Agenda

- What is BMS and what engineers worry about?
- Start with single battery cell and build a pack
- Design BMS algorithms
- Generate code and deploy
- Model V&V
- Hardware-in-Loop testing
- Summary
What is BMS?

**Software**

- Supervisory tasks
- SOC estimation
- Contactor management
- Isolation monitoring
- Fault detection and recovery
- Thermal Management
- Current & Power Limits

**Electronics**

- Block Voltage, Temperature Measurement
- Cell Diagnostic
- Cell Balancing

**Battery Pack**

- Battery Pack
- Software
- Electronics

MATLAB EXPO 2019
Motivation

Collaboration Gap

Long Iteration Cycles

Safety Critical System

Multi-Domain Modeling Environment

Simulations and Code Generation

Model V&V and Hardware-In-Loop Testing

MATLAB EXPO 2019
Where do we start?

Gain insight into cell behavior and model it
Equivalent Circuit Model

\[ [E_x \ R_x \ C_x] = f(SOC, \ Temperature\ldots) \]
Battery Cell Blocks in Simscape

- Chose block for fidelity and simulation speed
- Parameterize as function of SOC & Temperature
- Add thermal and fade effects
- Create custom battery blocks using Simscape language or Simulink
Parameter Estimation

Goal:
Model should match real data as closely as possible

Parameter Estimation

Goal:
Model should match real data as closely as possible

Parameter Estimation

Goal:
Model should match real data as closely as possible

Parameter Estimation

Goal:
Model should match real data as closely as possible

Parameter Estimation

Goal:
Model should match real data as closely as possible

Parameter Estimation

Goal:
Model should match real data as closely as possible

Parameter Estimation

Goal:
Model should match real data as closely as possible

Parameter Estimation

Goal:
Model should match real data as closely as possible

Parameter Estimation

Goal:
Model should match real data as closely as possible

Parameter Estimation

Goal:
Model should match real data as closely as possible

Parameter Estimation

Goal:
Model should match real data as closely as possible

Parameter Estimation

Goal:
Model should match real data as closely as possible

Parameter Estimation

Goal:
Model should match real data as closely as possible

Parameter Estimation

Goal:
Model should match real data as closely as possible

Parameter Estimation

Goal:
Model should match real data as closely as possible

Parameter Estimation

Goal:
Model should match real data as closely as possible

Parameter Estimation

Goal:
Model should match real data as closely as possible

Parameter Estimation

Goal:
Model should match real data as closely as possible

Parameter Estimation

Goal:
Model should match real data as closely as possible

Parameter Estimation

Goal:
Model should match real data as closely as possible

Parameter Estimation

Goal:
Model should match real data as closely as possible

Parameter Estimation

Goal:
Model should match real data as closely as possible
Parameter Estimation

Before

After

Double click here to open the Parameter Estimation App with preloaded data

MATLAB EXPO 2019
Look-up Tables

Repeat parameter estimation for each Temperature break-point in LUT

<table>
<thead>
<tr>
<th>Em (Volts)</th>
<th>SOC 1</th>
<th>SOC 0.9</th>
<th>SOC 0.8</th>
<th>...</th>
<th>SOC 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>5°C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20°C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40°C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Start with Simulation
Battery Cell ↔ Large Battery Pack

Cell Dynamics

Thermal Model

MATLAB EXPO 2019
Evaluate System Behavior

MATLAB EXPO 2019
Discharge + Charge

Cell Voltages

Temperatures

Pack Current

SOC
Discharge Only

Cell Voltages

Cell Temperatures

Pack Current

SOC
Charge Only

Cell Voltages

Temperatures

Pack Current

SOC
Evaluate SOC Estimation

Coulomb counting

EKF
UKF

UKF_EKF
SOC_EKF
Cell_Voltage
Temp
UKF_EKF_1TC
SOC_UKF
Current
Temp
SOC
Curr

MATLAB EXPO 2019
Generate C/C++ Code From BMS Algorithm Models

Demo
Did we generate code too early? Is this ready to ship?

What if there are bugs?

Where are they? How do we find them?

