

白皮书

# 对电力电子器件控制设计进行硬件在环测试

采用 Simulink 和 Speedgoat 实时目标机

## 简介

对混合动力和电动汽车的需求不断增长,可再生能源的发展,以及对更高效、更高性能电机的跨行业需求,正在推升对更复杂的电力电子控制系统以及为其提供支持的嵌入式软件的需求。宽带隙半导体材料的采用带来了更快的开关频率,但也加剧了为这些系统开发嵌入式软件的挑战。为了应对这些挑战,工程师们使用系统级仿真,利用代表有源和无源电路组件的电气行为的被控对象模型,评估闭环系统和控制算法调度的性能。

本白皮书重点介绍在工程实践 workflow 中,在系统级仿真之后、提交最终硬件设计之前执行的一个重要步骤:使用硬件在环(HIL)仿真进行实时测试。对于基于电力电子的系统,如太阳能逆变器或电机驱动设备,HIL 仿真用于电气系统的确定性模型(包括电源和负载)实时验证该系统的数字控制器。本白皮书中贯穿引用了 *Simulink*<sup>®</sup> 软件和 *Speedgoat* 实时硬件系统来说明 HIL 仿真的各个方面。*Simulink* 是用于控制系统系统级仿真的桌面软件,能够生成可用于实时测试的 C 和 HDL 代码。*Speedgoat* 实时硬件系统是使用 *Simulink Real-Time*<sup>™</sup> 运行从 *Simulink* 模型生成的代码的计算机。

## 从桌面仿真到 HIL 仿真

使用系统级模块图模型在 *Simulink* 中执行电力电子控制设计时,您可以构建以多个采样率运行的多级反馈和调度控制算法。若需对要控制的电力电子系统建模,您可以通过添加 *Simscape Electrical*<sup>™</sup> 中的组件,该产品基于 *Simulink* 构建,带有预置电源、负载和电路元件库(图 1)。您可以修改这些组件,根据第一原理或测试数据构建自己的组件,或从其他仿真软件导入设计结构。根据您的设计目标,您可以有选择地提高模型保真度以改善仿真精度,或降低保真度以提高仿真速度。

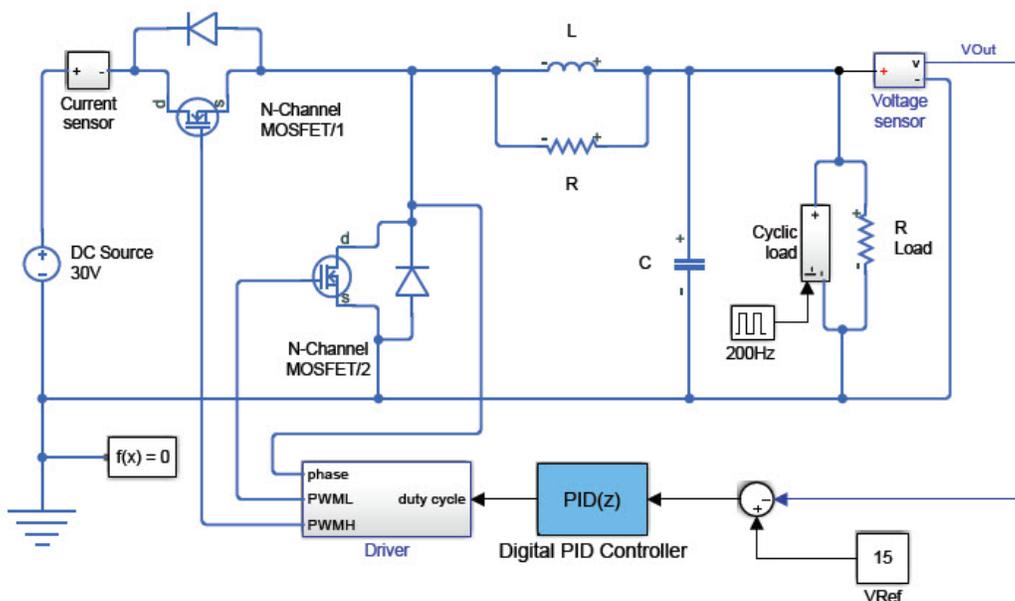


图 1. Buck 变换器的 Simulink 模型。

“对我们使用 *Simulink* 和 *Simscape Electrical* 构建的被控对象模型进行仿真，使我们能够在最终完成硬件设计之前对设计进行改进和优化。仿真免除了硬件预样件的多次迭代。”

