

2024 MathWorks 中国汽车年会

整车大规模仿真的最佳实践

楚骏楠, MathWorks



基于Simulink平台的虚拟车辆仿真环境

车辆建模

软件集成

场景定义

仿真分析

仿真发布

车辆模型模板
控制器及零部件模块库
建模规范

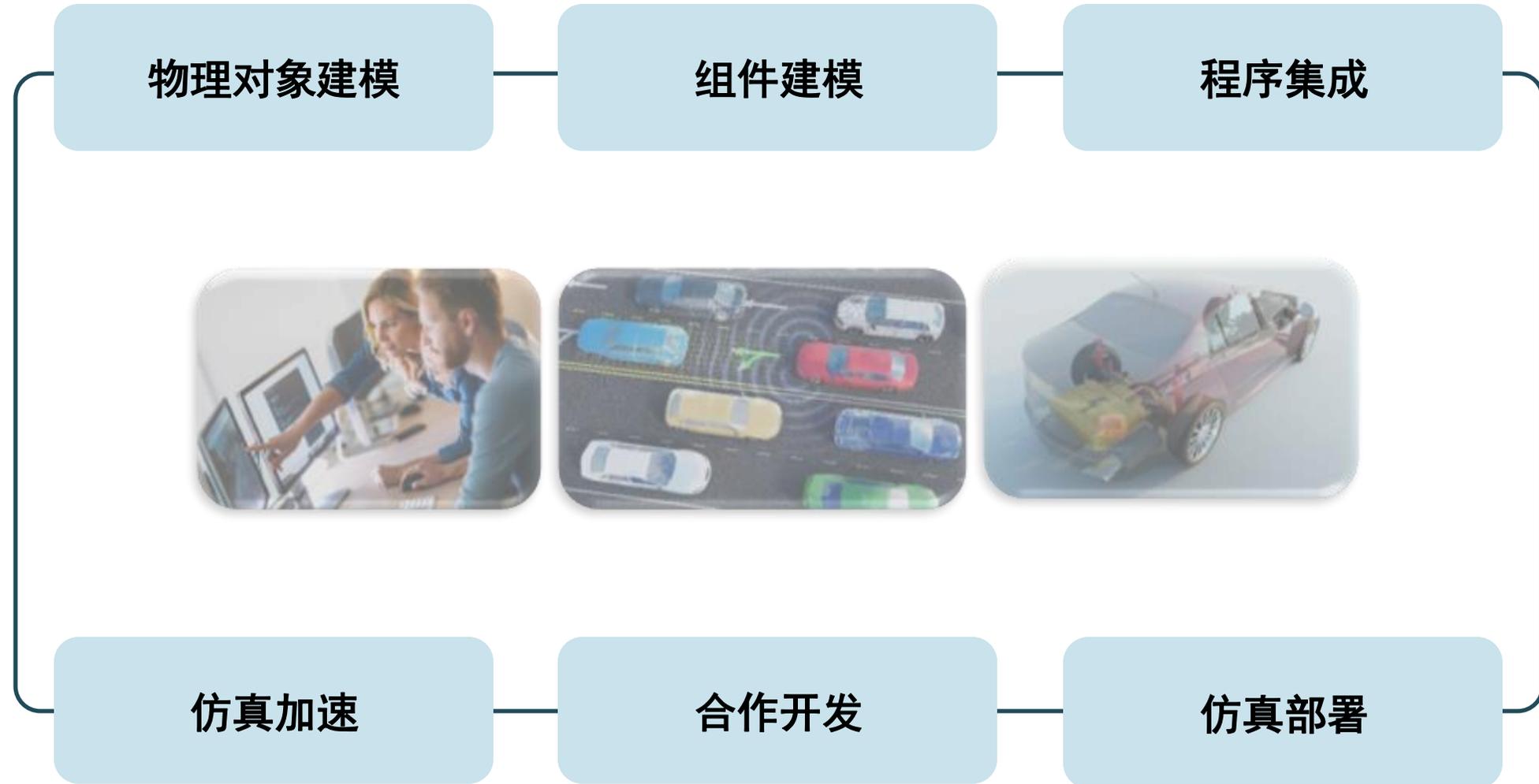
C/C++ 接口
ROM阶模型
FMU 集成

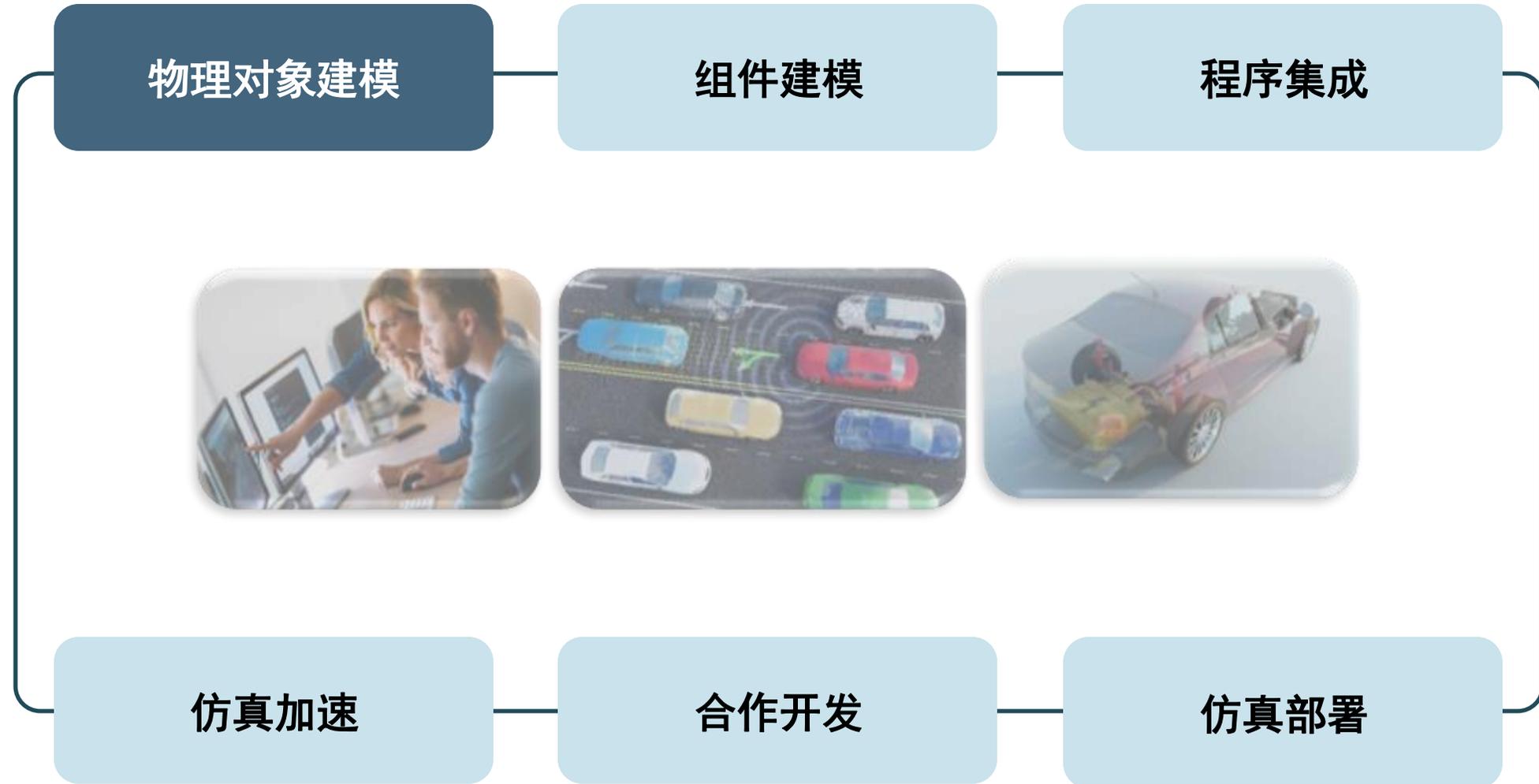
地图和场景
各种开放标准
驾驶循环

结果显示
数据分析
报告生成

云端集成
Datalake 集成
HIL实施



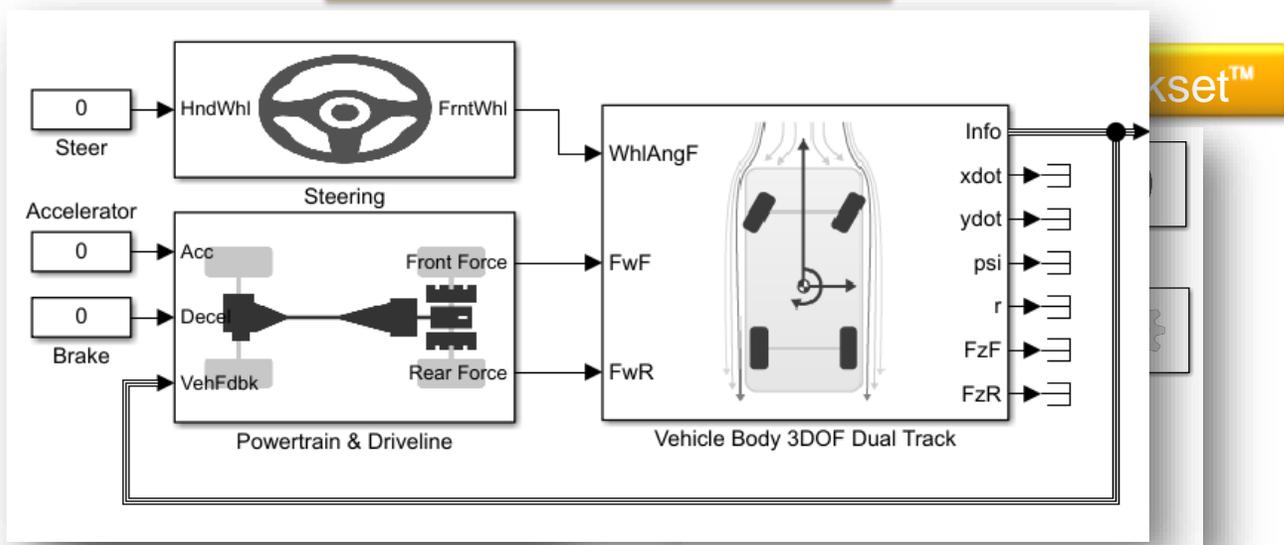




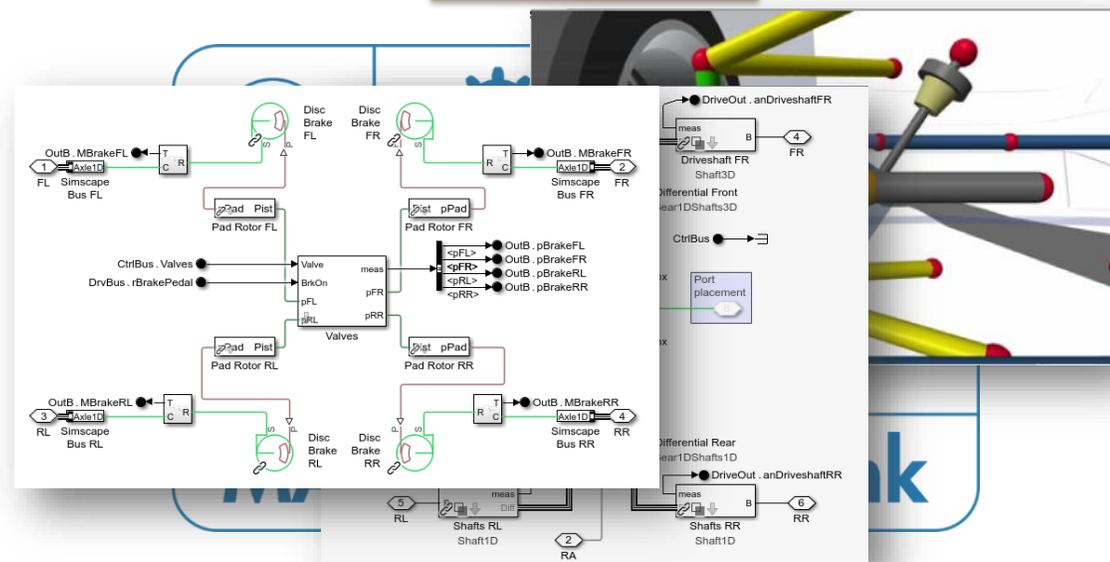
车辆物理对象模型

- 搭建车辆模型的挑战
 - 如何快速构建整车模型
 - 如何方便地自定义不同详细度的部件模型

Automotive Blocksets



Simscape™