Let's Find Out
Why Testing, Verification and Validation

Safety Critical System

Functional Safety Certification
Typical Development Workflow

Most errors introduced

Unit test finds some errors

Errors found during integration or in field

Requirements → Specification

C/C++/HDL

MATLAB EXPO 2019
Challenge: Errors introduced early but found late

Most errors introduced

Unit test finds some errors

Errors found during integration or in field

Requirements → Specification → C/C++/HDL → Field
Validate Design via Desktop Simulation

- Requirements
- Design Model

Unit test finds some errors
Errors found during integration or in field
Model-Based Design Workflow

Requirements

Design Model

Model used for production code generation

Simulink Models

C/C++/HDL

Generated code

Code Generation

MATLAB EXPO 2019
Model-Based Design Verification Workflow

**Model Verification**
*Discover design errors at design time*

- Component and system testing
- Review and static analysis

**Code Verification**
*Gain confidence in the code*

- Equivalence testing
- Equivalence checking

---

**Simulink Models**

**Requirements** → **Design Model** → **Model used for production code generation** → **C/C++/HDL** → **Generated code**

- ✔ Faster development
- ✔ Reduced cost
- ✔ Less hassle
- ✔ More Engineering
How do we test?

Is it functioning correctly?

- Requirements
- Design Model
- Model used for production code generation
- Simulink Models
- C/C++/HDL
- Generated code
Systematic Functional Testing with Simulink Test

Test Case

Inputs
- MAT file (input)
  - Scenario
  - Signal Editor
  - Test Sequence
- Excel file (input)

Assessments
- MAT file (baseline)
  - MATLAB Unit Test
  - Assessments
- Excel file (baseline)

and more!

Model Sim through SIL, PIL and HIL
Scale with PCT and CI

Test Harness

Main Model
Create Test Harnesses
Systematic Functional Testing with Simulink Test

Use Test Manager to:
- Author, execute, manage test cases
- Review, export, report
Manage and Organize TestCases
How do we know we have enough test?

- Does the design meet all requirements?
- Is it functioning correctly?
- Is it completely tested?
How do we know we have enough test?

Does the design meet all requirements?

Is it functioning correctly?

Is it completely tested?

Requirements → Design Model → Model used for production code generation → C/C++/HDL → Generated code
Integrate with requirements tools and author requirements

- Import from:
  - Word / Excel
  - IBM® Rational® DOORS®
  - ReqIF™ standard

- Update synchronizes changes from source

- Edit and add further details to import

- Author requirements

- Export ReqIF
  - Enables roundtrip with external tools
Import External Requirements & Author New Requirements
SW Requirements - Implementation Status
# Track Implementation and Verification

<table>
<thead>
<tr>
<th>Index</th>
<th>ID</th>
<th>Summary</th>
<th>Implemented</th>
<th>Verified</th>
</tr>
</thead>
<tbody>
<tr>
<td>crs_req_func_spec</td>
<td>1</td>
<td>Driver Switch Request Handling</td>
<td>Implemented</td>
<td>Passed</td>
</tr>
<tr>
<td></td>
<td>1.1</td>
<td>Switch precedence</td>
<td>Implemented</td>
<td>Passed</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>Avoid repeating commands</td>
<td>Implemented</td>
<td>Failed</td>
</tr>
<tr>
<td></td>
<td>1.3</td>
<td>Long Switch recognition</td>
<td>Implemented</td>
<td>Unexecuted</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>Cancel Switch Detection</td>
<td>Implemented</td>
<td>Passed</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>Set Switch Detection</td>
<td>Implemented</td>
<td>Passed</td>
</tr>
<tr>
<td></td>
<td>1.6</td>
<td>Enable Switch Detection</td>
<td>Implemented</td>
<td>Passed</td>
</tr>
</tbody>
</table>

**Implementation Status**
- Implemented
- Justified
- Missing

**Verification Status**
- Passed
- Failed
- Unexecuted
- Missing
Requirements Verification with Simulink

Requirements
- Driver Switch Request Handling
  - 1.1 Switch precedence
  - 1.2 Avoid repeating commands

Implemented By
Simulink / Stateflow

Verified By
Test Harness

Test Case
- Inputs
  - Scenario Signal Editor
  - MAT / Excel file (input)
  - Test Sequence

Test Harness
- Simulink Test

Assessments
- MAT / Excel File (baseline)
- Test Assessments

MATLAB Unit Test
Link Tests to Requirements
How do we know we have enough test?

Does the design meet all requirements?

Is it functioning correctly?

Is it completely tested?