— Jakub Vonkomer 博士, VONSCH

» 阅读案例

在台式计算机上进行系统级仿真，是验证电力电子设备的交互，以及在各种情况下（包括可能损坏电力电子设备的故障情况）下控制这些设备的算法的绝佳方法。但是，它对于在正常和异常情况下验证在实际控制器上运行的嵌入式软件的实时行为帮助很小。

验证控制器实时行为的其中一种方案是为控制算法手写代码、编译代码，并将其部署到微控制器，然后将控制器作为实际电力电子系统的一部分进行测试。尽管应该坚持执行此类控制器验证，但不建议将所有实时测试推迟到完成软硬件集成之后的项目开发后期。众所周知，在后期阶段发现的缺陷更难补救，而且补救代价更高。

第二种方案是在实时机上运行系统模型（被控对象）的环境下验证控制器。这种 HIL 仿真可以看作是将桌面仿真扩展到实时仿真，能够在电力电子原型硬件可以进行测试之前执行。

从包含被控对象和控制器的系统模型开始，从描述被控对象的模型部分生成 C 或 HDL 代码（图 2）。从被控对象模型生成的代码在实时计算机上部署运行，该计算机可以足够快的速度执行代码，来仿真实际被控对象硬件的实时行为。例如，来自 *Speedgoat* 的实时目标机器使用 *Simulink Real-Time* 执行从 *Simulink* 生成的代码。利用系统的这种实时表示，您可以测试在微控制器、FPGA 或其他实时测试系统上运行的嵌入式控制软件。借助 HIL 仿真，您还可以注入故障来评估控制器在不利条件下保持最佳和安全运行的能力。

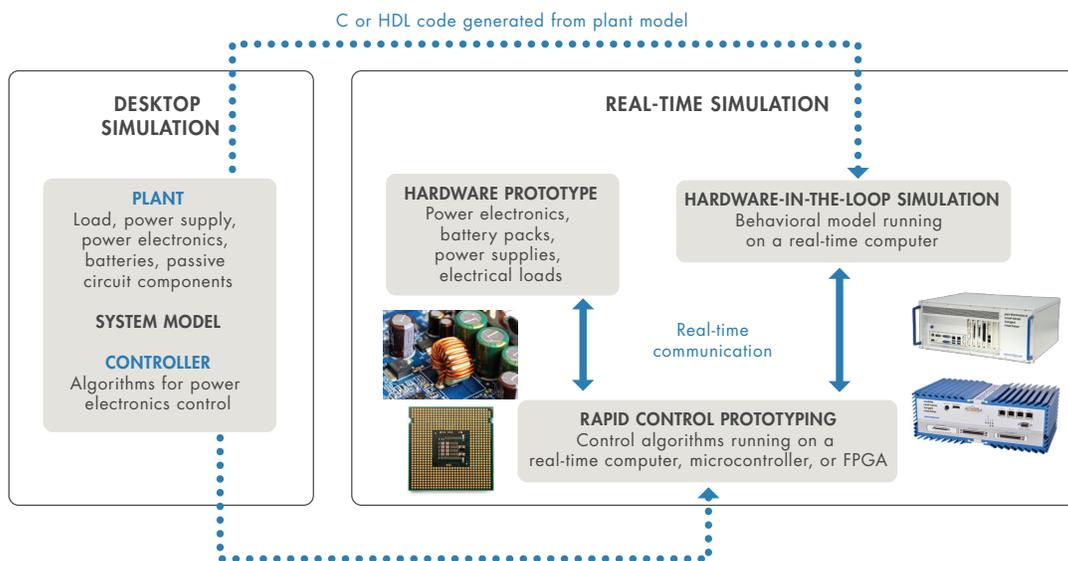


图 2. 使用从桌面仿真模型生成的代码进行快速原型和 HIL 仿真。

“Simulink、Simulink Real-Time 和 Speedgoat 目标硬件让我们能够证明我们的早期设计对于特定应用的可靠性，而无需在真正的飞行器上安装这些单元。使用基于模型设计，我们能够进行持续的验证和确认，而无需等到电力电子控制单元的所有开发全部完成。”