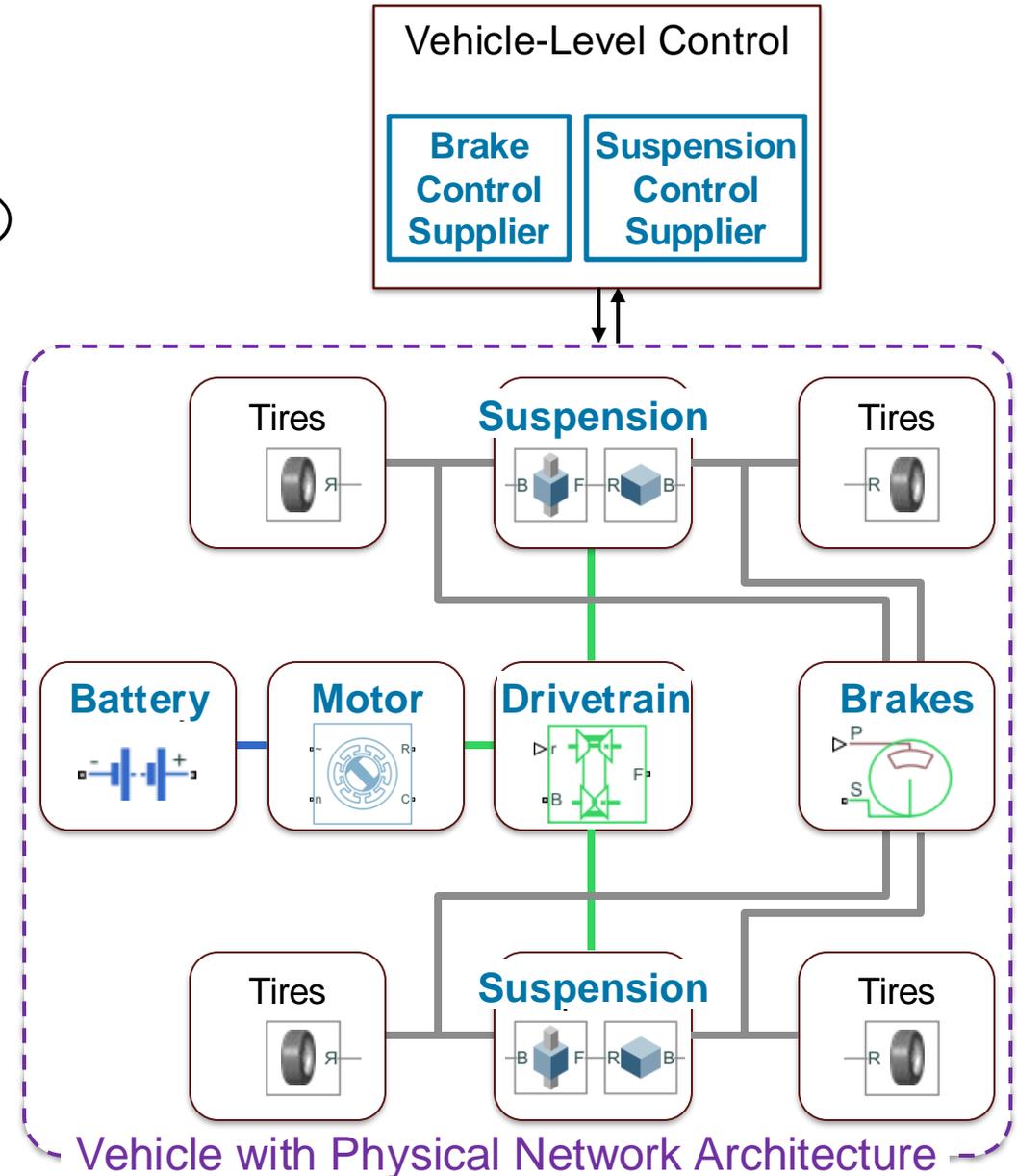
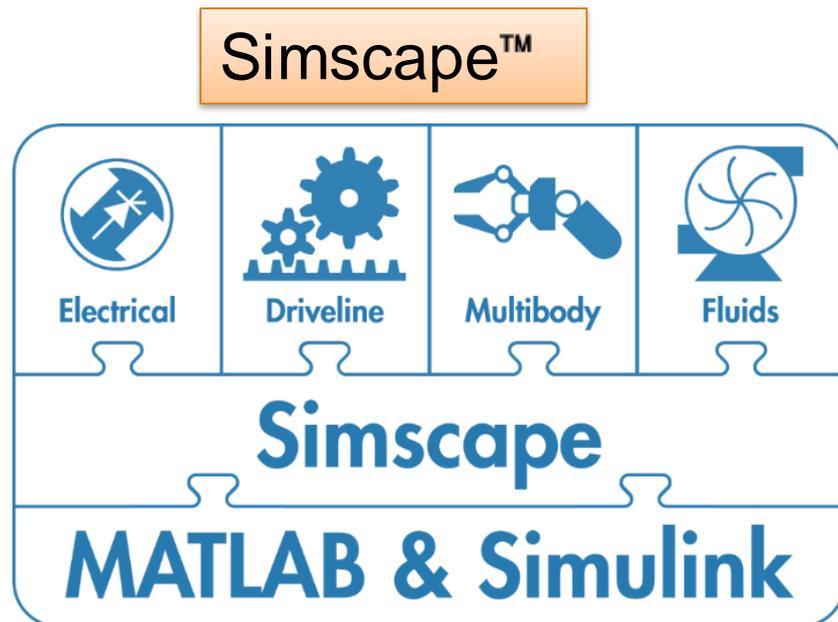


快速构建整车系统模型

自由地构建车辆部件级物理模型

Simscape 物理建模平台

- Simscape专注于基于物理系统建模，
 - 支持跨物理域建模（机械、电气、液压、热力学等）
 - 支持高度自定义
 - 制动系统、悬架系统、电池、传动系统、热管理，电机等
 - 提供各种精度模型。

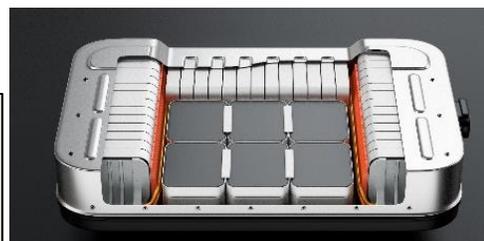
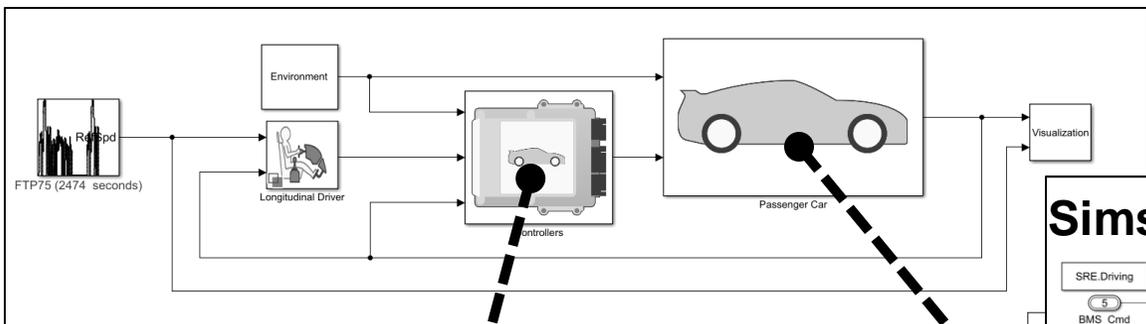


例：Powertrain Blockset：电动汽车的参考应用

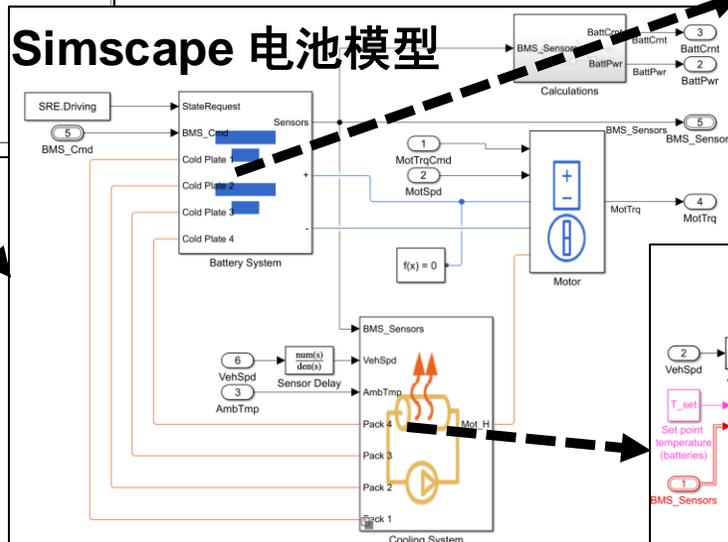
集成电池管理系统（BMS）和电池模型

- 在Powertrain Blockset提供的示例模型：电动车模型中，集成了基于Simscape搭建的更详细的电池模型和电池热管理系统
- 可以仿真电池管理系统的效果并改进电池冷却。

