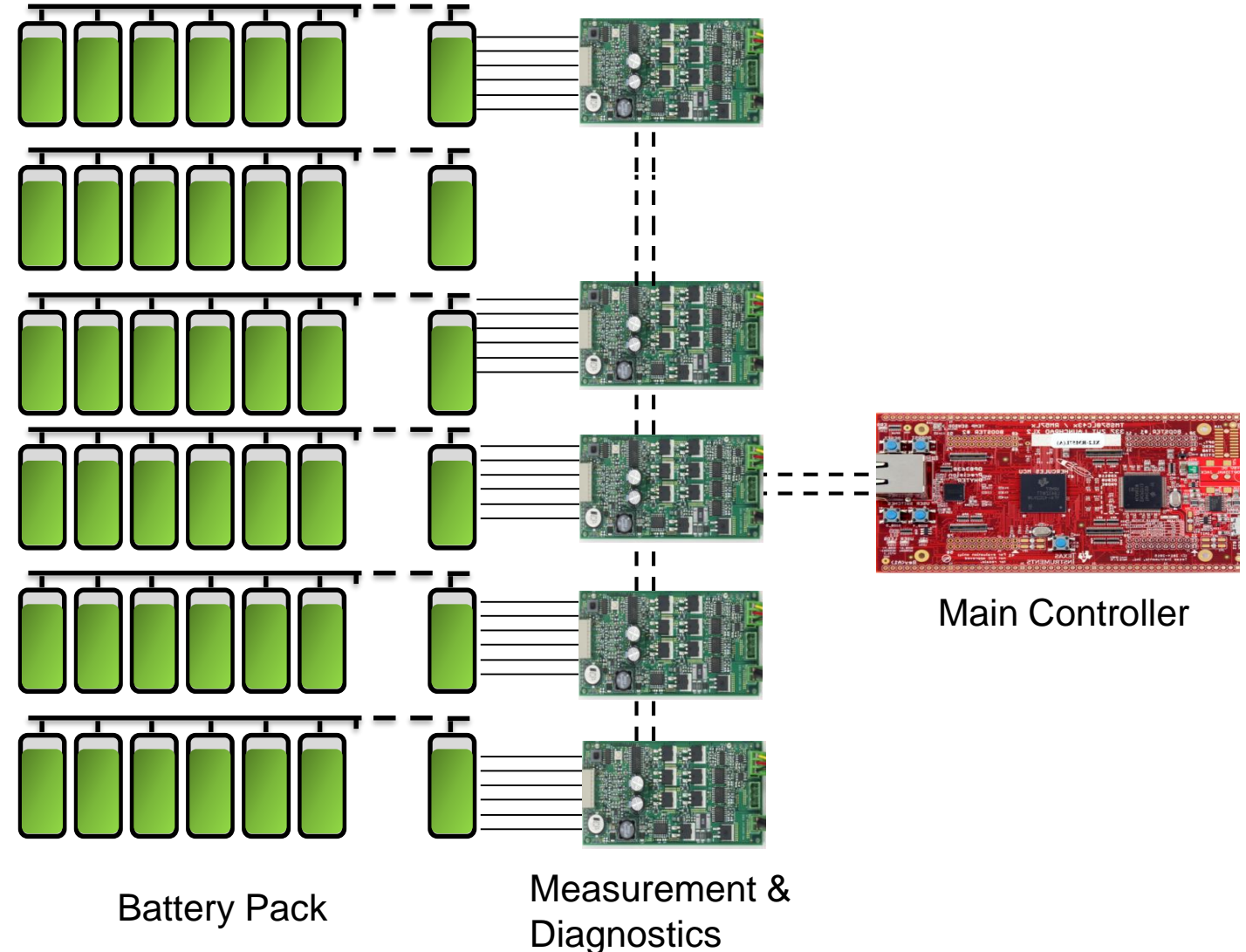
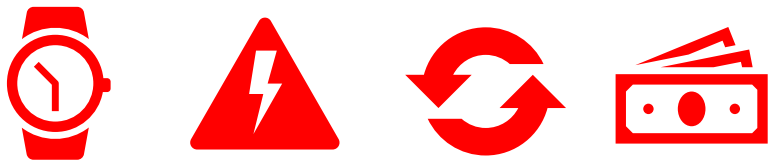
1  /*
2  * File: BMS_Software.c
3  *
4  * Code generated for Simulink model 'BMS_Software'.