— Shane O’Donnell, Microsemi

» [阅读案例](#)

在为电力电子系统开发 HIL 仿真时，请在项目初期花些时间考虑以下因素：

- 仿真速度与模型复杂度之间的权衡
- 处理器和延迟要求

这些因素将影响您如何进行电力电子控制系统建模，以及选择哪种实时测试硬件来验证控制器性能。

### 权衡仿真速度和模型复杂度

最开始，请考虑为了确保系统稳定、安全地运行，控制器必须达到的响应速度。图 3 显示了一系列物理系统及其各自的响应时间；请注意，电力电子控制器通常必须以近似微秒或更快的速度单位进行采样。但是，您的 HIL 仿真所需的速度取决于您需要在系统模型中仿真的动态特性。

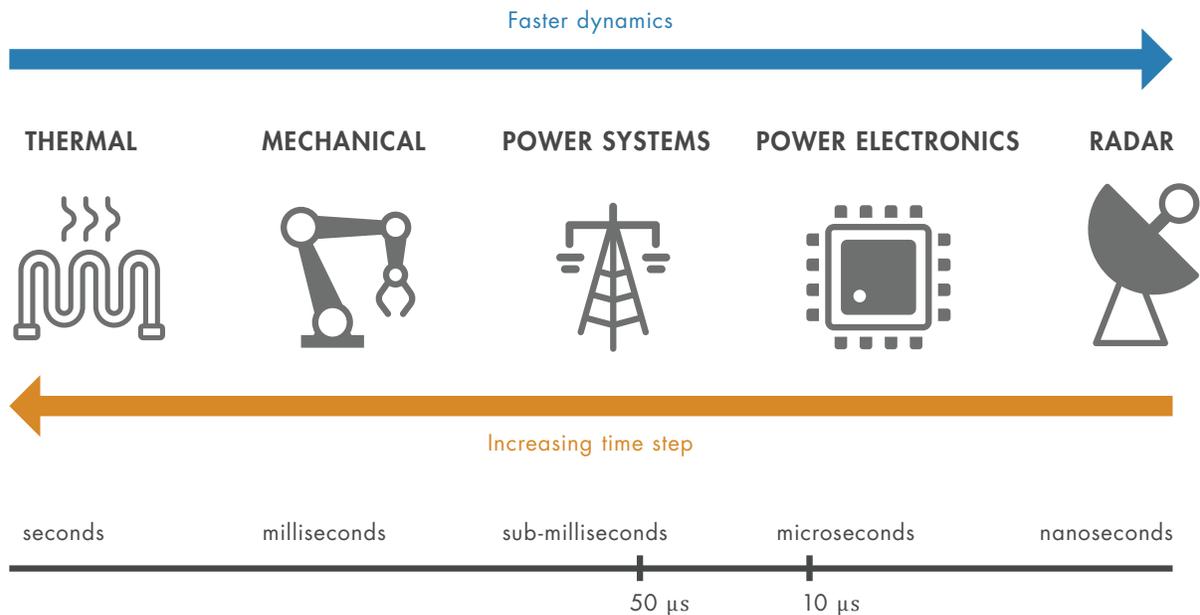


图 3. 动力学与时间步的关系。

通常，更简单的动态特性所描述模型的仿真速度更快，但是准确性不够高。具有更复杂动态特性的模型的仿真速度更慢，但是能更准确地代表真实电力电子器件行为。使用 Simulink，您可以[从大量的固定步长或可变步长仿真求解器中进行选择](#)。对于桌面仿真，可变步长求解器会调整步长时间以确保准确性。在动态特性更简单的模型中，可变步长可能更大，从而使模型的计算速度更快。

