

白皮书

使用 Simulink 和基于模型的设计开发电池 管理系统

各行各业对电池储能的依赖性越来越大,使得电池管理系统 (BMS) 的重要性日渐凸显。该系统能够确保实现电池最大化性能、安全运行,以及在各种反复充放电和环境条件下的最长使用寿命。为了设计出满足这些目标的 BMS,工程师们开发了具有以下功能的**反馈和调度控制算法**:

- 监测电芯电压和温度
- 估计荷电状态 (SOC) 和健康状态 (SOH)
- 限制功率输入和输出,以实现散热和过充保护
- 控制电池充电曲线
- 平衡各个电芯的荷电状态
- 必要时将电源和电池隔离开

本白皮书将介绍工程师们如何使用 Simulink® 进行系统级仿真来开发 BMS 算法和软件。通过使用 Simulink 进行基于模型的设计,您能够深入了解电池包的动态行为,探索软件结构,测试不同运行工况并尽早开始硬件测试,从而减少设计错误。利用基于模型的设计,BMS 模型可用作所有设计和开发活动的基础,包括对设计的功能方面进行桌面仿真、对照行业标准进行形式化测试与验证,以及生成代码用于实时仿真和硬件实现(图 1)。

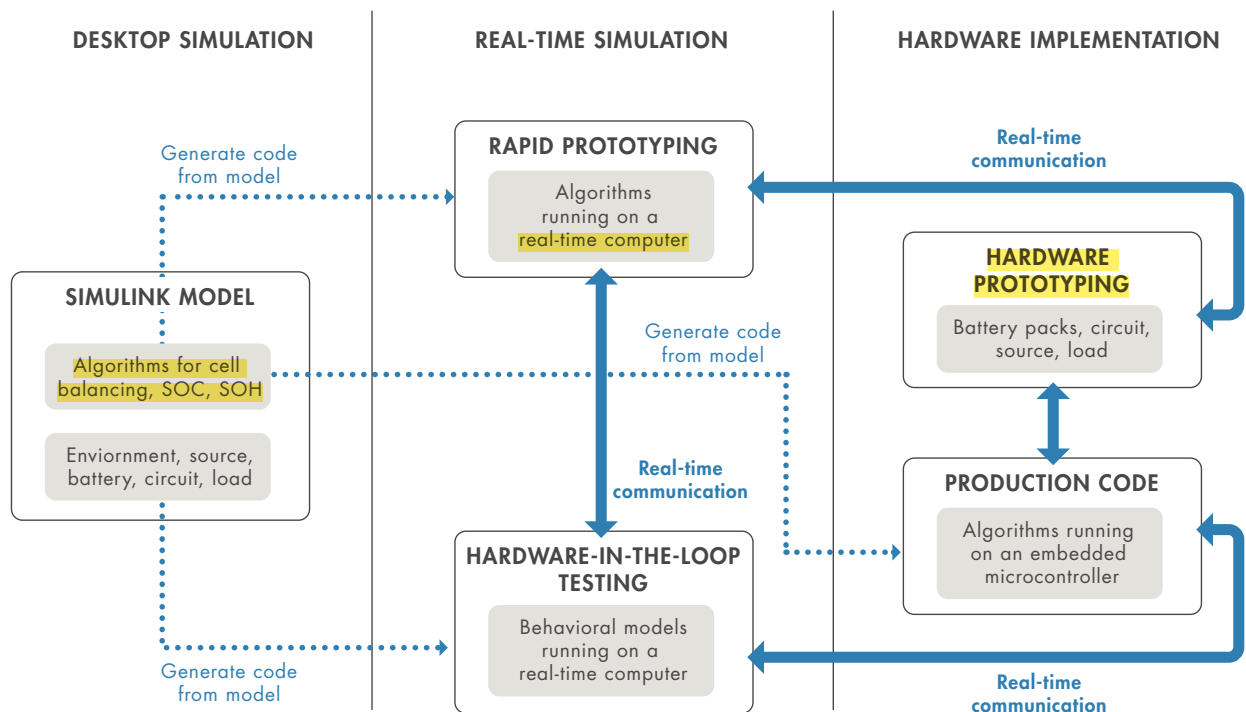


图 1. 使用 Simulink 和基于模型设计的电池管理系统开发工作流程。

桌面仿真。利用 Simulink 中的桌面仿真，您能够验证 BMS 设计的功能方面，例如充放电行为（使用单电芯等效电路公式）、电子电路设计、反馈和调度控制算法。在桌面上，使用行为模型对电池系统、环境和算法进行仿真。例如，您可以探索主动式与被动式电芯平衡配置和算法，评估每种平衡方法对于给定应用场景的适用性。您可以使用桌面仿真探索新的设计构想，并在应用到硬件原型之前测试多个系统架构。您还可以在桌面仿真中执行需求测试，例如通过验证在检测到隔离故障时会阻止接触器打开或关闭。