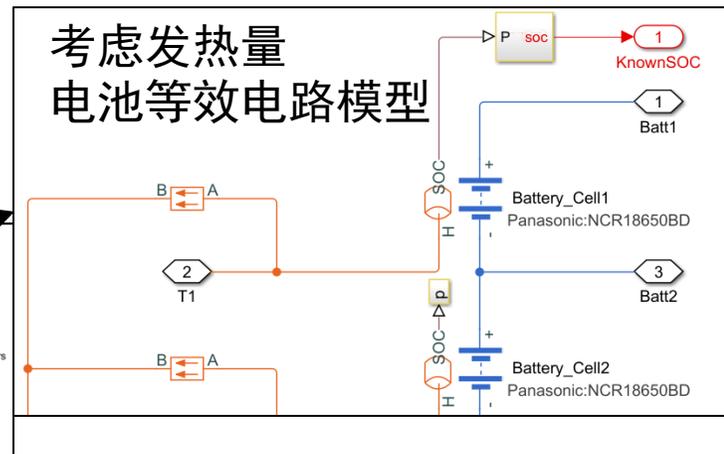
电池电动汽车参考应用模型



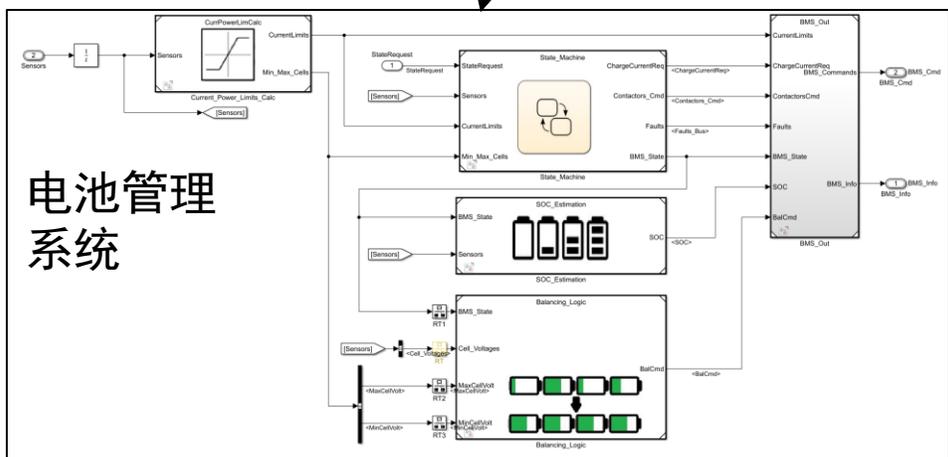
Simscape 电池模型



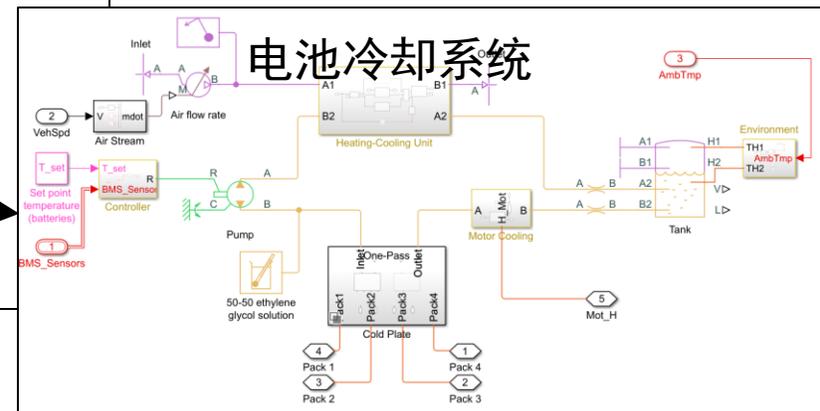
考虑发热量 电池等效电路模型



电池管理系统



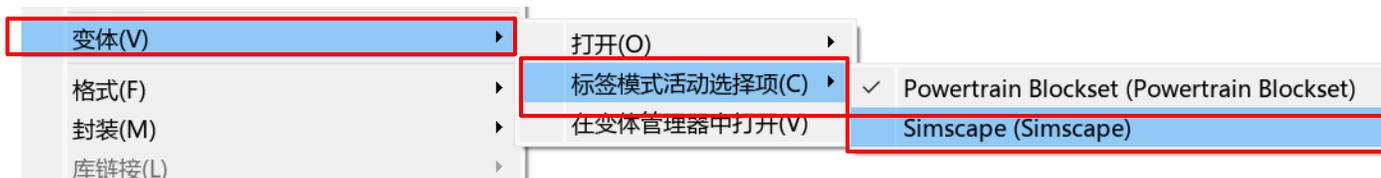
电池冷却系统



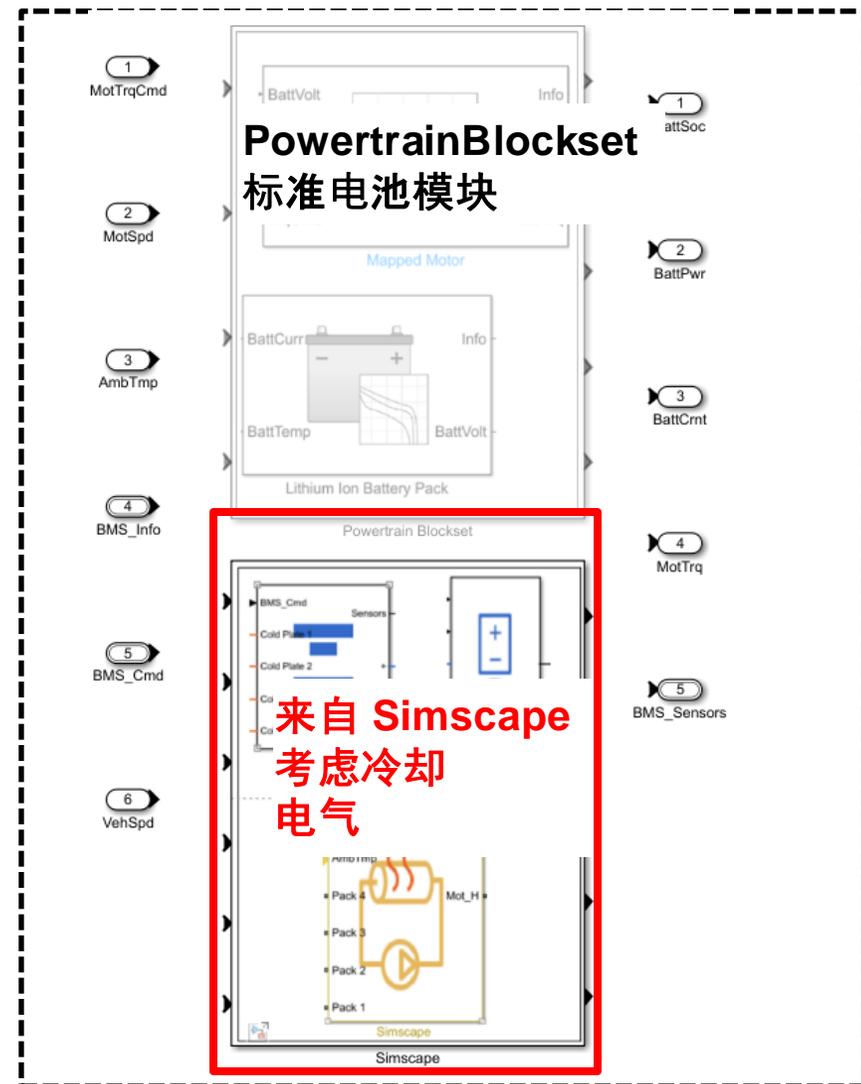
例：Powertrain Blockset电动汽车的参考应用 使用可变子系统（Variant Subsystem）

详细的热管理系统和电气系统

- Simscape 与Simulink 无缝集成。
- 使用“可变子系统（Variant Subsystem）”的功能，将更复杂的Simscape 电气系统模型集成在整车模型中
- 可变子系统（Variant Subsystem）
 - 一个可变子系统重可以有多个子系统
 - 在仿真过程中切换和运行子系统模型
 - 活动变体的切换可以从程序中实现，而无需更改模型



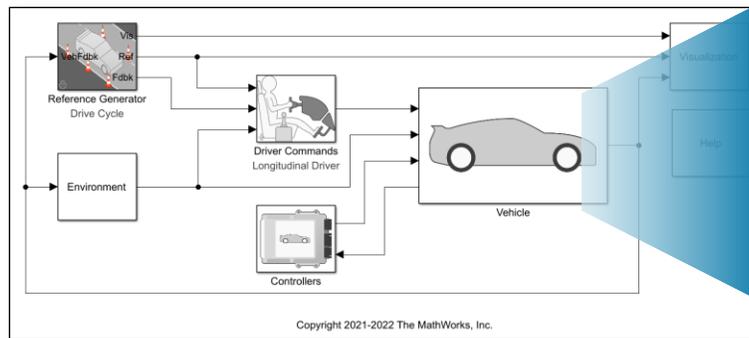
Battery EV参考应用程序模型 电气系统的可变子系统模型



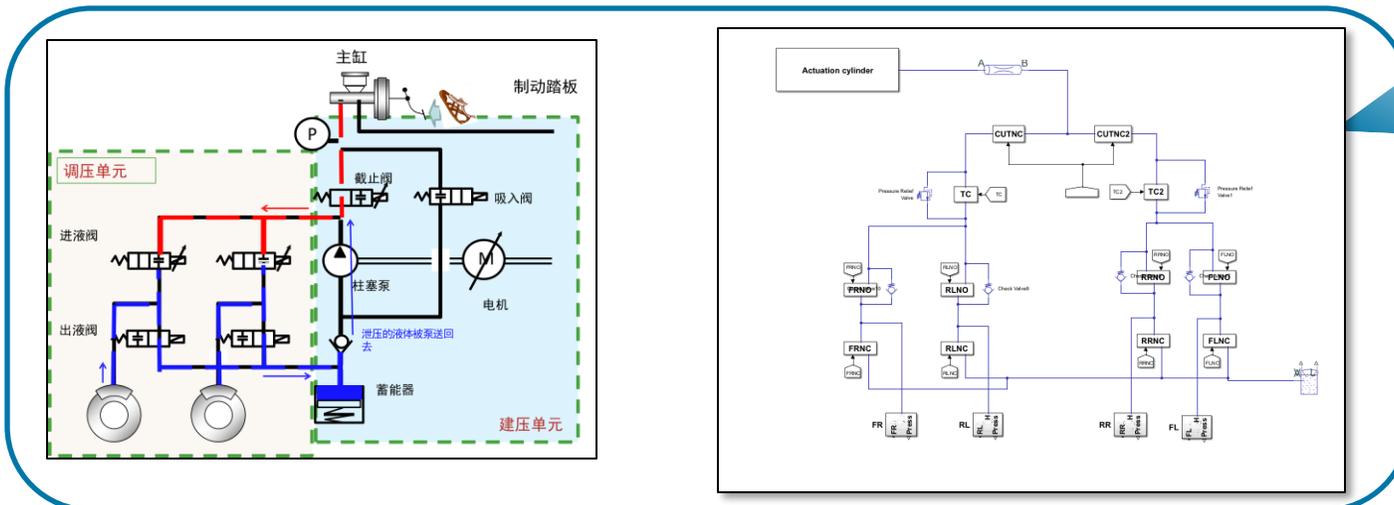
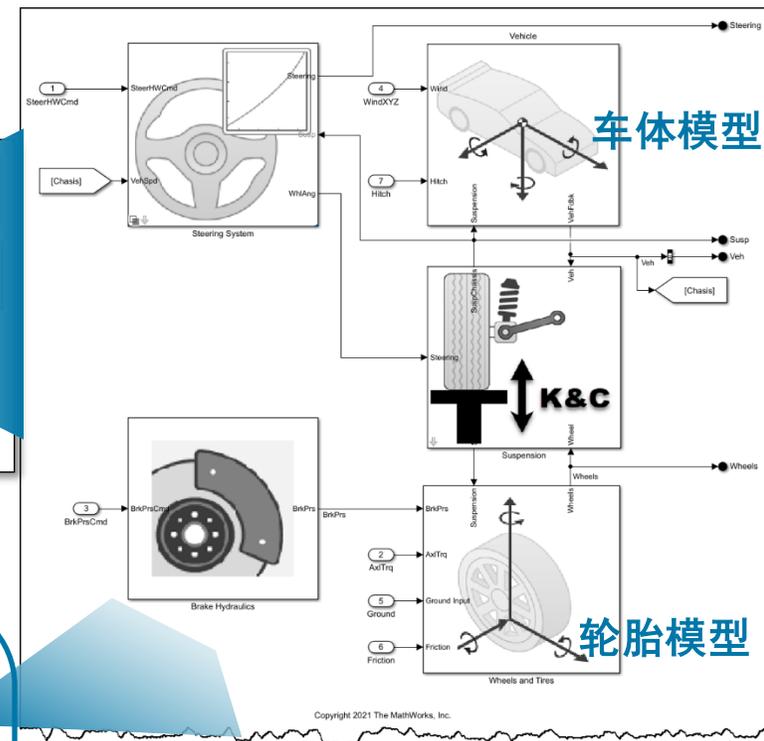
例：搭建液压制动系统EHB模型集成到整车

xEV的集成线控制动的物理模型

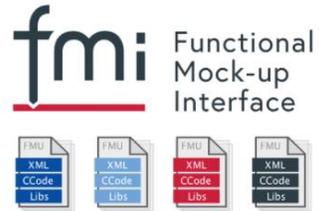
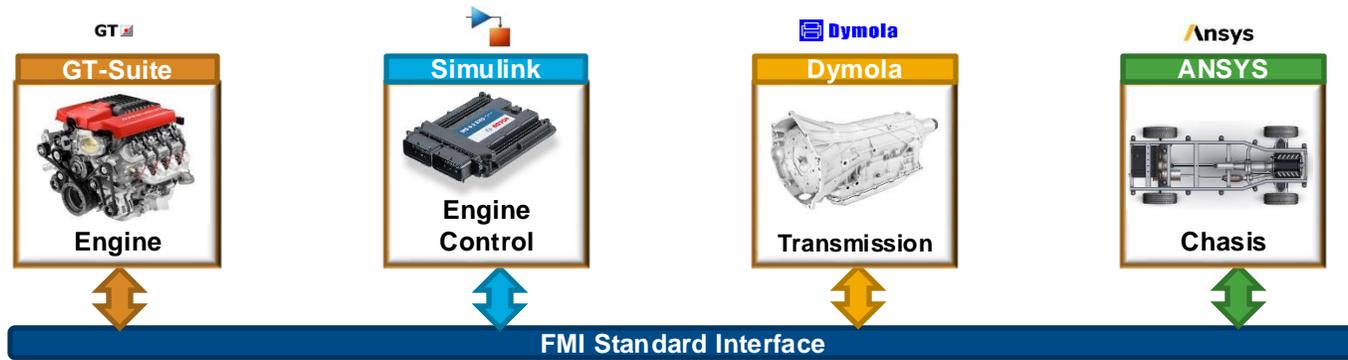
- 基于Virtual Vehicle Composer搭建整车模型。
- 基于Simscape搭建液压制动系统
- 集成细节制动系统到整车模型中。