例如，在设计用于永磁同步电机 (PMSM) 的矢量控制算法时，可以将逆变器电力电子器件的电压输出建模为功率晶体管工作周期的平均值电压。可以将电机建模为具有集总参数的一组线性微分方程。此仿真不会捕获开关特性的影响，但适用于在正常运行条件下调整控制算法和测试控制。因为它使用更简单的数学方式表示电力电子器件行为和 PMSM，所以仿真的运行速度要比基于反映电力电子器件的详细切换和 PMSM 的非线性模型的仿真运行速度更快。

现在，想象一下 PMSM 用于电动汽车的动力总成，其中所有的电损耗都会消耗电池组的电荷并减少续航里程。在这种情况下，您可能希望逆变器模型包括电力电子器件的开关特性以及温度对电效率的影响。另外，您还可以包括从有限元分析 (FEA) 软件 (如 [JMAG](#)) 导入的 PMSM 的非线性模型，该模型可以捕获转子和定子之间的磁性饱和和空间谐波。从数学角度看，此被控对象模型比包含线性电机模型和平均值逆变器的被控对象模型更加复杂。因此，桌面仿真的运行速度会更慢，求解器将使用小步长来捕获模型中描述的更详细的动态特性。

运行桌面仿真并获得令人满意的结果后，下一步是实时仿真。这是指从被控对象模型生成可执行代码并将其部署到目标计算机。对于实时测试，Simulink 可以通过 [Simulink Coder™](#) 生成 C 代码，通过 [HDL Coder™](#) 生成 HDL 代码。代码与硬件无关，可以在任何实时硬件上部署。[将 Simulink Real-Time 与 Speedgoat 硬件一起使用](#)可以简化实施 C 或 HDL 代码的过程，并让实施以及自动化实时测试变得更加简单。

*“Speedgoat 目标机可以在恶劣的电磁环境 ( $E = 1V/m$ ) 中提供对开关半导体的快速、鲁棒的控制。”*

— Piotr Dworakowski, SuperGrid 电源转换器团队负责人

» [阅读案例](#)

## 满足处理器和延迟要求

系统动态特性的复杂性不仅会影响桌面仿真的运行速度，还会影响您对 HIL 仿真的硬件选择。实时仿真的模型时间步长必须大于实时硬件的系统延迟，才能避免发生任务超限错误 (图 4)。系统延迟取决于两个因素：