实时仿真。通过仿真验证 Simulink 模型后，可以使用这些模型生成 C 和 HDL 代码进行快速原型 (RP) 设计或硬件在环 (HIL) 测试，在实时环境中进一步验证 BMS 算法。利用 RP，您不需要手工编写用于实时测试的控制软件代码，而是从您的控制器模型生成代码，然后将代码部署代替控制器功能的实时计算机。利用自动代码生成，在模型中进行的算法更改可以在实时硬件上进行测试，数小时即可完成整个过程。您还可以在 Simulink 内与实时控制硬件交互，更改算法参数并记录测试数据。

与快速原型一样，HIL 测试也涉及从 Simulink 模型生成代码并将代码部署到实时计算机。就 HIL 测试来说，区别在于是从电池系统模型，而不是从控制算法模型生成代码，提供了一个代表电池包、有源和无源电路元件、负载、充电器和其他系统组件的虚拟实时环境。此虚拟环境让您能够在开发硬件原型之前实时验证 BMS 控制器的功能，而且是在不会造成硬件损坏的环境中。桌面仿真过程中开发的测试可以延展到 HIL 测试，以确保随着 BMS 设计的推进而始终符合要求。虽然 HIL 测试的主要用途是测试微控制器或 FPGA 上运行的代码，但是您也可以在选择生产控制器硬件之前，改用连接到 HIL 设置的快速原型系统，如 *Simulink Real-Time™* 和 *Speedgoat 目标硬件*。

硬件实现。在硬件实现阶段，使用已通过桌面仿真、RP 和 HIL 验证的 Simulink 控制模型，生成用于 BMS 的为生产准备就绪的高效代码。如有必要，可将产品级代码生成步骤合并到汽车、航空航天和其他受监管行业中使用的符合正式认证标准的工作流程中。

“借助基于模型的设计，我们建立了从构思到产品级代码的一套集成开发流程。MathWorks 工具使我们能够运用自身的专业技术开发关键的电池管理技术，为我们提供了一种便于早期以及持续验证设计的环境。”

— 东风电动汽车公司刘小康博士

» [阅读案例](#)

桌面仿真: BMS 软件建模

真实的仿真能力是开发 BMS 控制软件的核心环节,而这从精确的**电池包模型**开始。电池的设计通常使用有限元分析 (FEA) 模型来仿真物理属性并捕获其电-热化学属性。虽然这些模型非常适合设计和优化电池包的化学和几何结构,但是控制工程师需要更加适合系统级设计和软件开发的模型。

“我们的许多初始电池模型都是凭经验设计的,有理想的电压源和固定阻抗。现在我们使用比较复杂的第一性原理模型,因此对电池这种电化学装置获得了非常有价值的见解。我们使用 Simulink 构建了高级等效电路模型,可以在不同的荷电状态、放电速率、温度和老化级别上预测性能。”

“我们使用类似的方法执行了安全关键型仿真,预测电池内部的冷却性能,确保电池包不会过热。要捕获所有多域物理、化学和热传导效应,通常需要有限元分析工具和很大工作量。借助 MathWorks 工具,我们执行分析并获得深刻见解,大大促进了电池技术的发展。”

— 特斯拉汽车公司 Chris Gadda 博士和 Andrew Simpson 博士

» [阅读文章](#)

电池电芯建模与特征化

在 Simulink 中开发 BMS 算法时,您可以使用等效电路模拟电池电芯的热电行为。等效电路通常由一个电压源、一个串联电阻和一个或多个并联电阻电容对组成(图 2)。电压源提供开路电压,而其他组件对电芯的内阻和时间依赖行为建模。这些等效电路元件通常依赖于温度和荷电状态(SOC)。因为这种依赖性对于每个电池的化学原理来说是独一无二的,所以需要根据对电池单元的实验来获取,该电池单元的类型必须与为其设计控制器的电池相同。您可以使用 Simulink 和 MATLAB® 中的优化,通过 [实验数据模型校准](#)实现等效电路参数化。