转向系统



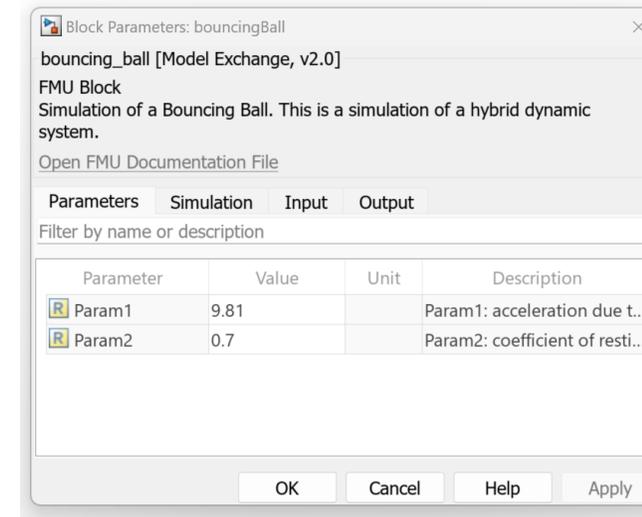
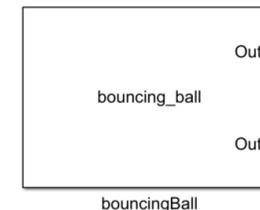
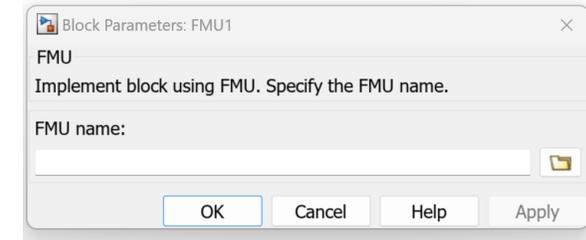
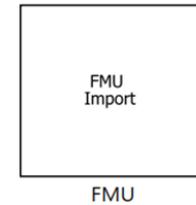
通过FMU集成外部软件



FMI 是一个免费标准，它定义了一个容器和一个接口来交换动态模型。

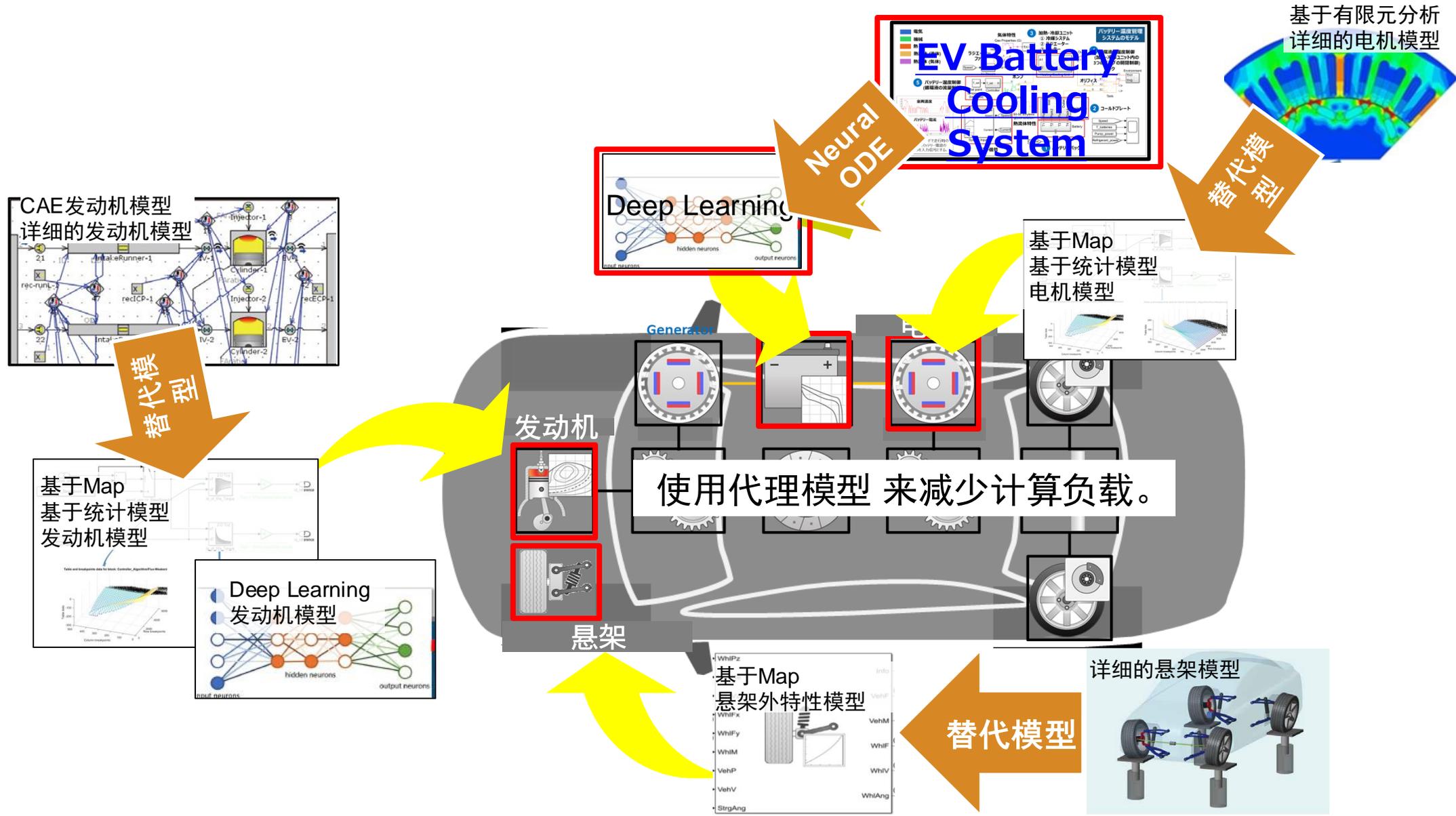
<http://fmi-standard.org/tools/>

- Simulink Extras/FMU 导入/FMU
- 支持 FMI v1.0、v2.0 和 v3.0（自 2023B 起）
- 自动选择 FMU 模式
- 提供对话框树视图，便于参数管理



[Import FMUs](#)

模型降阶 (Reduced Order Modeling)



详细物理模型的代理模型

模型降阶 (Reduced Order Modeling)

为详细的物理模型创建以下代理模型（替代模型），以减少仿真负载

模型类型

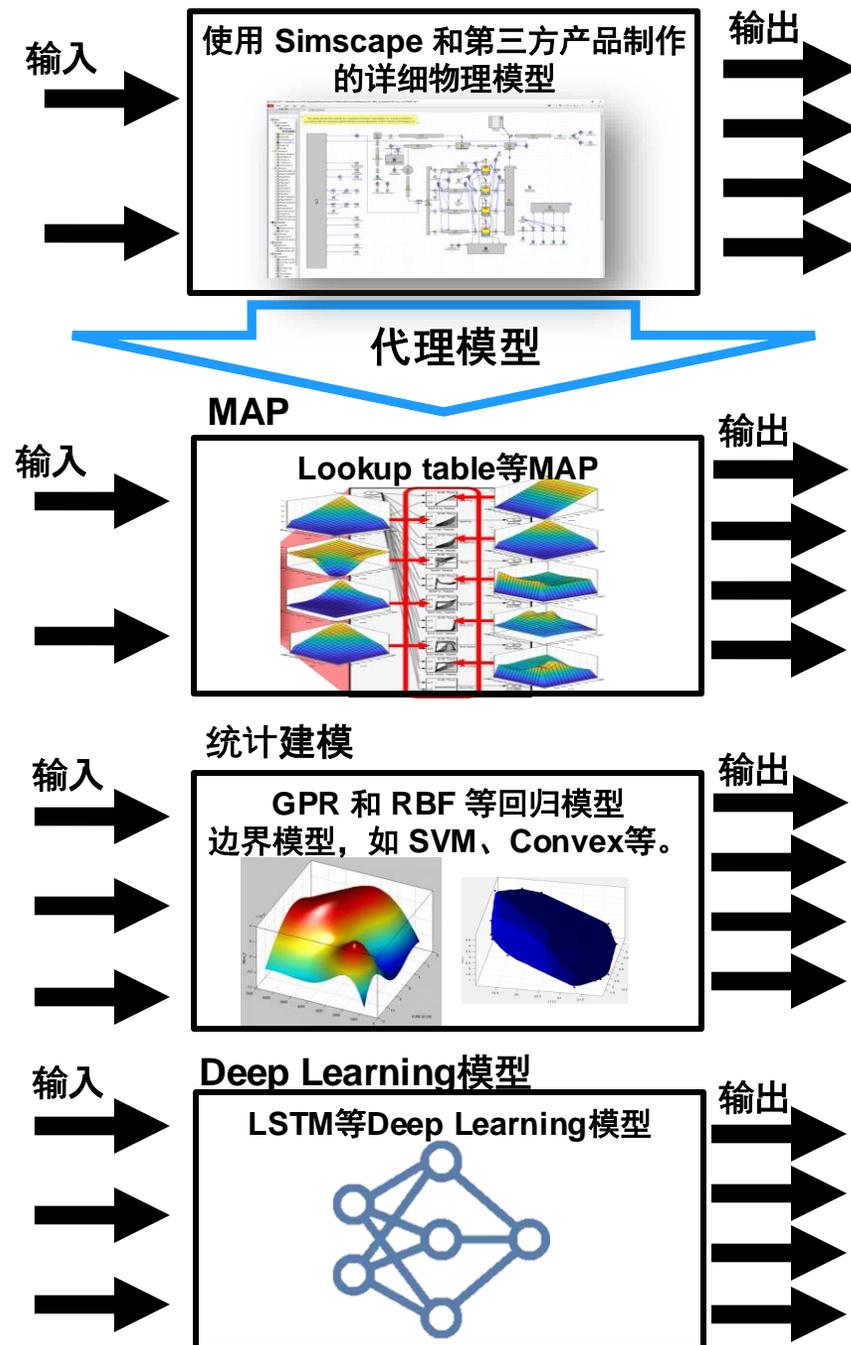
⇒ 稳态模型

- 基于MAP 模型
- 基于统计的机器学习模型

⇒ 瞬态模型

- Deep Learning模型(例:LSTM 模型)
- 使用系统识别进行建模
(System Identification Toolbox)

在 Simulink 上对现有的详细模型执行虚拟测试，以获取上述模型所需的数据。

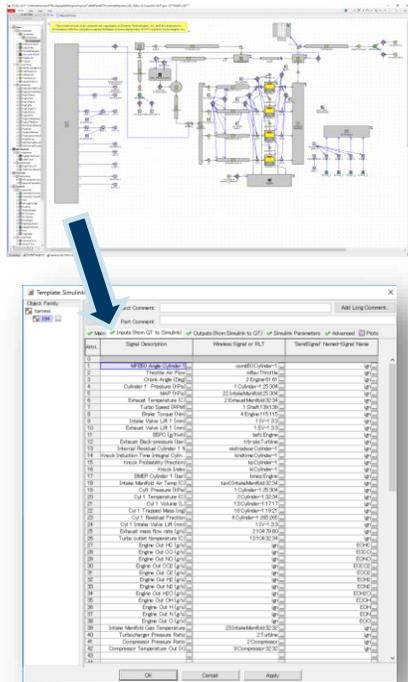


模型降阶 (Reduced Order Modeling)