- 处理器上运行的应用程序产生的软件延迟
- 输入或输出接口产生的硬件延迟

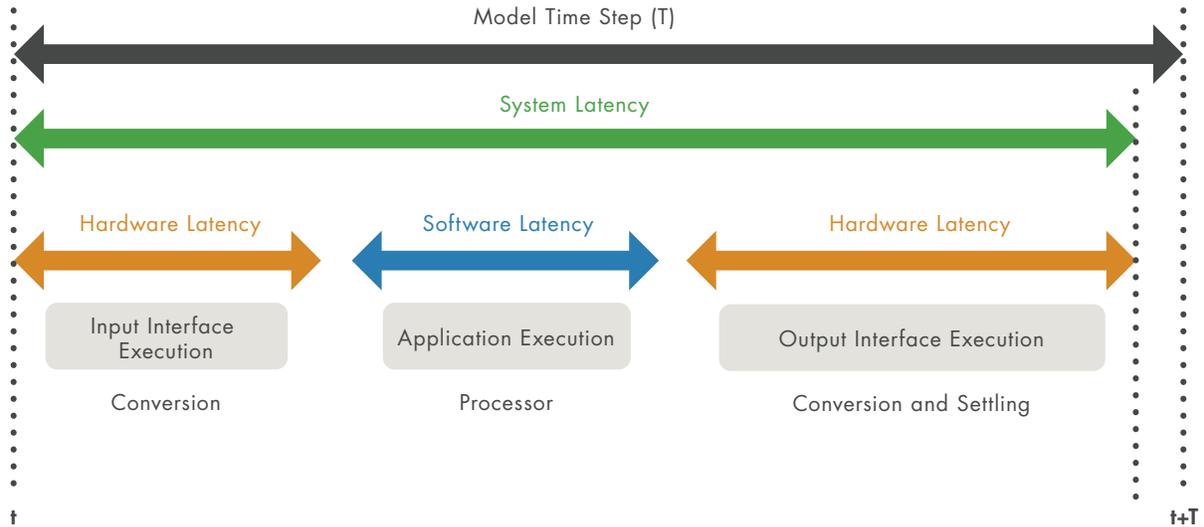


图 4. 导致系统延迟的因素。

软件延迟受应用程序的计算复杂度和所使用处理器的计算能力影响。您可以采用更快的 CPU 或 FPGA 并应用并发执行（在多个 CPU 核上运行任务）来减少软件延迟。将多核处理器与 FPGA 结合使用，您可以对系统进行分区，以便在 CPU 上执行低动态特性和数据记录的子系统，而在 FPGA 上运行复杂动态特性和信号调节的子系统。

考虑一下涉及电力电子器件开关特性的上一个 PMSM 案例。出于谨慎，您希望 HIL 仿真以比实际系统的预期开关速度快 100 倍的采样频率运行。例如，如果 PMSM 将以 10 kHz 的开关频率运行，HIL 仿真则需要以 1 MHz 或更快的频率运行，以捕获由开关器件造成的影响。对于需要以此频率运行的 HIL 系统，您可以考虑使用基于 FPGA 的实时系统来准确表示电子器件的开关行为。虽然经验表明，小于 50 微秒 (20 kHz) 的时间步可能需要使用 FPGA 硬件，但根据模型的复杂度和 I/O 延迟情况，有时也可以使用低至 10 微秒 (100 kHz) 的基于 CPU 的仿真。决定使用 CPU 还是 FPGA 取决于模型的数学复杂度和 I/O 信道的数量和类型。对于电力电子器件的 HIL 仿真，许多组织正在转向使用基于 FPGA 的系统来仿真开关系统。通过在电力电子系统中使用碳化硅 (SiC) 和氮化镓 (GaN) 半导体，以更快的控制系统时钟速度实现更高的运行频率，这一趋势正在得到强化。

大多数实时硬件供应商，包括 Speedgoat 在内，都提供基于 FPGA 和基于 CPU 两种系统。借助 Speedgoat 和 Simulink Real-Time，您可以对模型进行分区，在多个 CPU 核（通过**并发执行**）和 FPGA 上分别运行不同部分。

硬件延迟取决于所使用的 I/O 信号。数字 I/O 的延迟往往低于模拟 I/O。在 FPGA 上，数字 I/O 的延迟可以保持在一纳秒以内。相比之下，对于基于 CPU 的系统，数字 I/O 在微秒范围内。使用模拟信号和通信协议时，硬件延迟取决于模数转换器 (ADC) 技术和信号触发选项，对于采用顺序采样的 ADC，还取决于被测信道的数量。您会看到，每秒的采集速率从数千到数百万个样本不等。

硬件延迟还受处理器与其 I/O 信道之间的通信影响。在 CPU 上运行的全开关逆变器和非线性电机模型的仿真，通过总线（如 PCI Express）与其 I/O 通讯，将以大约 20 微秒或更大时间步长执行，具体取决于信道数量。其中部分时间步长是由与 PCI Express 总线关联的硬件延迟引起的。相比之下，在具有板载可配置 I/O 的 FPGA 上运行的相同应用程序可消除总线延迟，并将以大约 1 微秒的时间步长执行。另一种比较是，将多个 FPGA 与低延迟总线（如 Aurora64B/66B 串行通信）连接，增加了它们之间约 1 微秒的通信延迟。

将输入/输出和应用程序安排为并行执行也可以降低延迟。例如，通过让模型执行、数据采集和信号写入并行执行，将直接内存存取 (DMA) 用于模拟输入和输出，可以明显降低系统延迟（图 5）。