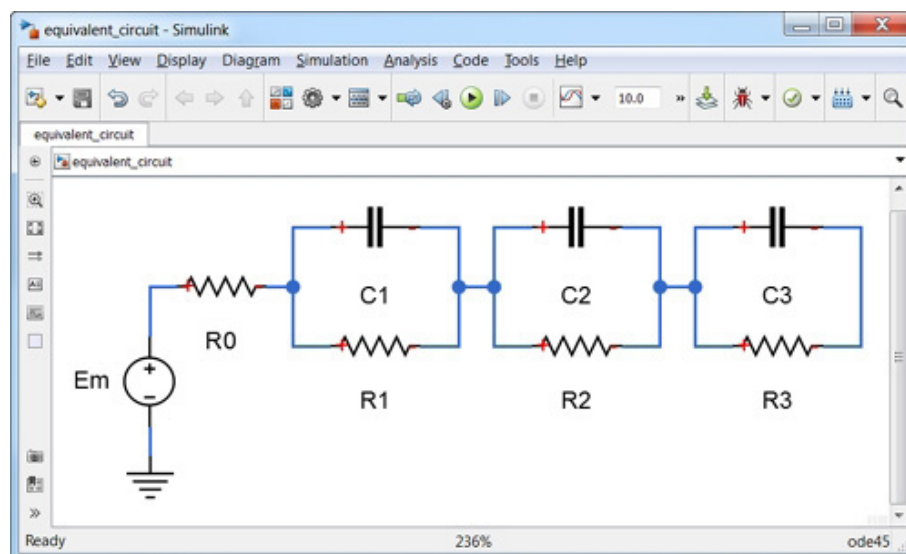


图 2. 带有三个不同时间常数、内阻和开路电压的电池的等效电路。

通过集成 *Simscape Electrical™* 组件，您可以从电芯级扩展到模块和电池包级，直观地将电芯与围绕电路相结合。借助电池包级仿真，您可以评估各种电池包配置对储能容量、电力输送率和散热的影响。电池包级仿真还让您能够考察电池包与其他系统组件（如电源、负载和保护电路）的交互。

了解有关电池电芯建模与特征化的更多信息

- [锂电池电芯 -- 两个 RC 分支等效电路](#) - 示例
- [锂离子温度依赖电池模型](#) - 示例
- [电池模型](#) - 文件交换
- [可反复充电电池模型](#)的参数化 - 示例
- [自动估计电池模型参数](#) (9:55) - 视频
- [使用分层技术估计电池模型参数：锂离子磷酸盐电池的使用实例](#) - 技术论文

对电力电子器件和无源组件建模

除了电池包模型以外，真实 BMS 仿真还需要连接电池系统与电源和负载的电路组件的精确模型。Simscape Electrical 是 Simulink 的一个附加产品，提供构建完整电池系统电路所需的有源和无源元件（如用于电芯平衡的电路）的完整库。充电源可由直流电源组成，比如光伏 (PV) 系统，也可由 经过整流 的交流电源组成。

使用 Simulink 进行系统级仿真时，您可以围绕电池构造复杂的充电源，在各种操作范围和故障条件下验证 BMS。可采用类似的方法对电池包负载进行建模和仿真。例如，电池包可能通过逆变器连接到电动汽车 (EV) 中的永磁同步电动机 (PMSM)。利用仿真，您可以通过循环工况法改变 EV 的运行方式，评估 BMS 能否有效应对不断变化的工作条件。

了解更多关于电力电子器件和无源组件建模的信息

- [高电压电池通过受控的三相逆变器给 IPMSM 供电 - 示例](#)

开发调度控制算法

开发 BMS 调度控制算法的工程师使用状态机对调度逻辑建模，实现故障检测和管理、充电和放电功率限制、温度控制和电芯平衡（图 3）。Stateflow® 是一个基于状态机和流程图来构建组合和时序逻辑决策模型并进行仿真的环境。在为 BMS 开发调度控制算法时，您可以使用 Stateflow 对电池系统响应事件、基于时间的条件以及外部输入信号的方式进行建模。例如，在恒流恒压 (CCCV) 充电的情况下，您可以开发并测试状态逻辑，控制电芯何时从电流充电模式变换到电压充电模式。

