

Developing Battery Systems with Simulink and Simscape

Battery System Development Workflow

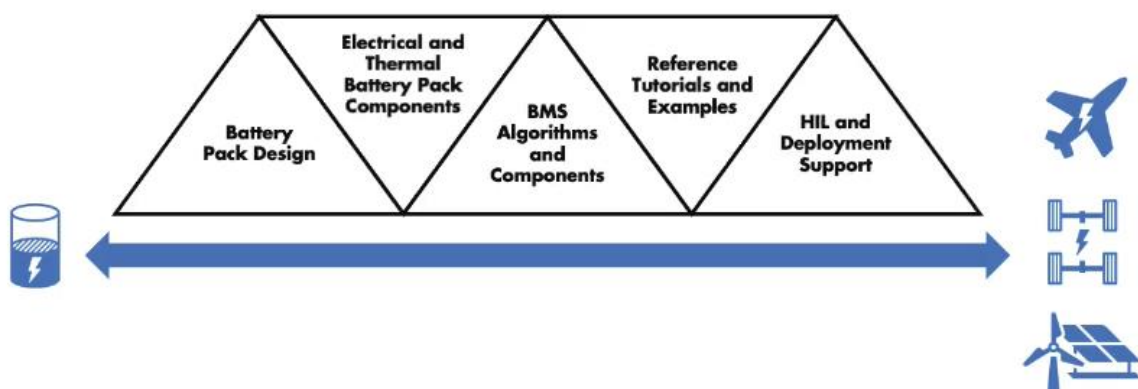
Electrification is driving the use of batteries for a range of applications, including electric vehicles (e.g., cars, buses), ships, electric aircraft, grid-tied energy storage systems, and photovoltaic systems. These applications have different requirements for battery system design in terms of cell selection, power/energy density, volume, weight, and lifetime.

Simulation of the battery system design before testing provides insight into the dynamic behavior of the battery pack. It also lets you:

- Explore and compare software algorithms.
- Expand operational test cases.
- Shorten the technology development cycle from battery cell to battery system.

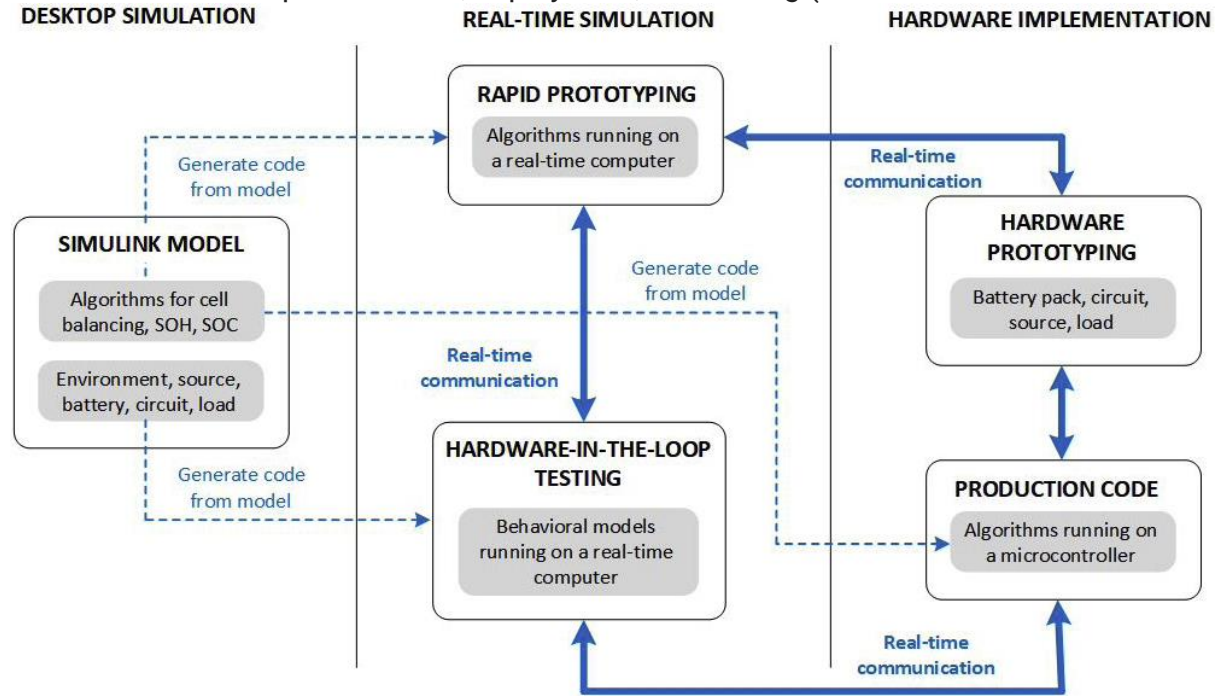
The workflow for battery system development begins with building the battery cell. Five major tasks build a bridge from battery cell design to a battery system. Those steps include:

- Battery pack design
- Electrical and thermal battery pack component design
- Battery management system (BMS) algorithm development
- Integrating components to run desktop simulations
- Hardware-in-the-loop (HIL) testing and deployment



Building a bridge between battery cell and battery system.