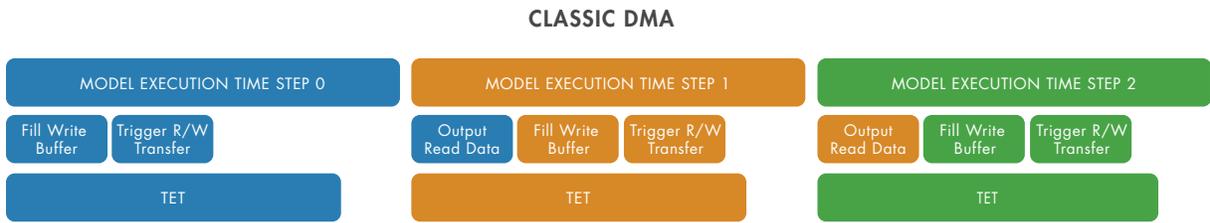


图 5. 使用 DMA 支持模型和 I/O 传输的并行执行。

## 电力 HIL 仿真

到目前为止，所讨论的 HIL 场景关注的都是开发实时模型来验证基于电力电子的系统要使用的控制器。由有源和无源电路、负载和电源组成的被控对象在目标计算机上作为实时模型运行。控制器或被测设备 (DUT) 在正常和故障条件下接受测试，这样便可以在不损坏电力设备的前提下，及早发现大多数控制算法的错误和问题。这种 HIL 仿真类型如图 6 所示，有时称为控制 HIL。在这里，实时系统仿真光伏 (PV) 电池、逆变器电子器件和电网，可以提供与实际电厂相同的模拟、数字和协议接口。

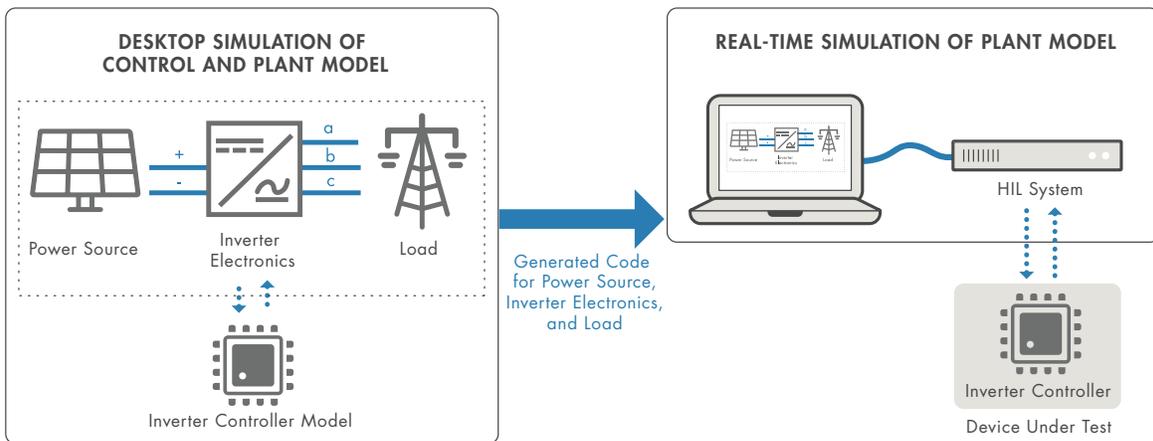


图 6. 使用实时仿真测试控制器的控制 HIL。

与 DUT 就是控制器本身的控制 HIL 相比, 电力 HIL 仿真采用更复杂的 DUT, 其中包含了控制器和硬件组件。在电力 HIL 仿真中, DUT 通过功率放大器连接到实时仿真。