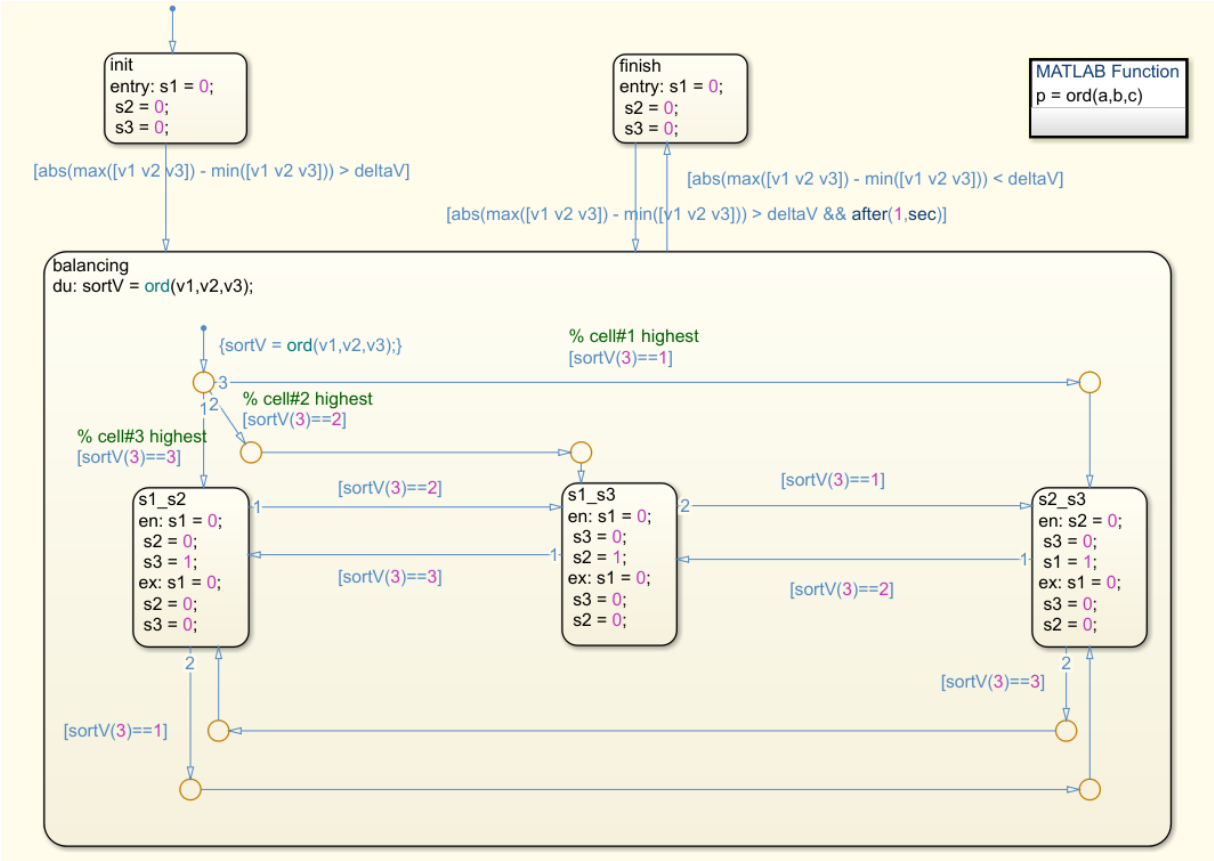


图 3. 在 Stateflow 图中实现的电芯平衡逻辑。

了解更多关于开发调度控制算法的信息

- [Simulink 中的电池管理系统开发 \(7:18\)](#) - 视频

估计荷电状态

精确的电池模型对于开发用于估计 SOC 的算法至关重要。开路电压 (OCV) 测量和电流积分 (库仑计数) 等传统 SOC 估计方法在某些情况下相当精确。但是, 估计具有平面 OCV-SOC 放电特征的现代化学电池的 SOC 需要使用不同的方法。扩展卡尔曼滤波 (EKF) 就是这样一种方法, 实践表明, 它可以在合理的计算工作量前提下提供精确的结果。Simulink 包含一个 EKF 模块, 使您能够开发用于估计 SOC 的观察器。此类观察器通常包括一个非线性系统 (电池) 的模型和一个递归算法。该模型使用从电芯测得的电流和电压作为输入, 该算法根据一个两步式预测/更新进程来计算系统的内部状态 (其中包括 SOC) (图 4)。

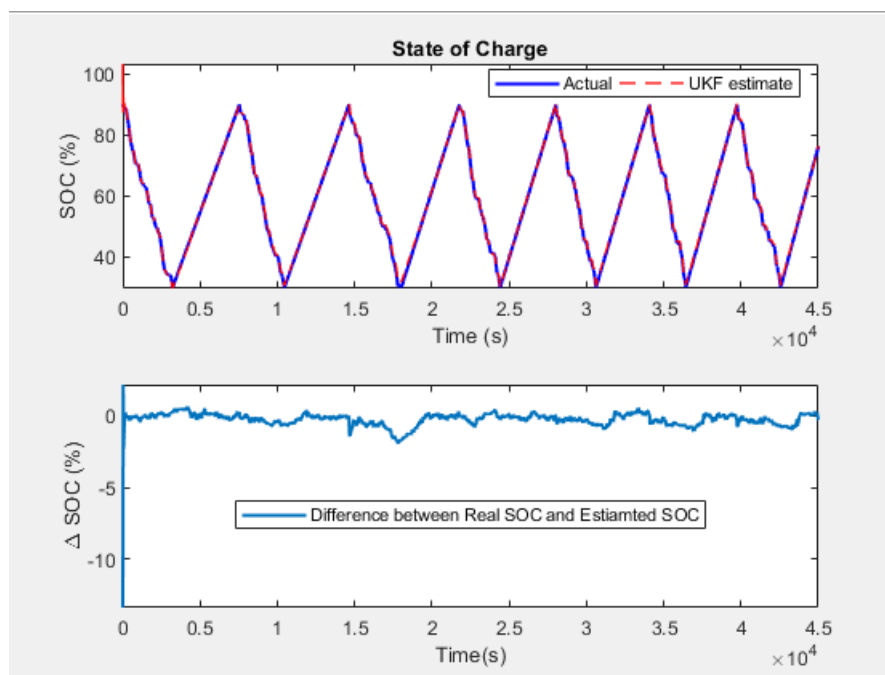


图 4. 使用无味卡尔曼滤波器估计电池荷电状态。

了解更多关于估计荷电状态的信息

- [基于扩展卡尔曼滤波器模型的荷电状态 \(SoC\) 评估 - 文章](#)
- [电池管理系统参考设计 - Intel 产品文档](#)
- [电池老化系统的非线性状态估计 - 示例](#)
- [扩展卡尔曼滤波器 - 产品文档](#)