Using Simulink® and Simscape™, the battery system development workflow begins with integration of the system components so that you can perform desktop simulation to validate the component designs and algorithms (see [Desktop Simulation](#)). The next step is real-time simulation of models using rapid prototyping and hardware-in-the-loop testing (see [Real-Time Simulation of Battery Systems](#)). The final stages of development involve hardware implementation, deployment, and testing (see [Hardware](#)

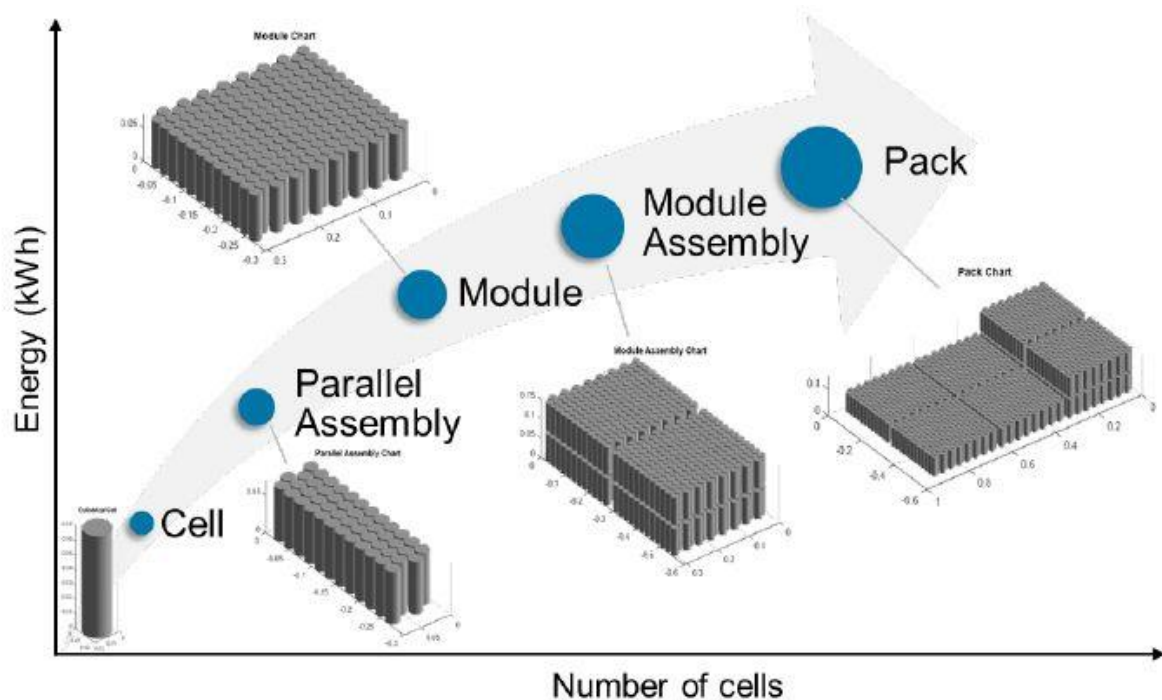


[Implementation](#)).

Battery system development workflow with Simulink and Simscape.

Battery Pack Design

Using the [Simscape Battery™](#) application programming interface (API) in MATLAB®, you can [design a battery pack](#). Foundational elements of the design include cell design, parallel assembly, module, module assembly, and battery pack design.



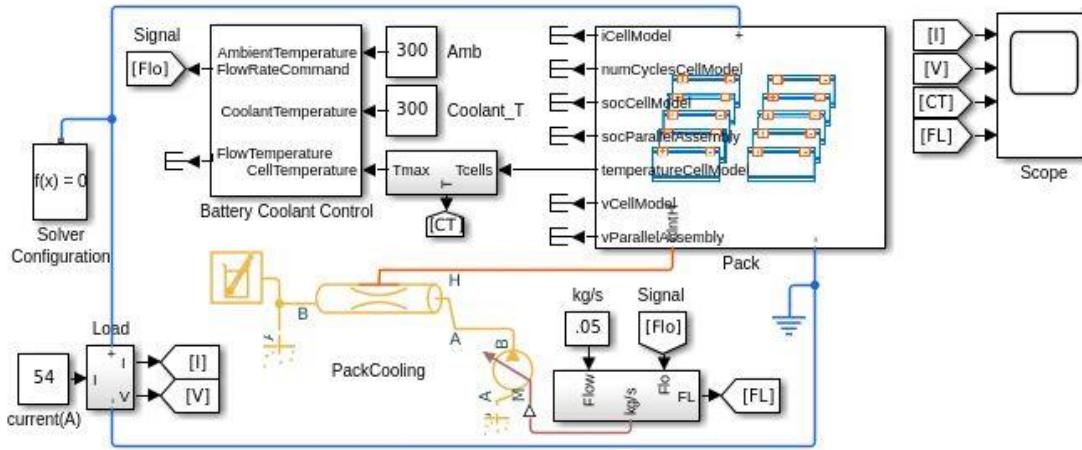
Battery pack design from cell to pack.