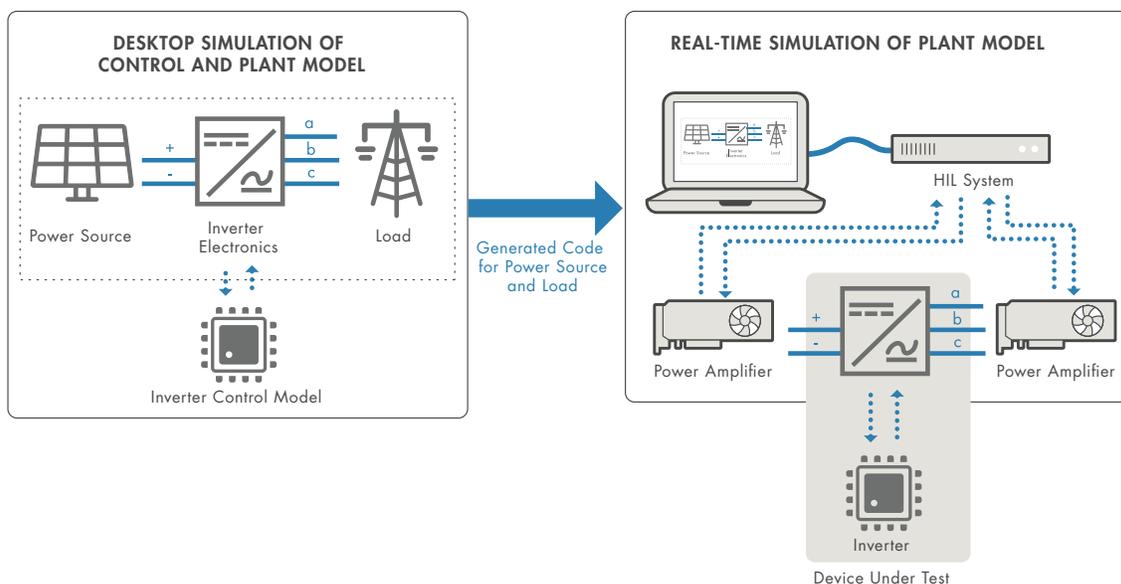


图 7. 通过功率放大器将被测设备连接到实时仿真来实现的电力 HIL 仿真。

在图 7 所示的电力 HIL 仿真示例中, DUT 包含逆变器 (逆变器电子器件和控制器)。逆变器接口是与光伏系统的直流电压连接以及与电网的三相交流电压连接。为了提供必需的直流和交流功率级别, 在 DUT 与实时计算机之间连接了高速交流和直流功率放大器。实时计算机仍然会仿真光伏系统和电网连接, 提供高速信号来驱动直流和交流功率放大器。

其他电力 HIL 仿真应用还包括电力传动系统测试、微电网仿真和电池管理系统测试。

## HIL 工作流程注意事项

在评估电力电子控制设计的 HIL 测试解决方案时, 务必考虑从桌面仿真到实时仿真的整个开发流程。否则, 与执行测试相比, 您最终可能要在如何实现软件与实时硬件产品的配合使用上花费更多时间。需要考虑的另一个重点是代表电力电子控制系统的仿真模型的保真度。例如, 系统级仿真软件必须提供全套的控制设计功能, 用于反馈和调度控制。同样, 电力建模库必须提供可以覆盖不同模型保真度级别的组件。例如, 库中的绝缘栅双极晶体管 (IGBT) 应该具有模型变体, 此变体在光谱的一端代表理想行为, 在另一端代表基于物理的特征。最后, 如果您希望选择将仿真模型部署到各种实时硬件目标, 用于生成可移植代码的仿真软件也至关重要。

MathWorks 和 Speedgoat 共同开发了 Simulink Real-Time, 这是来自 MathWorks 的实时操作系统。通过这次密切合作, 用于桌面仿真的 Simulink 和 Simscape Electrical 模型可以在实时测试中重用, 使用 Speedgoat 实时目标机通过 CPU 和 FPGA 来完成测试。Speedgoat 产品还为电力电子器件提供了广泛的 I/O 连接和通信协议, 例如:

- 高速模拟输入和输出
- 光纤链路
- 高速数字输入
- 通信协议, 包括 EtherCAT、具有灵活数据速率的 CAN 和 PROFINET

有关针对电力电子控制设计 HIL 测试的 MathWorks 和 Speedgoat 解决方案的更多信息, 请[联系销售](#)。