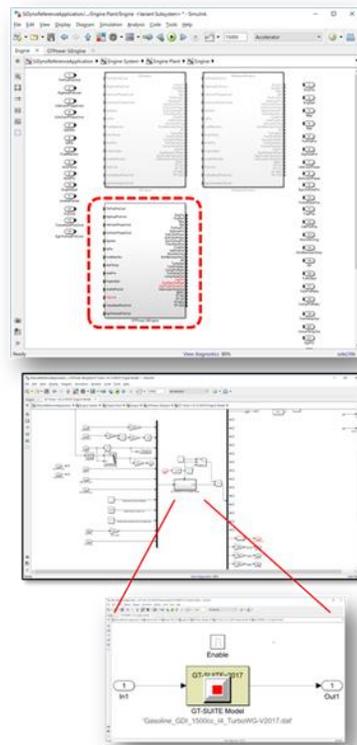
机器学习建模：发动机模型降阶的机器学习模型生成 workflow



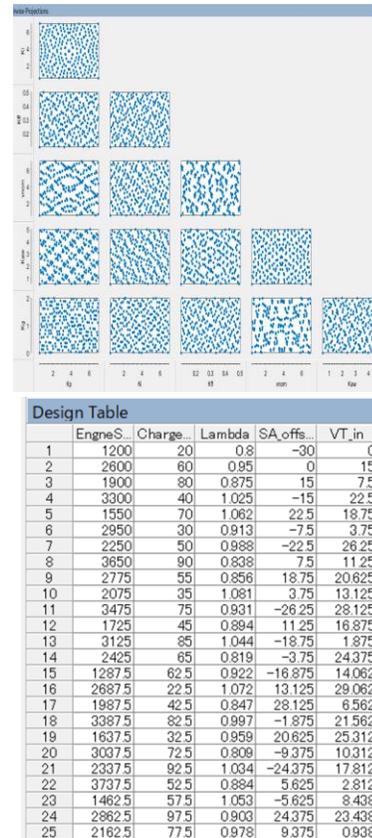
GT-Power 模型



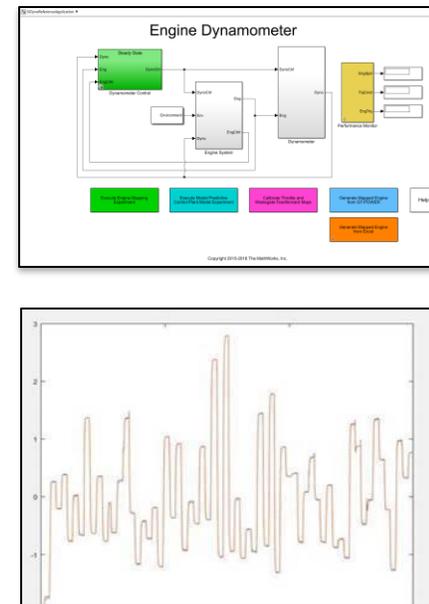
Simulink 模型



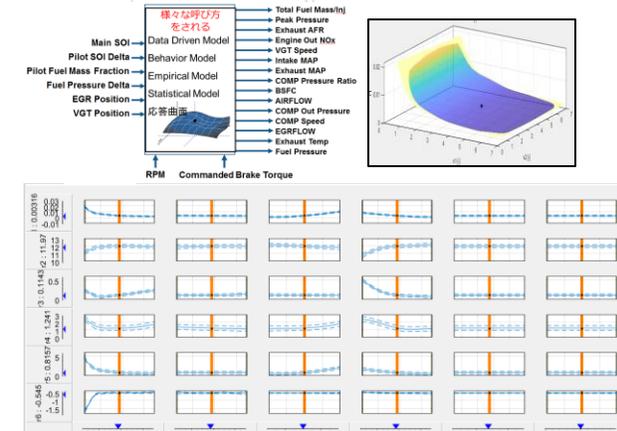
DoE(实验设计)



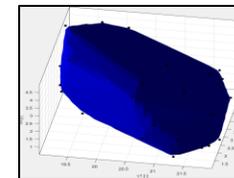
虚拟试验台架



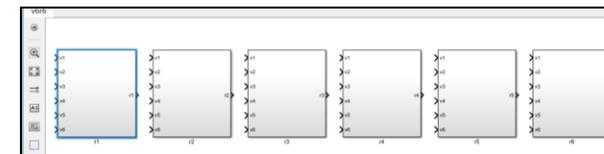
机器学习回归模型创建



由凸包 (Convex hull) 定义的边界模型



将回归模型和边界模型导出到 Simulink 仿真模型



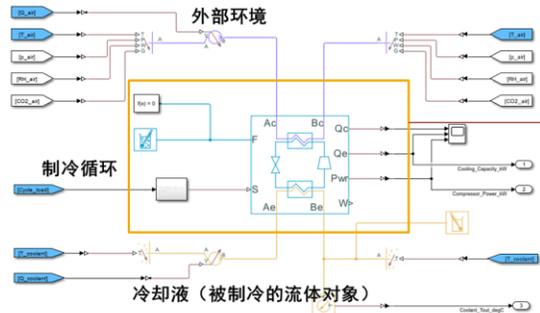
模型降阶 (Reduced Order Modeling)

深度学习建模：制冷循环系统模型的模型降阶

①通过虚拟仿真测试获取训练数据

②创建并训练深度学习的Neural ODE 模型

③降阶模型仿真结果与原物理模型仿真对比

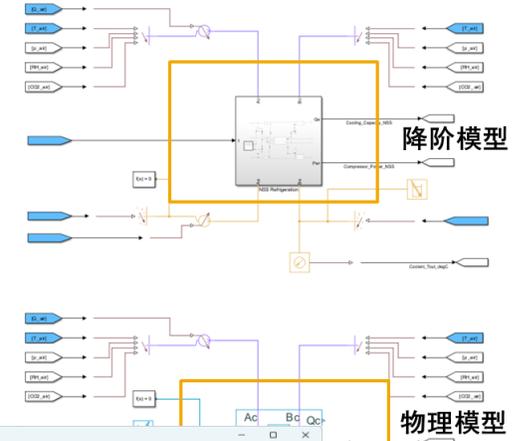
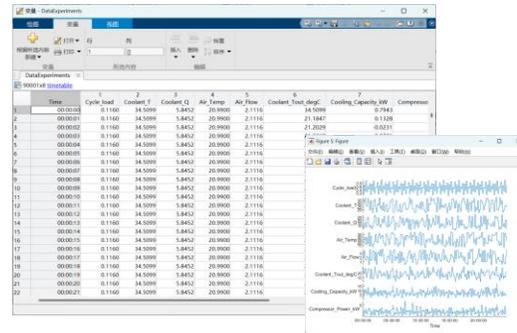
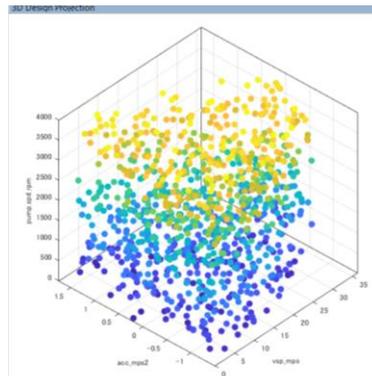


创建基于深度学习的非线性状态空间模型

$$\begin{cases} \dot{x} = f(x, u) \\ y = g(x, u) \end{cases}$$

State Network (f)

Output Network (g)



* Neural ODE 模型集成到Simulink模型中并与物理模型仿真结果进行对比验证

例如，使用实验设计 (DoE) 设置与输入变量相关的参数



组件建模技巧

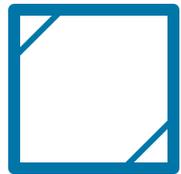
组件建模方法

要点



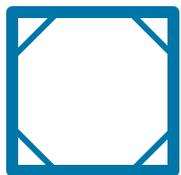
Library

- 存储在单独的文件中
- 添加到 Simulink 库浏览器
- 用于小型、稳定和高度可重复使用的组件



Subsystem Reference

- 存储在单独的文件中
- 动态接口
- 支持 Simscape 物理模型接口
- 非常适合存储部件设计模型



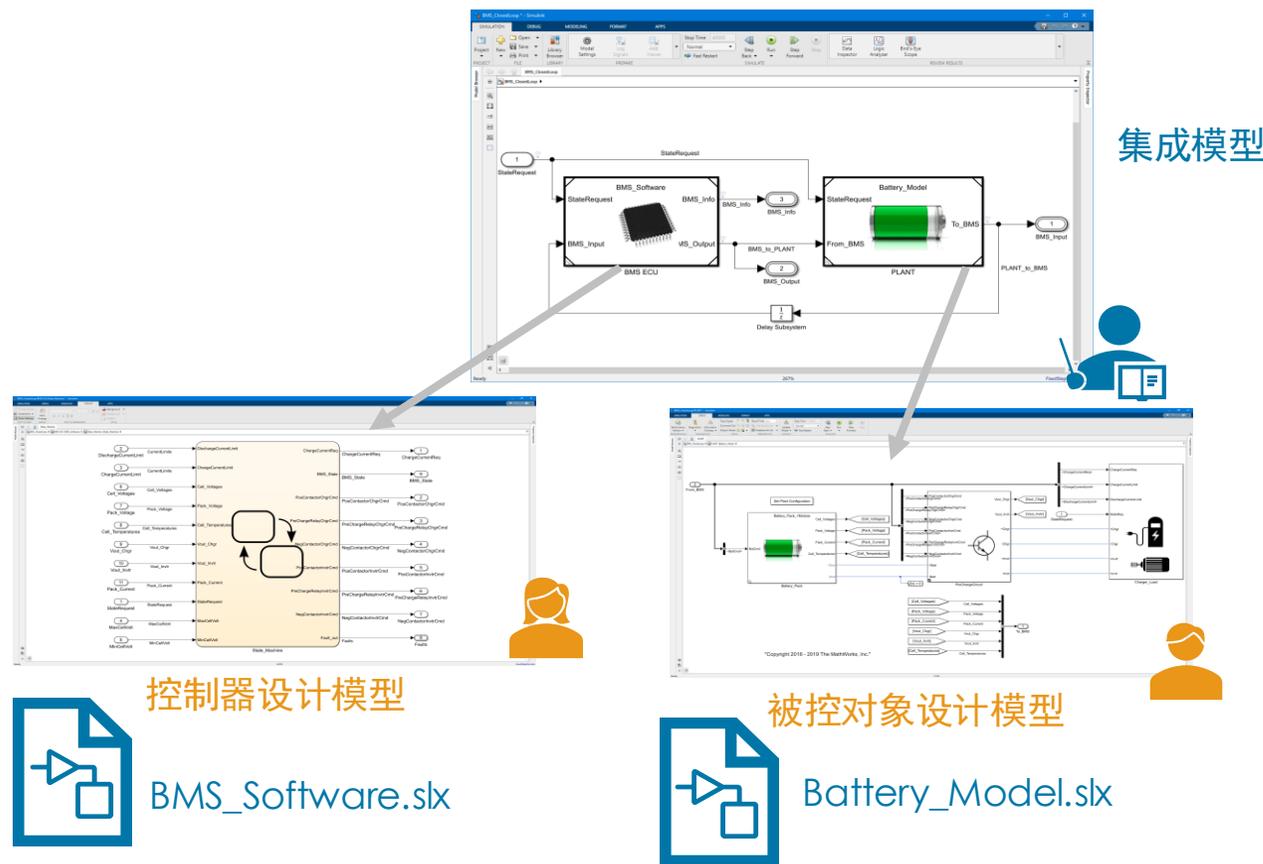
Model Reference

- 存储在独立文件中
- 可以独立运行
- 支持单独开发
- 需要定义接口

组件建模技巧

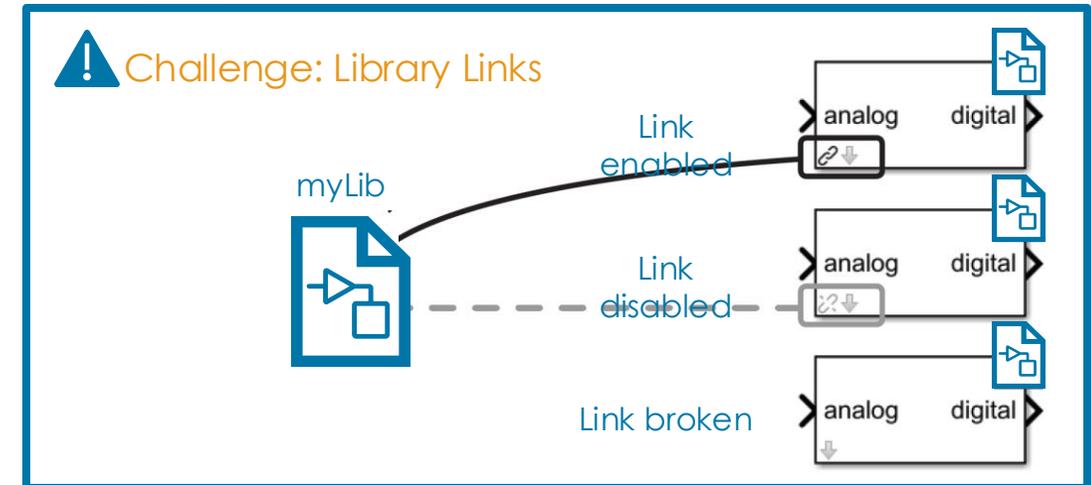
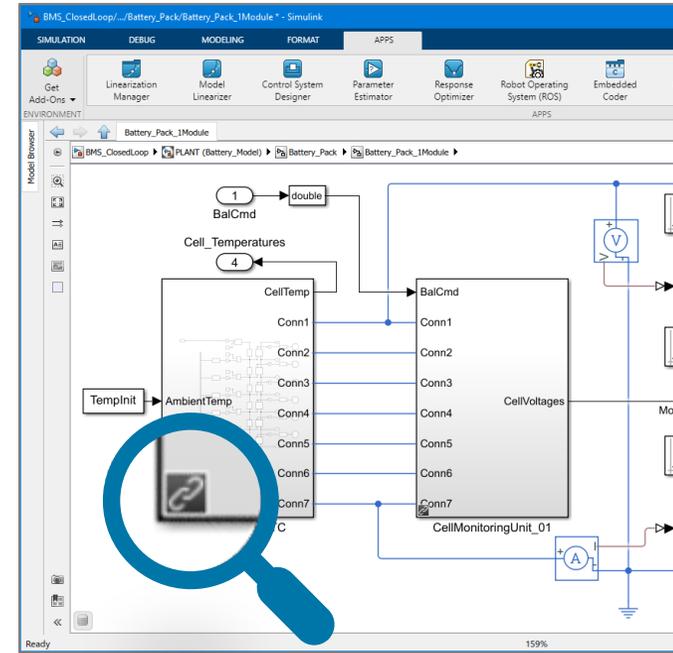
模型引用 (Model Reference)