估计健康状态

所有电池, 包括在出厂时符合性能规格的电池, 由于使用寿命和充放电循环, 性能会随着时间而降低, 表现为储能会逐渐损失, 同时内部电阻增加。使用短时间测量方法进行内部电阻估测是一个相对直接的方法, 而通常需要完全充电或放电过程才能准确计算, 但这种方法有时并不可行。此项挑战导致健康状态 (SOH) 估计以及 EKF 公式的开发越来越受关注, 这类方法现已不只包括状态, 还包括电池参数。精确估计瞬时内部电阻对于 BMS 建立功率限制非常有帮助。

SOH 估计比 SOC 估计更主观; 对于如何定义 SOH 尚无共识。因此, 每个组织可能有其自己特定的量化 SOH 估计的方法, 因而不可能使用通用的现成解决方案。利用 Simulink, 您可以开发和仿真自定义 SOH 估计算法, 与您的组织特定的电池健康状况的解释保持一致。

了解更多关于估计健康状态的信息

- [为电动汽车应用中健康和老化的锂电池进行基于模型的参数识别](#) - 技术论文

桌面仿真测试

将桌面仿真集成到您的测试程序后, 您可以编写并执行测试用例, 运用 BMS 检验所有可能的逻辑分支和闭环控制 — 进行硬件测试时少见的一种覆盖级别。采用这种方法, 仿真模型充当推动 BMS 设计和测试的可执行规范 (图 5)。Simulink 有一系列的功能, 支持通过桌面仿真进行测试, 使您能够:

- 将限制、容差、逻辑检查和时态条件等需求集成到模型中, 并追溯到原始规范
- 构造复杂的基于仿真的测试序列, 以便执行功能、基准、等价和背对背测试
- 跟踪行业标准指标, 如决策、条件和修改的条件/决策覆盖率 (MC/DC) 以及关系边界覆盖率
- 生成测试输入, 以实现全面的模型覆盖和自定义目标
- 使用形式化方法, 识别导致整数溢出、死逻辑和被零除的隐藏设计错误