Using Simscape Battery, you can:

- Model electrothermal behavior and include charge dynamics, aging, thermal, and heat transfer effects in battery cell models.
- Parameterize cells based on manufacturer data sheets.
- Build and visualize battery models with different geometries and topologies, from cell to module and from module to pack.
- Model cooling plates with customizable fluid paths and thermal connections to the battery pack.
- Explore cell-to-cell temperature variation and measure cooling efficiency.
- Generate a custom Simulink library model for your battery pack design.
- Set a suitable model resolution to strike a balance between model fidelity and simulation speed.

Battery Thermal Management System

Engineers can use MATLAB and Simulink to design a battery thermal management system to regulate battery pack temperature within specifications and ensure it delivers optimal performance for a variety of operating conditions.

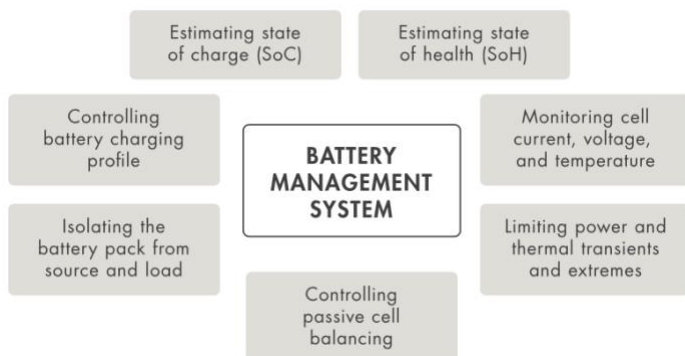


[Thermal analysis comparison](#) of a new and aged lithium-ion battery using Simscape Battery.

Battery Management System Algorithms

A well-designed battery management system (BMS) ensures maximum performance, safe operation, and optimal lifespan under diverse charge-discharge and environmental conditions. Simulink and Simscape enable you to gain insight into the dynamic behavior of the battery pack, explore software architectures, test operational cases, and begin hardware testing early, reducing design errors. Engineers can use built-in BMS control blocks in Simscape Battery to evaluate the designed battery pack performance, develop a thermal management system, and run system-level simulations.

To meet these objectives, the BMS consists of algorithms that control the behavior and performance of the battery pack.



Functions of the battery management system.

Desktop Simulation

Desktop simulation in Simulink enable you to verify functional aspects of the battery system design. On the desktop, the battery system, environment, and algorithms are simulated using behavioral models. For example, you can explore active versus passive cell balancing configurations and algorithms to evaluate the suitability of each balancing approach for a given application. You can use desktop simulation to explore new design ideas and test multiple system architectures before committing to a hardware prototype. You can also perform requirements testing in desktop simulations; for example, you can verify that contactors are prevented from opening or closing when an isolation fault is detected.

Real-Time Simulation of Battery Systems

Once validated via desktop simulation, Simulink models can be used to generate C and HDL code for rapid prototyping (RP) or hardware-in-the-loop (HIL) testing to further validate the BMS algorithms in real time. With RP, instead of handwriting control software code for real-time testing, you generate code from your controller model and deploy it to a real-time computer that performs the functions of the production microcontroller. With automatic code generation, algorithm changes made in the model can be tested on real-time hardware in hours rather than days. Further, you can interact with real-time control hardware from within Simulink to change algorithm parameters and log test data.

As with rapid prototyping, HIL testing involves generating code from a Simulink model and deploying it to a real-time computer. In the case of HIL testing, code is generated from the battery system models rather than the control algorithm models, providing a virtual real-time environment that represents battery pack, active and passive circuit elements, loads, charger, and other system components. This virtual environment lets you validate the functionality of the BMS controller in real time before developing a hardware prototype and in an environment where hardware will not be damaged.

Tests developed during desktop simulation can be carried over to HIL testing to ensure that requirements are met as the BMS design progresses. Though HIL testing is employed primarily to test code running on a microcontroller or FPGA, you can instead use a rapid prototyping system, such as [Simulink Real-Time™](#) and [Speedgoat® target hardware](#), and connect it to the HIL setup before production controller hardware is selected.

Hardware Implementation

In the hardware implementation stage, the Simulink control models that have been verified via desktop simulation, RP, and HIL are used to generate efficient, production-ready code for the BMS. If necessary, production code generation can be incorporated into workflows compliant with formal certification standards used in the automotive, aerospace, and other regulated industries.