5  *
6  * Model version          : 1.545
7  * Simulink Code version  : 9.1 (R2019a) 23-Nov-2018
8  * C/C++ source code generated on : Tue May 7 09:58:22 2019
9  */
10
11 * Target selection: ert.tlc
12 * Embedded hardware selection: Texas Instruments->TMS570 Cortex-R4
13 * Code generation objectives:
14 *   1. MISRA C:2012 guidelines
15 *   2. Execution efficiency
16 * Validation result: Not run
17 */
18
19 #include "BMS_Software.h"
20 #include "BMS_Software_private.h"
21
22 /* Block signals and states (default storage) */
23 DW_BMS_Software_T BMS_Software_DW;
24
25 /* External inputs (root inport signals with default storage) */
26 ExtU_BMS_Software_T BMS_Software_U;
27
28 /* External outputs (root outports fed by signals with default storage) */
29 ExtY_BMS_Software_T BMS_Software_Y;
30
31 /* Real-time model */
32 RT_MODEL_BMS_Software_T BMS_Software_M;
33 RT_MODEL_BMS_Software_T *const BMS_Software_M = &BMS_Software_M;
34
35 /* Model step function for TID0 */
36 void BMS_Software_step0(void) /* Sample time: [0.1s, 0.0s] */
37 {
38     /* Update the flag to indicate when data transfers from
39      * Sample time: [0.1s, 0.0s] to Sample time: [5.0s, 0.0s] */
40     BMS_Software_M->Timing.RateInteraction.b_TID0_1 =
41         (BMS_Software_M->Timing.RateInteraction.TID0_1 == 0);
42     (BMS_Software_M->Timing.RateInteraction.TID0_1)++;
43     if ((BMS_Software_M->Timing.RateInteraction.TID0_1) > 49) {
44         BMS_Software_M->Timing.RateInteraction.TID0_1 = 0;
45     }
46
47     /* ModelReference: '<Root>/Current_Power_Limits_Calc' incorporates:
48      * Input: '<Root>/Cell_Temperatures'
49      * Input: '<Root>/Cell_Voltages'
50      * Input: '<Root>/Pack_Current'
51      * Output: '<Root>/CurrentLimits'
52      * Block requirements for '<Root>/Current_Power_Limits_Calc':

```

- Generate target optimized C/C++ code