- 存储在独立文件中
- 可以独立运行
- 支持单独开发
 - 可以直接被任何模型重用
 - 模型可以被在其他模型引用进而集成。
 - 重用时减少构建时间。
 - 引用的模型现在可以使用自己的定步长求解器进行仿真
 - 支持加速器和快速加速器模式。



组件化技术 库 (Library)

- 存储在单独的文件中
- 添加到 Simulink 库浏览器
- 用于小型、稳定和高度可重复使用的组件

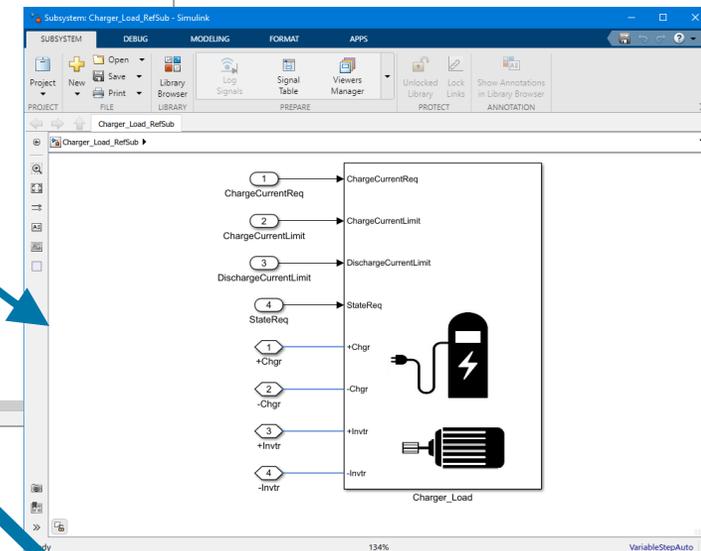
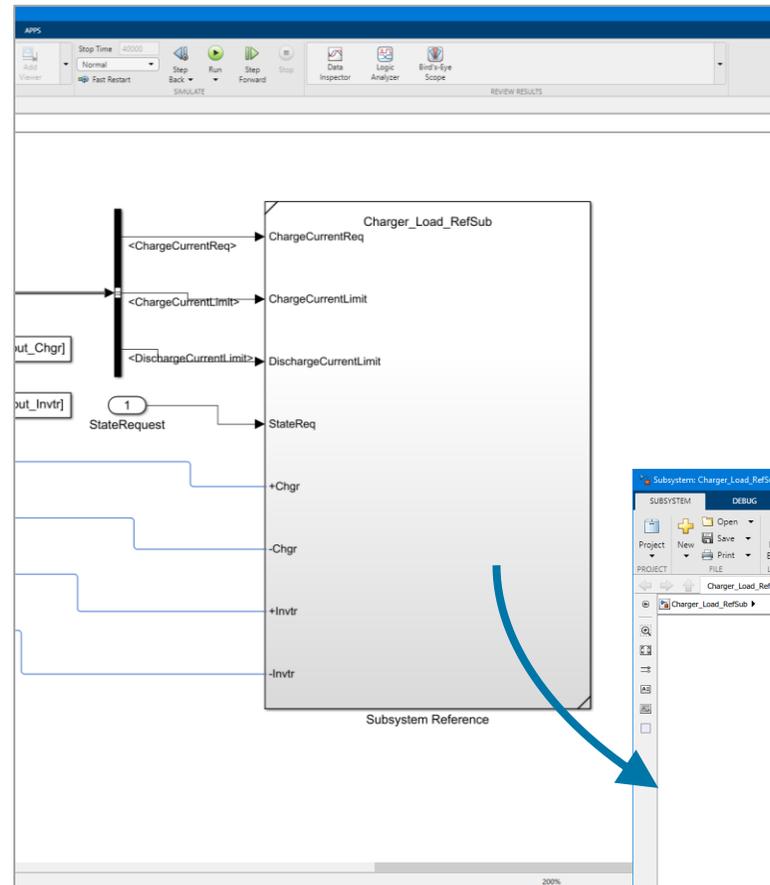


组件化技术

子系统引用 (Subsystem Reference)



- 存储在单独的文件中
- 动态接口
- 支持Simscape 物理模型接口
- 非常适合存储部件设计模型

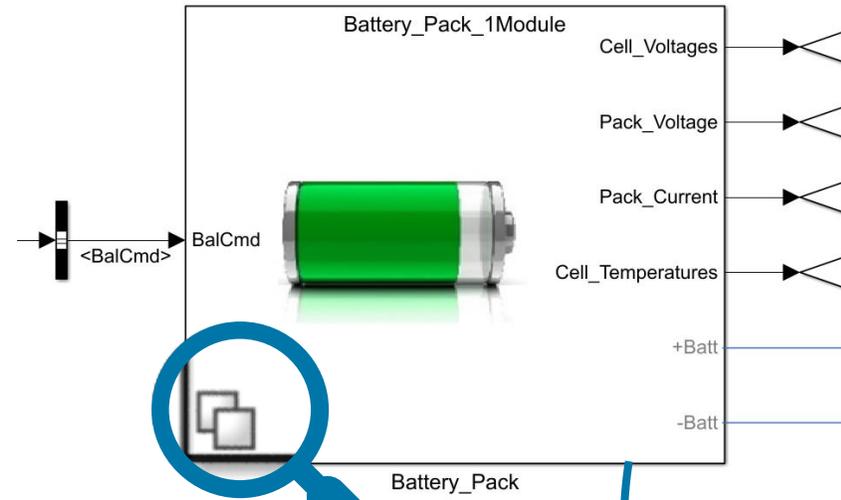
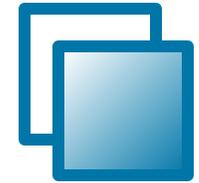


Charger_Load_RefSub.slx

组件化技术

可变子系统 (Variant Subsystem)

使用变体在组件的不同变体 (模型参考或子系统) 之间自动切换



Block Parameters: Battery_Pack

Variant Subsystem

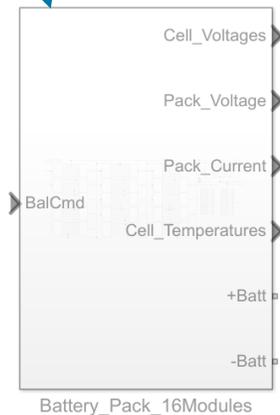
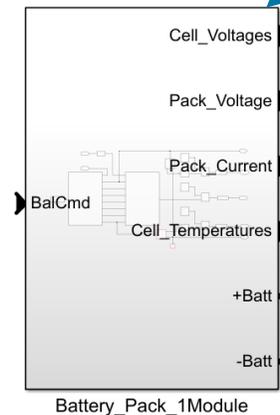
The Variant Subsystem contains one or more choices where each choice is a Subsystem or Model block. At most one choice is active in simulation. The active choice is determined by the variant control column of the variant choices table. Variant Subsystem can be in expression mode or label mode. In expression mode, the variant control can be a boolean expression, a Simulink.VariantControl object, a boolean expression, or '(default)'. In label mode, the variant control is a string that is not evaluated and the choice used is determined by the label mode active choice parameter.

Variant control mode: Expression

Name (read-only)	Variant control expression
Battery_Pack_16Modules	NumModules == 16
Battery_Pack_1Module	NumModules == 1

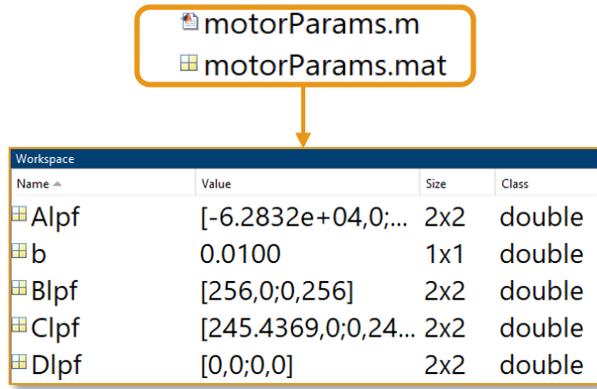
Variant Control

1
BalCmd



- 1 Cell_Voltages
- 2 Pack_Voltage
- 3 Pack_Current
- 4 Cell_Temperatures
- 1 +Batt
- 2 -Batt