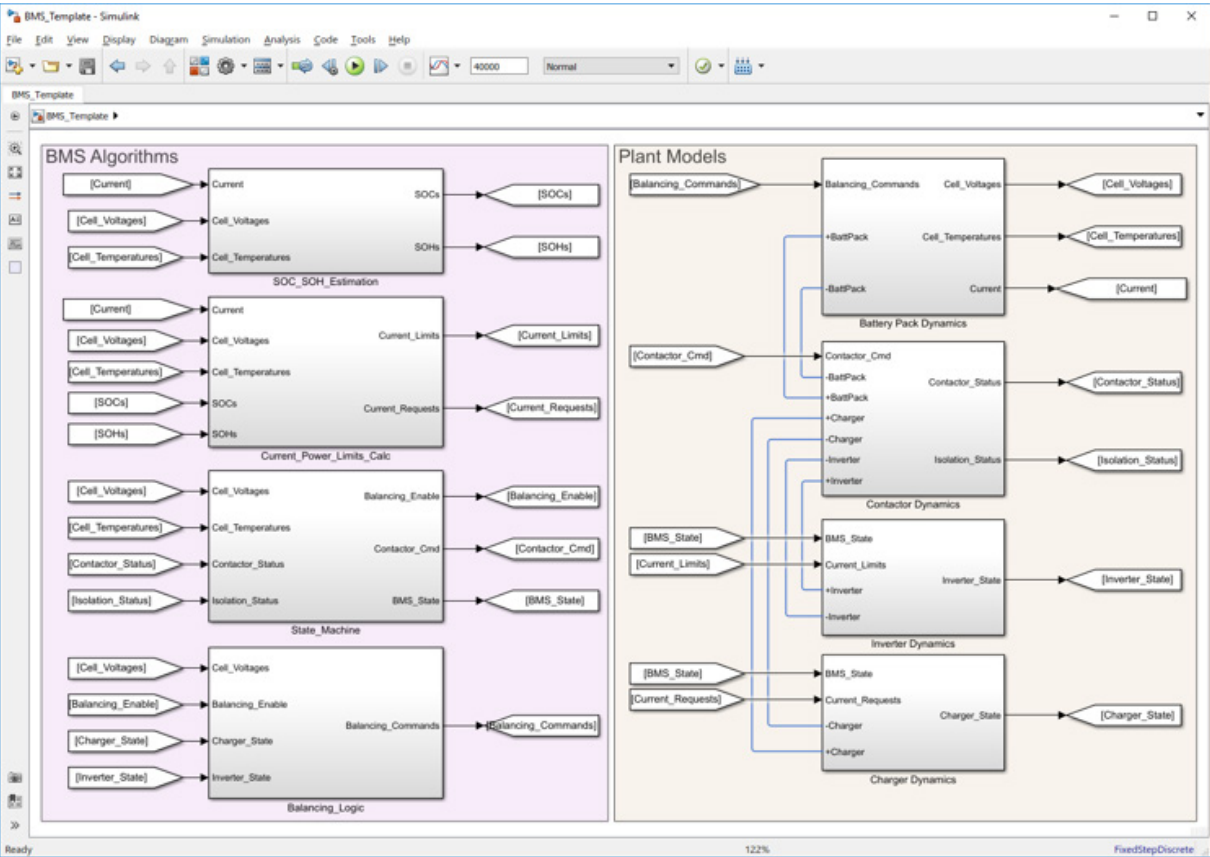


图 5. 在 Simulink 中创建的 BMS 算法和对象动态模型，包括电池包、接触器、逆变器和充电器。

当开发中的电池系统必须符合安全要求时，您可以将基于形式化方法的测试集成到符合 IEC 61508、IEC 61851 和 ISO 26262 等标准的软件开发流程中。

“在 Simulink 中开发控制器模型时，我们经常使用 Model Advisor 检查是否符合风格指南和建模标准。我们还使用 Simulink Design Verifier™ 检查我们的模型中是否存在死逻辑、被零除错误和其他设计错误。”

— LG 化学 Duck Young Kim、Won Tae Joe 和 Hojin Lee

» [阅读文章](#)

了解更多关于桌面仿真测试的信息

- 使用基于模型的设计为混合动力汽车电池管理系统开发符合 AUTOSAR 和 ISO 26262 标准的软件 - 文章
- 使用基于模型的设计开发 Tesla Roadster - 文章

实时仿真：验证 BMS 软件

实时仿真 (包含快速原型设计和 HIL 测试) 让电力电子控制工程师们能更加深入的了解 BMS 设计如何在硬件上执行。无论是在 RP 还是在 HIL 中, 目标都是在硬件中仿真总体设计的一个方面: 在 RP 中仿真 BMS 控制器, 在 HIL 中仿真电池系统的平衡。在 BMS 设计中, 实时仿真提供了几大优势, 让您能够:

- 在选择最终控制器硬件之前进行 RP, 开始执行验证算法
- 利用实时测试系统的灵活性进行快速设计迭代和测试
- 在获得电池系统原型硬件之前进行 HIL 测试
- 使用 RP 和 HIL 测试的组合, 测试BMS算法和设计硬件, 进行设计硬件无法进行的测试, 避免昂过硬件损坏 (图6)