- Fine-tune code optimizations, package and build generated code

# Real-Time Testing of Battery Management System

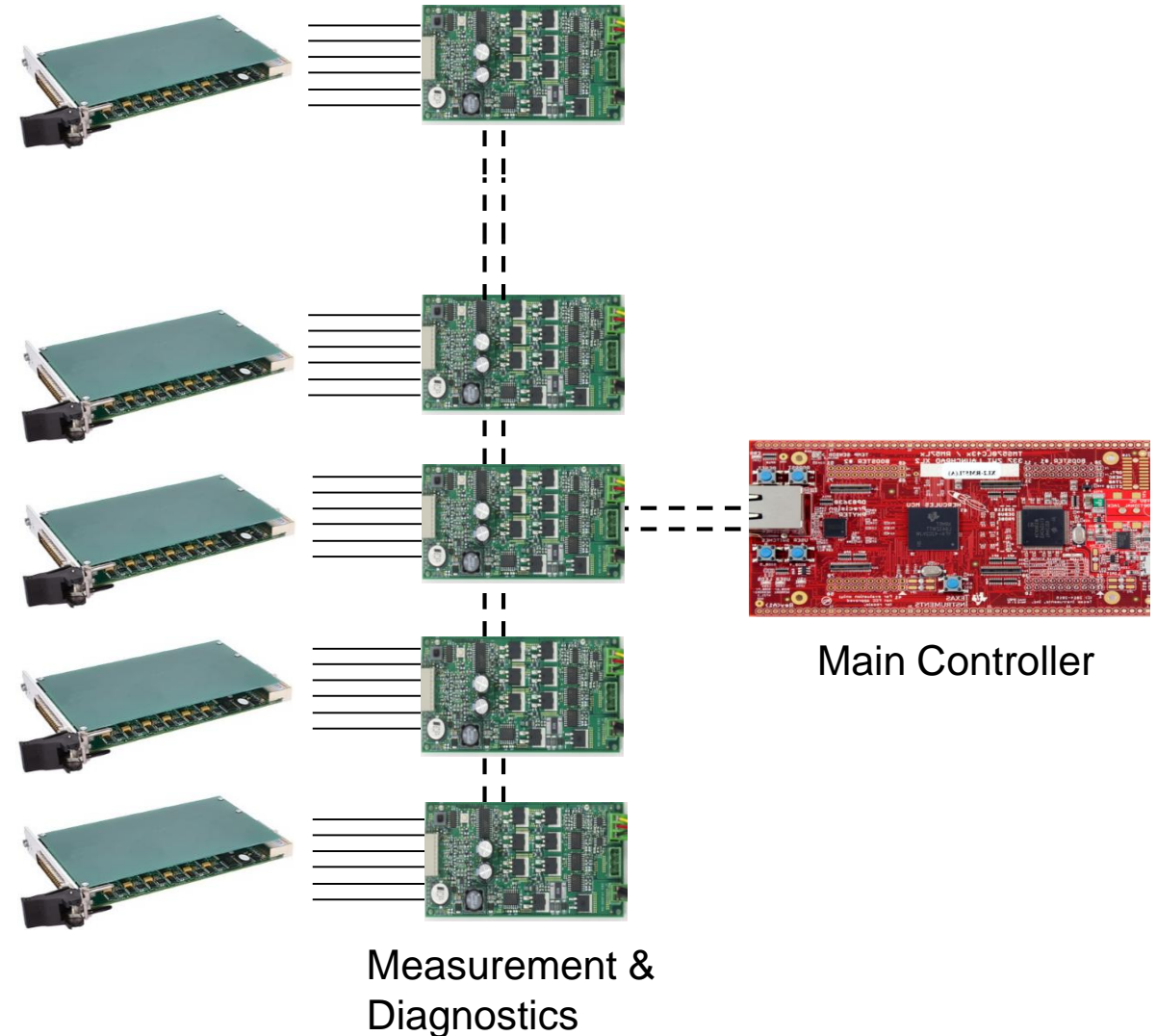
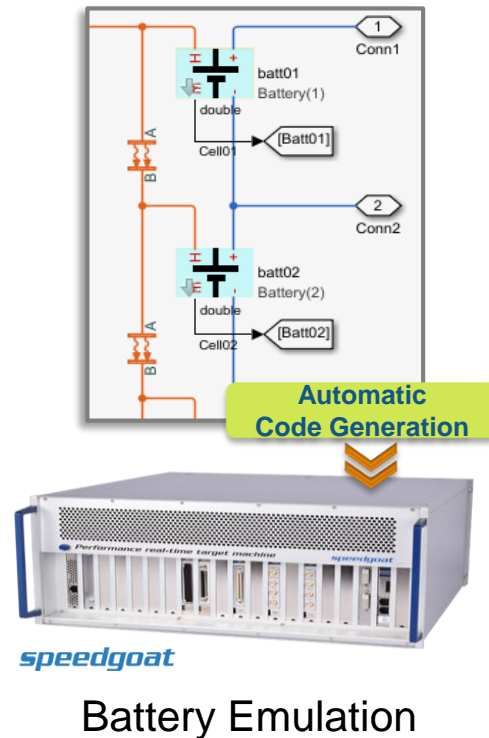
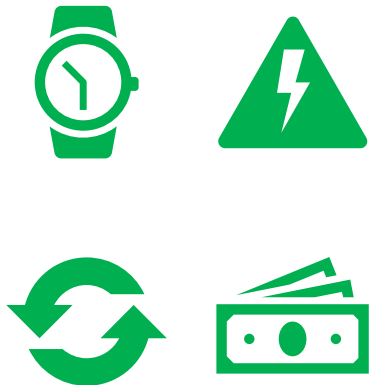
- Testing BMS with Battery Cells
    - Longer test cycles
    - Difficult to test fault conditions
    - Difficult to reproduce results
    - Limited test automation
- ➔ Costs (Hardware prototype, possible failure, several people to perform tests, etc)



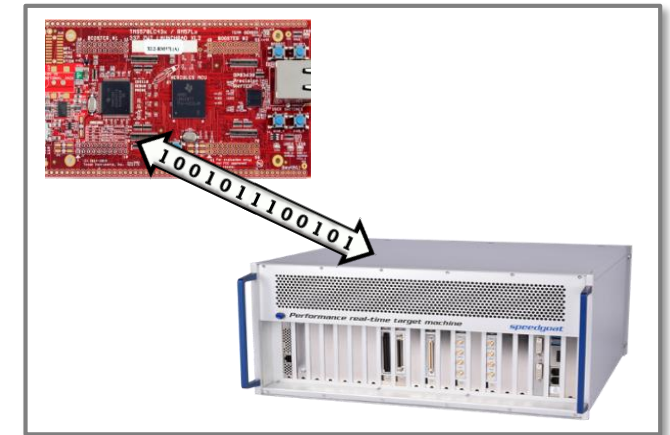
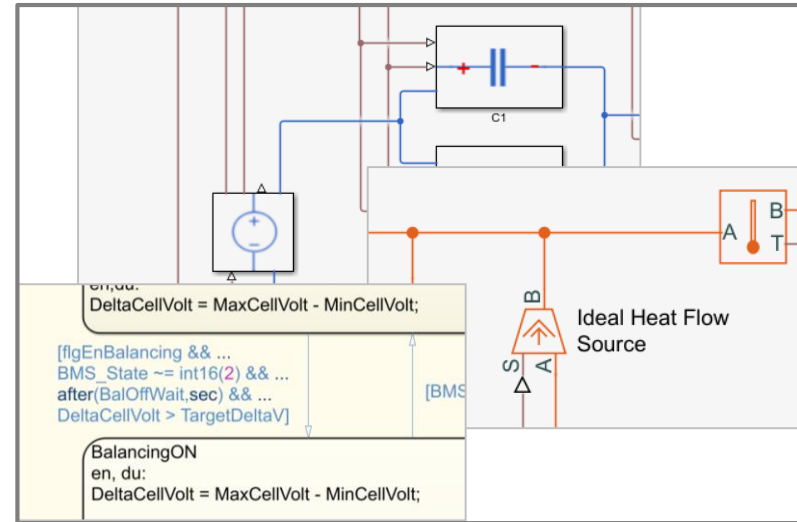
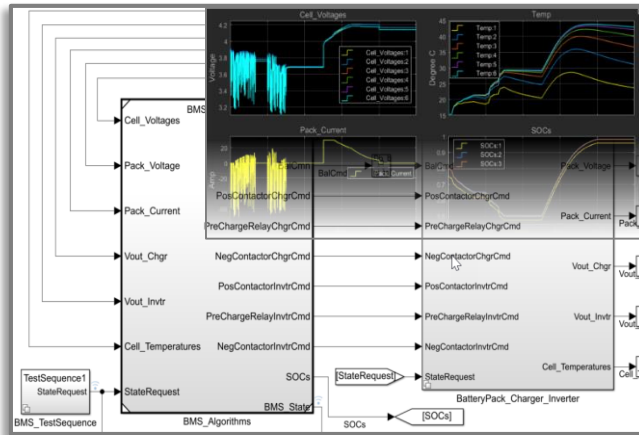


# Hardware-In-Loop Testing of Battery Management System

- Testing BMS with Emulated Battery Cells
  - Reduce testing time
  - Test fault conditions safely
  - Automate testing



# BMS Development with Simulink



**Reduce Design  
Iteration Time**



**Collaborate  
Across Domains**



**Gain Confidence  
in Design**