仿真数据管理存储在哪里？



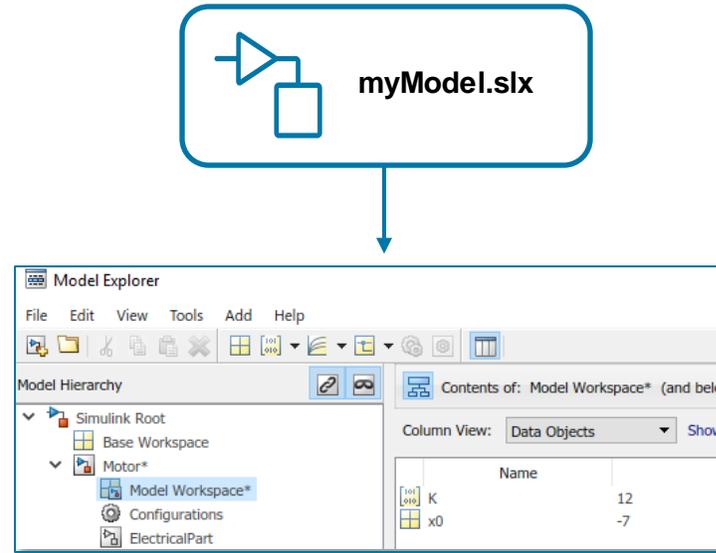
 **Base workspace**

 **MATLAB Script**
.m

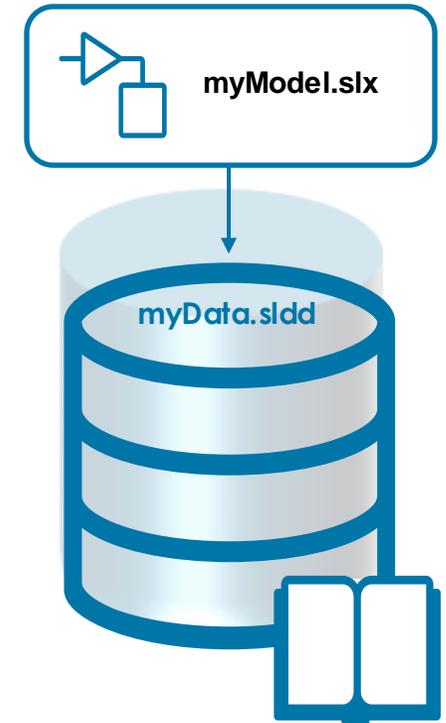
 **MAT file**
.mat

Pros:
易于阅读和修改
Cons:
运算负荷加重
加载速度较慢

Pros:
轻量
可快速加载
Cons:
难以阅读和修改

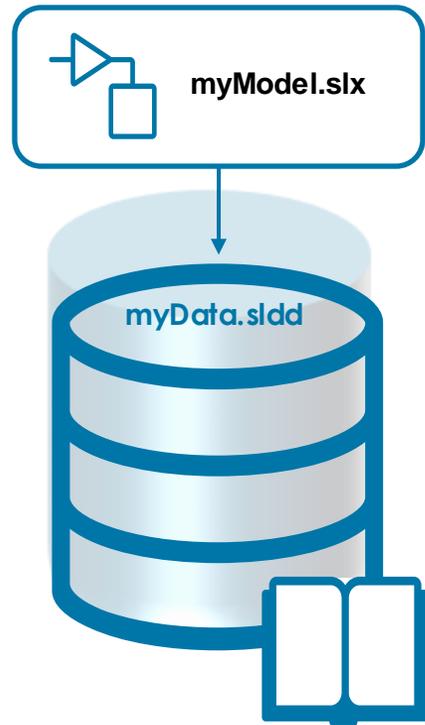


 **Model workspace**



Data Dictionary

数据字典



Data dictionary

Column View: Dictionary Objects

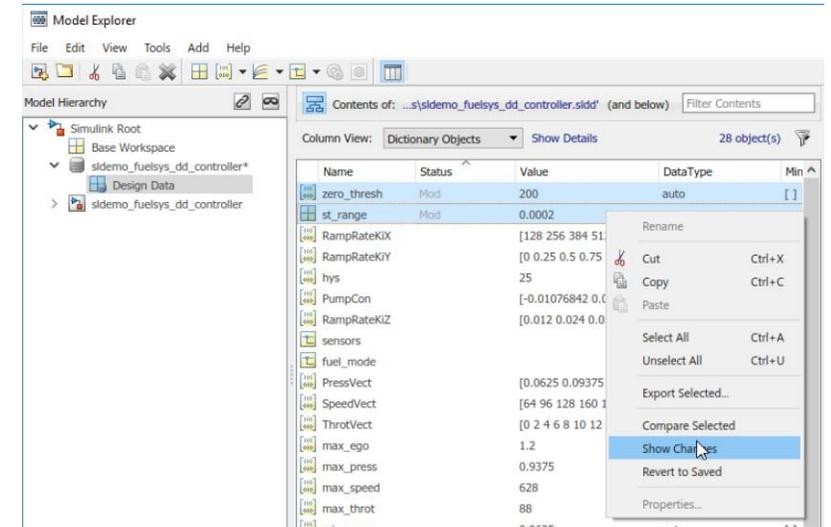
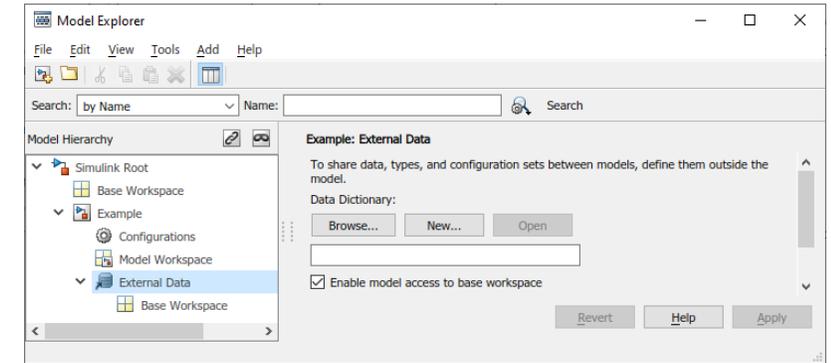
Name	DataType
CtrlBus	
SensorBus	
rho	auto
FloatType	
coeff	FloatType
diff	FloatType
init	FloatType
SimConfigSet	
CodeGenConfig...	

组件接口(总线)

自定义数据类型

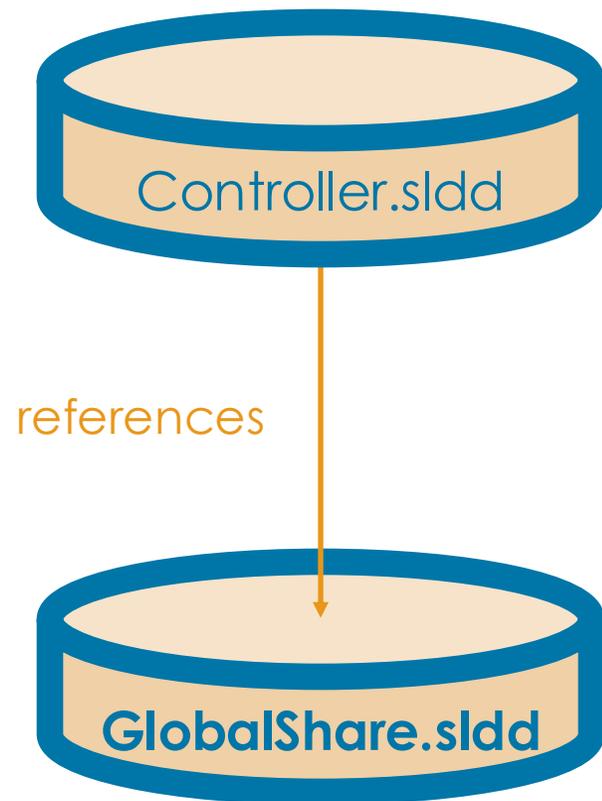
参数

配置参数



数据字典 – 数据分区

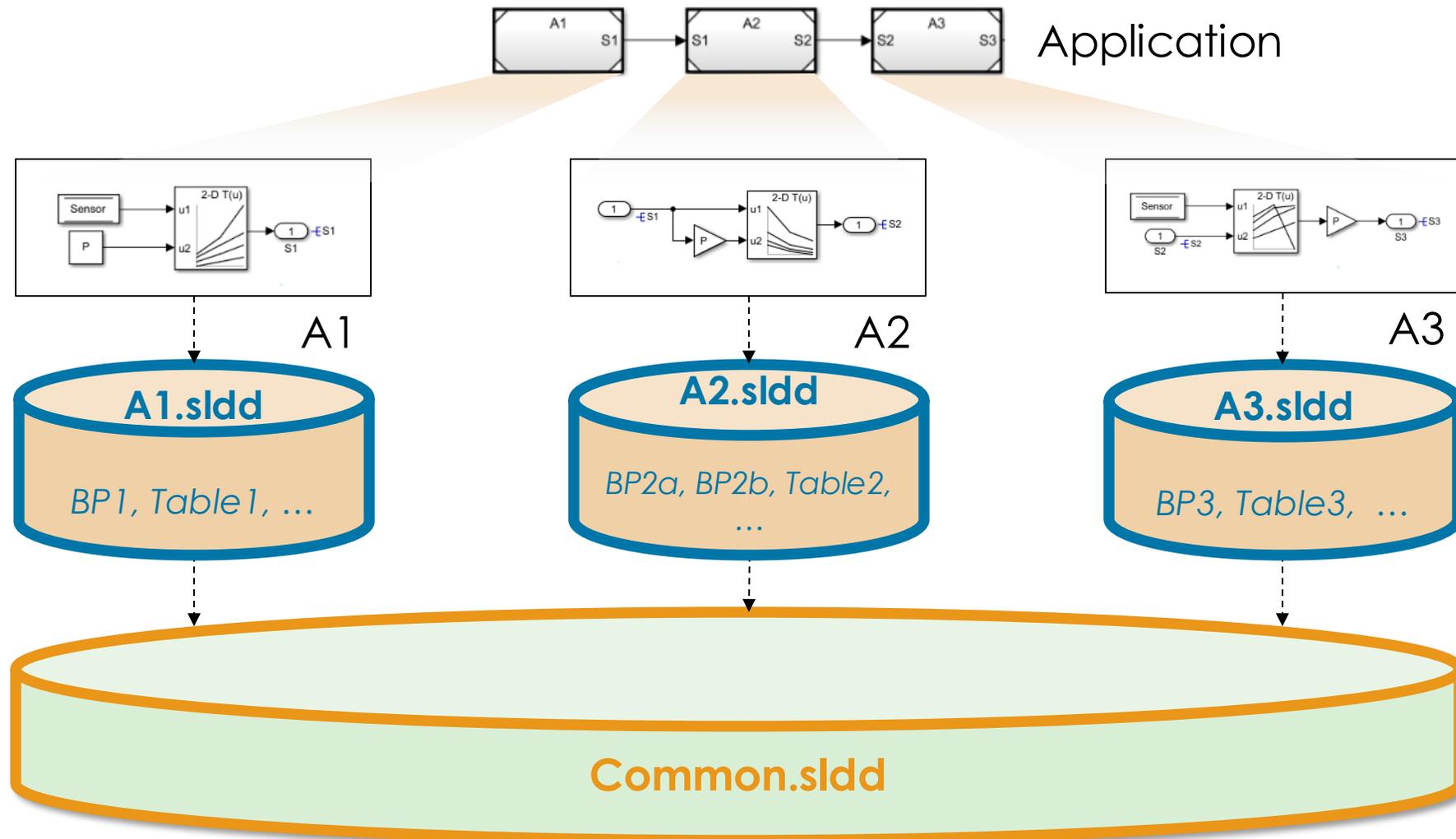
链式数据字典 - 将组件数据与常见数据区分开来



Column View: Dictionary Objects Show Details 9 object(s)

Name	DataType	Argument	Status	Description	DataSource
CtrlBus					GlobalShare.sidd
SensorBus					GlobalShare.sidd
rho	auto				GlobalShare.sidd
FloatType					Controller.sidd
coeff	FloatType				Controller.sidd
diff	FloatType				Controller.sidd
init	FloatType				Controller.sidd
SimConfigSet					GlobalShare.sidd
CodeGenConfig...					Controller.sidd

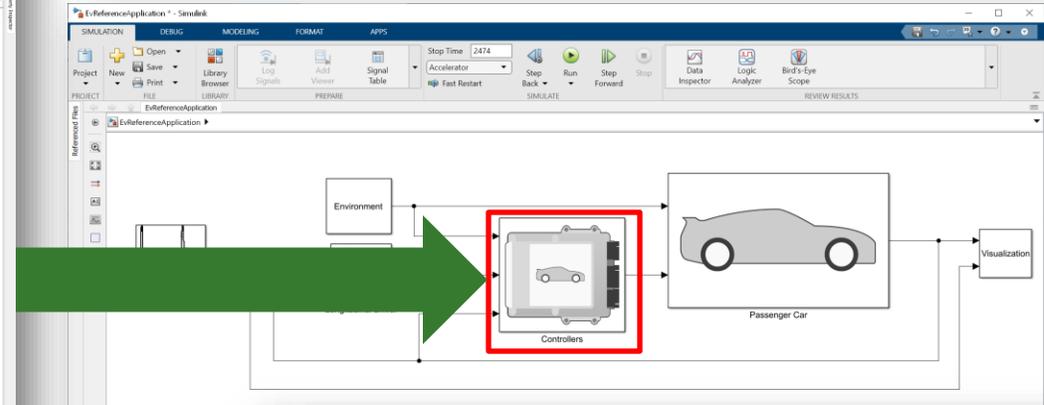
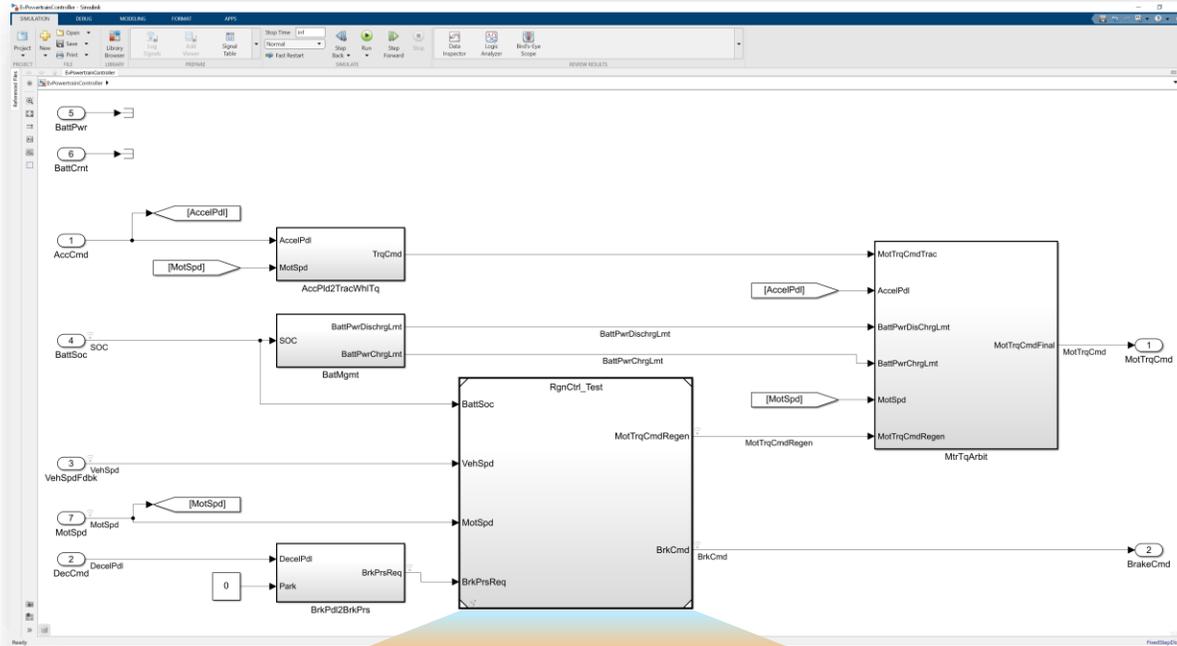
组件化数据管理