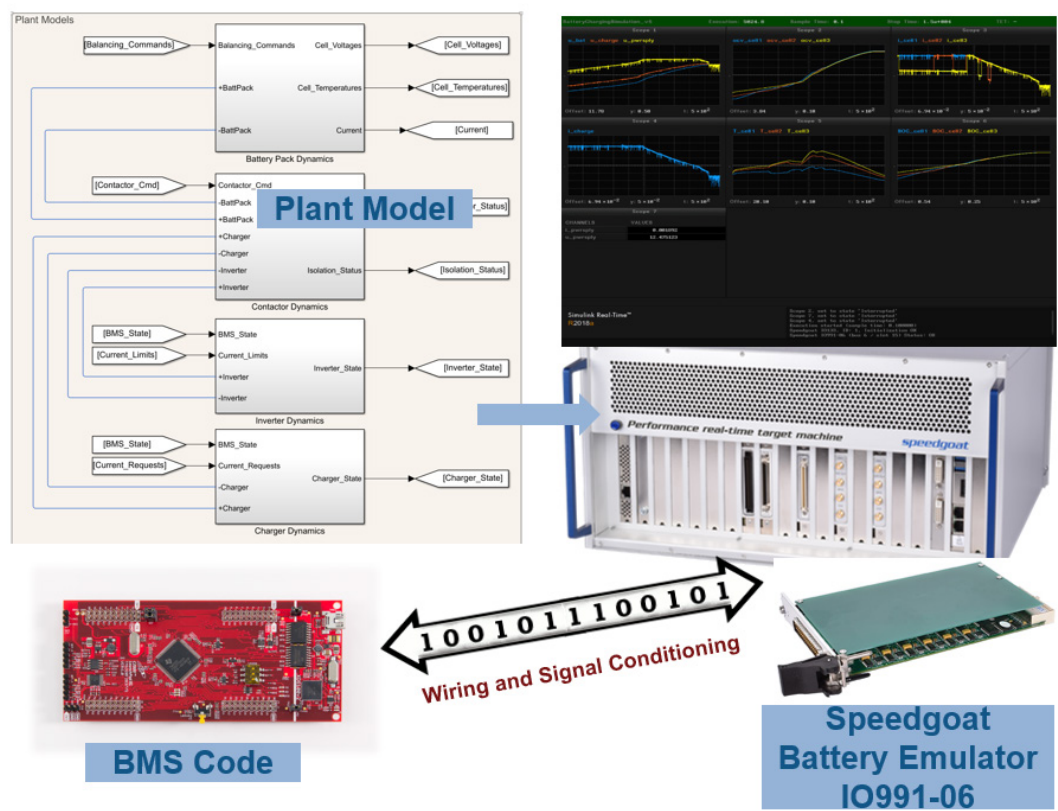


图 6. 电池管理系统软件的硬件在环 (HIL) 测试。从在 Simulink 中创建的 BMS 算法模型生成 BMS 代码并部署到 Texas Instruments C2000 微控制器。对象模型 (电池包、接触器、逆变器、充电器) 在 Simulink 中建模。生成并部署代码, 在配有电池模拟器的 Speedgoat 实时计算机上运行。

通过在 Simulink 中重复利用桌面仿真模型为实时仿真生成代码，可以缩短总体开发时间。您可以生成 C/C++ 和 HDL 代码，在专为实时性能而优化的计算机上执行。从 Simulink 模型生成用于实时仿真的代码包括一些接口，使您在实时仿真运行时能够调整控制参数。

执行快速原型设计

在硬件测试过程中，对控制器代码进行更改可能造成延迟并增加风险。手动修改代码、重新编译，然后将其部署到微处理器或 FPGA 也需要花费很长时间，如果您是控制算法开发者，依赖软件或硬件工程师做出更改，则可能需要更长时间。根据所需更改的程度，还面临向实现的代码中引入新问题的风险。

替代手工编写代码的控制器软件更新，您可以使用 Simulink 生成代码，在专用计算机上实时执行并使用高速 I/O 与测试硬件进行通信。除了消除手动编码及其相关的延迟以外，采用此 RP 方法的另一个优势是，先在桌面上运行仿真模型来验证对 BMS 软件的更改，确认没有引入其他问题。

硬件在环测试

因为构建和修改电池系统的硬件原型可能需要相当大的工作量，而且维修的成本通常很高，所以，测试此类原型在电池包所在的电子系统中的运行情况并不总是可行。由于这些限制，即使很小的设计改动也可能威胁到开发计划，往往造成 BMS 设计进展缓慢，因为团队认为严重偏离以前的设计太过冒险。

利用 Simulink，您可以从电池系统以及它所属的更大系统（包括电源和负载）中的硬件的模型生成 C/C++ 和 HDL 代码。将此代码部署到实时计算机后，您可以在电池系统原型中测试控制器之前先对您的控制器代码运行硬件实时仿真。因此，您可以提前找到并纠正控制设计错误，从而避免损坏昂贵且难以更换的硬件原型。您还可以发现硬件设计错误，例如不正确的零部件选型。

有很多集成了电池模拟器的 HIL 实时系统，包括 Speedgoat 目标硬件，使您能够模拟便携式电池电源，模拟电动汽车的电池组，或灌入电流来模拟充电中的电池。

“我认为 Speedgoat 为 Simulink 开发了即插即用的实时平台。对我们来说，这就意味着有更多时间测试我们的控制系统，减少开发 HIL 测试平台的时间。”

— Proterra 控制工程师 Joaquin Reyes

» [阅读案例](#)

了解更多关于使用硬件在环测试电池管理系统的信息

- [Speedgoat 的电池管理系统实时解决方案](#) - 概述

生成产品级代码

Simulink 从控制器模型生成可读、简练而高效的 C/C++ 和 HDL 代码，为在微处理器、FPGA 和 ASIC 上部署做好准备。与为 RP 生成的代码不同，为生产用途生成的代码不包括支持实时监控、参数调节和数据记录所需的额外接口。利用优化设置，您可以精确控制生成的函数、文件和数据，提高代码效率，并有助于与已有代码、数据类型和标定参数集成。

执行处理器在环仿真

在处理器在环 (PIL) 仿真中，C/C++ 或 HDL 代码在微控制器或 FPGA 上运行，与此同时，设备使用 BMS 硬件的 Simulink 模型一步步地执行，降低了在 BMS 代码的初始评估过程中损坏硬件原型的风险。PIL 仿真虽然不是实时仿真，但却是以真实代码运行的，可让您在各种条件下验证您的代码，并有信心在实际系统上一经部署便可正确执行。