Simulink Models

Requirements → Executable Specification → Model used for production code generation → C/C++/HDL → Generated code
Coverage Analysis to Measure Testing

- Identify testing gaps
- Missing requirements
- Unintended Functionality
- Design Errors
Measure Model Coverage
Addressing Missing Coverage

Test Cases from Functional Requirements

Partial Coverage

Test Generator

Simulink Design Verifier
Addressing Missing Coverage

New Test Cases

Test Cases

Full Coverage
Automatic Test Generation
Systematic Simulation Testing

Does the design meet all requirements?

Is it functioning correctly?

Is it completely tested?

Simulink Models

Requirements → Design Model → Model used for production code generation → Generated code

C/C++/HDL

Does the design meet all requirements?

Is it functioning correctly?

Is it completely tested?
Real-Time Testing of Battery Management System

Testing BMS with Battery Cells

- Longer test cycles
- Difficult to reproduce results
- Difficult to test fault conditions
- Limited test automation
Hardware-In-Loop Testing of Battery Management System

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

Battery Emulation
Automatic Code Generation
Wiring and Signal Conditioning
Measurement & Diagnostics
Main Controller

MATLAB EXPO 2019
Challenges in setting up HIL testing for BMS

Execute Large Battery Models in Real-time
- Cell Voltages: ~100Hz
- Current: ~1kHz (Accurate Coulomb Counting & current derating calculations)
- Cell / Pack Temperatures: ~1 to 10Hz (Large time constant for thermal system)

Sensor and Fault Emulation
- Produce Isolated Voltages
- Sink and Source Current
- Support Series and Parallel Configuration
- Temperature simulation
Execute Large Battery Models in Real-time

Prepare Model for Real-time Simulation (if using Simscape)
- Use Simscape Local Solver
- Configure fixed-cost simulations
- Try “Partitioning” option for non-linear systems*

Reducing model complexity
- Select right variant of battery block to match desired model fidelity
- Reduce order of charge dynamics by selecting fewer number of time-constants

Webinar on “Real-Time Simulation of Physical Systems Using Simscape”
Sensor and Fault Emulation

Battery Emulation (IO991-06)
- 6 independent isolated channels
- Architecture allows series and parallel stack combinations
- Voltage range of 0-7 V with 14-bit resolution
- Source 300mA and sink 100 mA

Temperature Sensor Emulation
- RTD (PT100, PT1000) – low resolution 6 channel module
- Thermocouple – accurate 32 channels
- High precision resistor simulation modules

Fault Insertion
- 1A to 40A current handling; Up to 4 fault buses
- Electromechanical or MOSFET switches
Conclusion

Leverage models to communicate technical specifications, design implementation, results and maintain traceability.

Test your design iterations every step of the way through simulations and Hardware-In-Loop testing.

Gain confidence in design and work towards safety certification.
Learn More about Battery Management System

**Developing Battery Management Systems with Simulink and Model-Based Design**

https://www.mathworks.com/discovery/battery-models.html

**Battery Modeling**

Model batteries when designing battery-powered systems

**Technical Articles and Newsletters**

Modeling and Simulating Battery Performance for Design Optimization

By Cecilia Wang, Romeo Power

**File Exchange**

Design and Test Lithium Ion Battery Management Algorithms

version 1.0.1 (65 MB) by Ching SHAW

**Battery Modeling**

Examples and How To

- Battery Management System Development in Simulink (7.17) - Video
- Lithium Battery Model with Thermal Effects for System-Level Analysis (24.05) - Video
- Automating Battery Model Parameter Estimation using Experimental Data (25.28) - Video
- Real-Time Simulation of Battery Packs Using Multicore Computers (22.57) - Video
- Battery Simulation and Controls - Consulting Services
- Sifting Through Multisource Data for Safer Battery Materials with Machine Learning - Article

**Papers**

- High Fidelity Electrical Model with Thermal Dependence for Characterization and Simulation of High Power Lithium Battery Cells - IEEE 2012
- Battery Model Parameter Estimation Using a Layered Technique - SAE 2013
- Simplified Extended Kalman Filter Observer for Battery SOC Estimation - SAE 2013
- Battery Pack Modeling, Simulation, and Deployment on a Multicore Real Time Target - SAE 2014
- Model-Based Parameter Identification of Healthy and Aged Li-ion Batteries for Electric Vehicle Applications - SAE 2015

Download Link to File Exchange

For more info:
Chirag Patel
cpatel@mathworks.com

THANK YOU