生成产品级代码

桌面仿真、RP、HIL 和 PIL 仿真都能让您验证和确认 BMS 的控制算法。利用 Simulink，您可以使用这些相同的算法作为基础，生成可付诸生产的代码 — 用于在微控制器上实施的优化且稳定的 C/C++ 代码，或用于 FPGA 编程或 ASIC 实现的可综合 HDL 代码。自动代码生成消除了手动算法转译错误，并且产生的 C/C++ 和 HDL 代码与 Simulink 中已验证的算法具有同等效果。通过尽可能完整的状态模拟和故障条件下验证仿真您的控制算法，即使无法面面俱到，也能让您对生成的代码在实时系统中处理同样的条件充满信心。如果以后的硬件测试表明需要更改算法，您只需在您的模型中修改算法，重新运行仿真测试用例来验证更改的正确性，然后生成经过更新的新代码。所有生成的 C/C++ 和 HDL 代码都完全可移植，并具有一系列选项对其进行优化，Simulink 模型与代码双向可追踪 (图 7)。

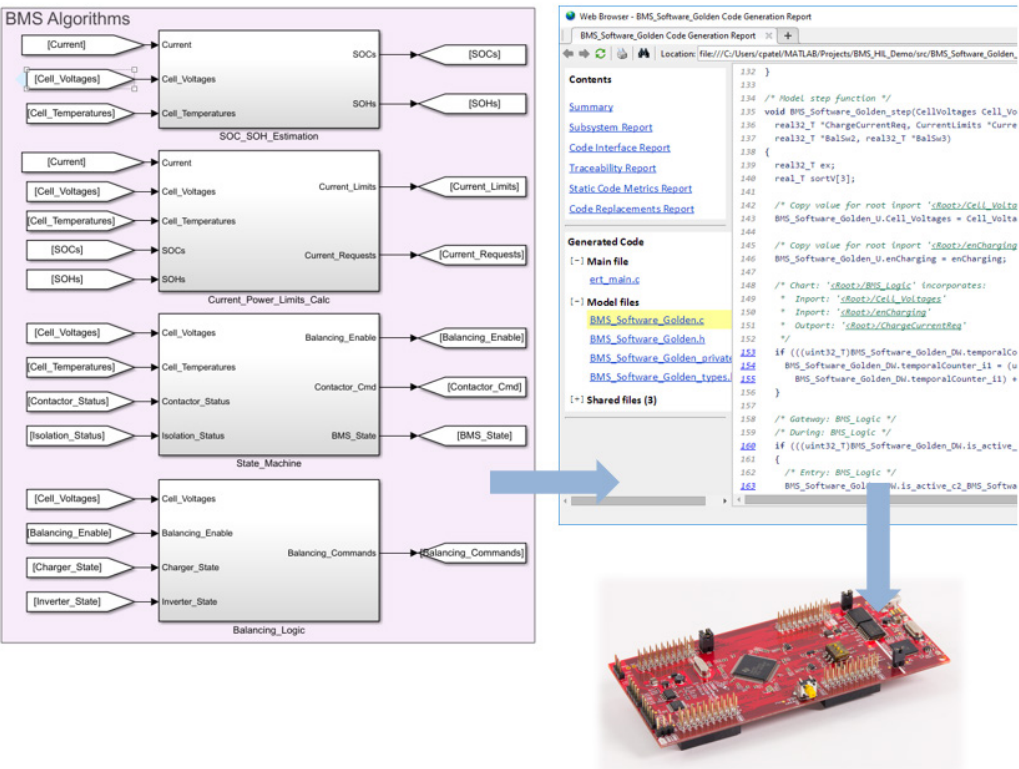


图 7. 从 Simulink 中创建的 BMS 算法模型自动生成 BMS 产品级代码。
代码部署到 Texas Instruments C2000 微控制器。

“在真正的电路上进行验证之前，我们在 Simulink 中通过仿真检查认证所需的每一项要求，因为我们使用 Embedded Coder 直接从我们的模型生成了代码，因此，我们的仿真和真正的嵌入式软件之间没有二义性。”

— 村田制作所 Yue Ma 博士

» 阅读案例

了解更多关于生成产品级代码的信息

- 通过基于模型的设计，村田制作所将能源管理系统控制软件的开发时间缩短了 50% 以上 - 用户案例
- 使用基于模型的设计为混合动力汽车电池管理系统开发符合 AUTOSAR 和 ISO 26262 标准的软件 - 文章

后续步骤

采取后续步骤, 加快您的电池管理系统项目。

浏览: [Simulink 中的电池管理系统](#)

下载: [电机和电力控制试用软件](#)

开始使用: [咨询服务](#)