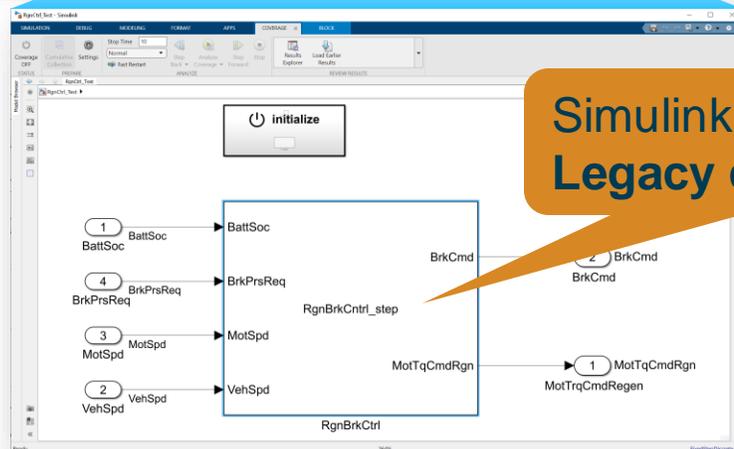


例：从手写代码进行基于模型的设计

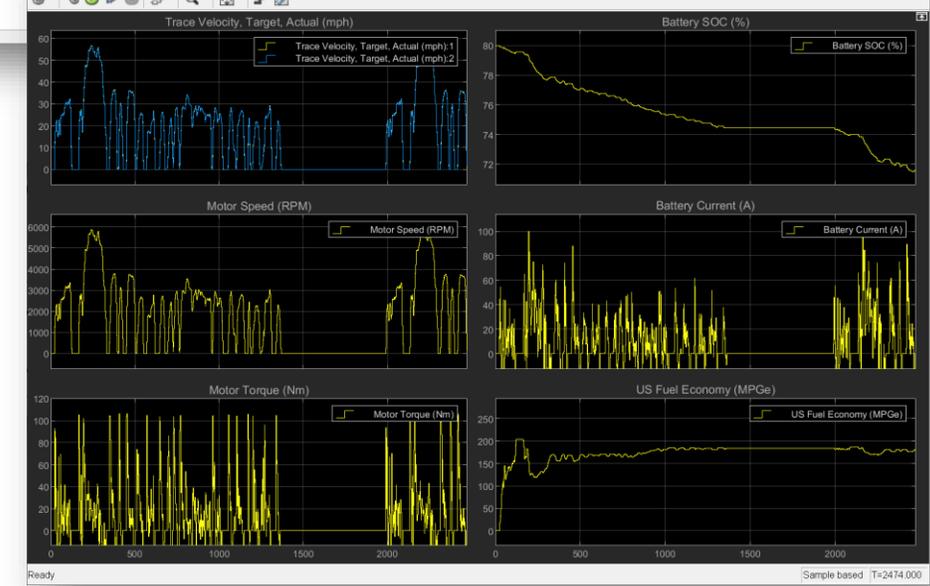
Example: EV Vehicle Simulation for VCU



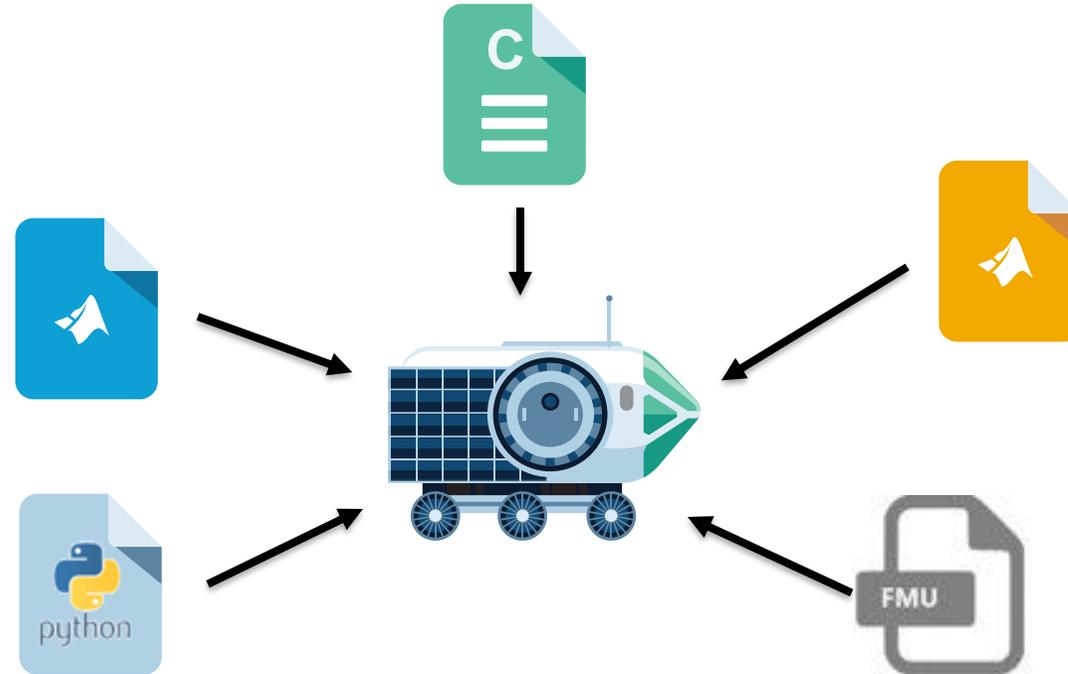
Analyze Power and Performance and FE Scope



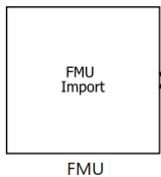
Simulink block calling Legacy codes



集成外部程序

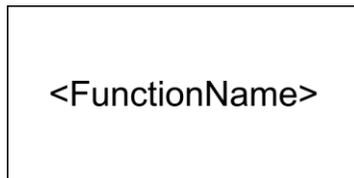


R2017b



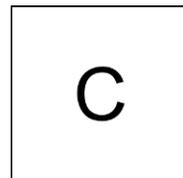
FMU Import

R2018b



C Caller

R2020a



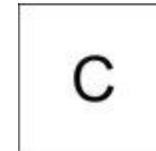
C Function

R2021a



Code Importer

R2022a



C Function Supports C++

R2023a



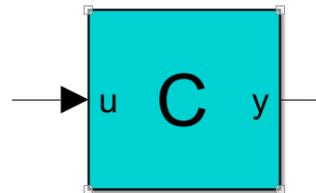
Python Importer

自定义 C\C++ 集成



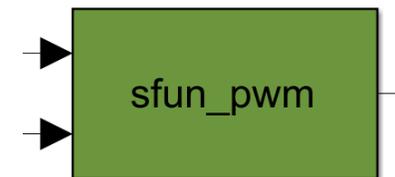
C Caller

- 简单地调用一个函数
- 自动创建 I/O 接口
- 导入共用库和 C 代码进行单元测试的理想选择



C Function

- 调用多个函数
- 支持在模块中实例化 C++ 类对象和调用类方法

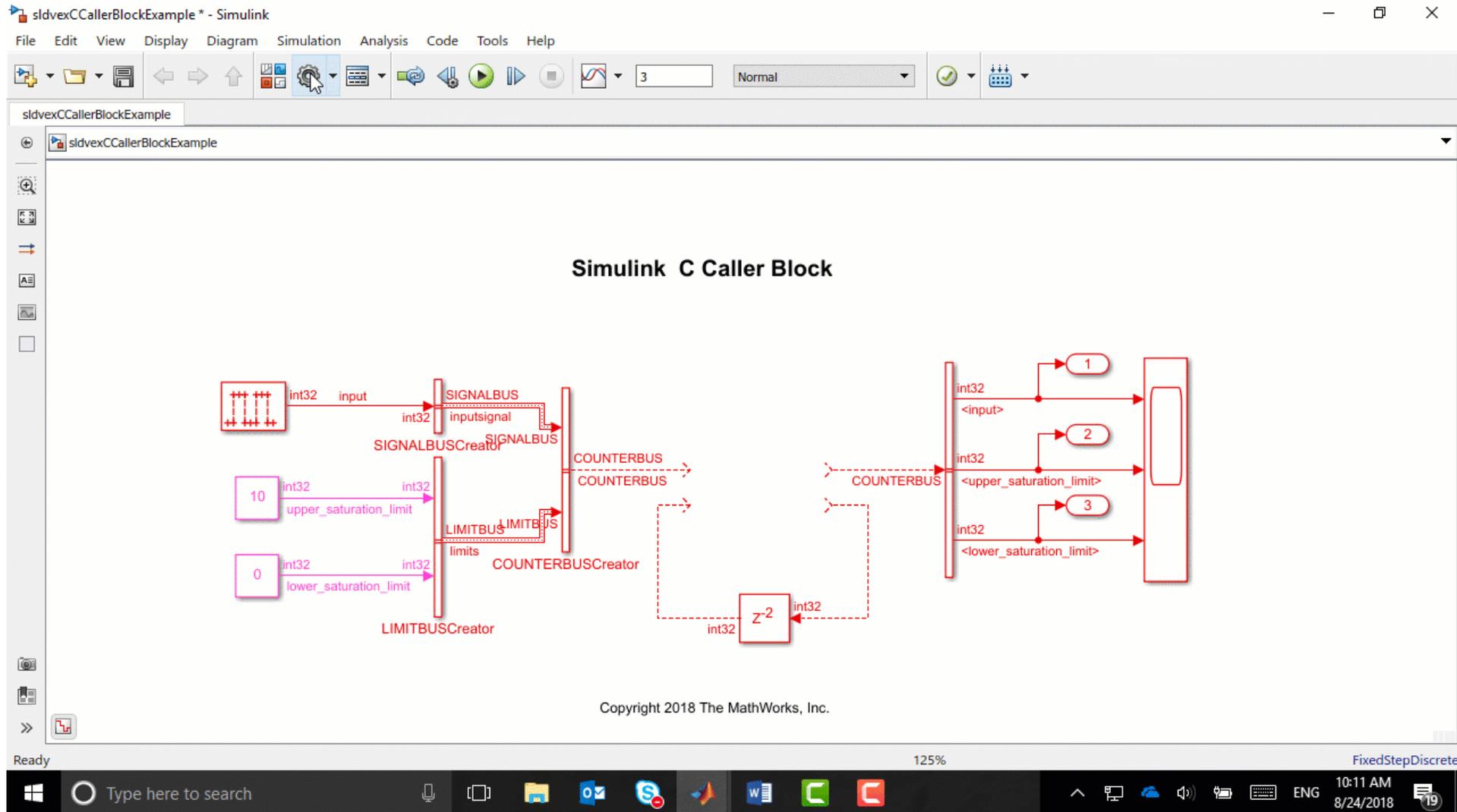


S-Function & S-Function Builder

- 通用性强
- 集成动态系统的理想选择

复杂度

Demo 演示 – C Caller 模块集成单一 C 函数

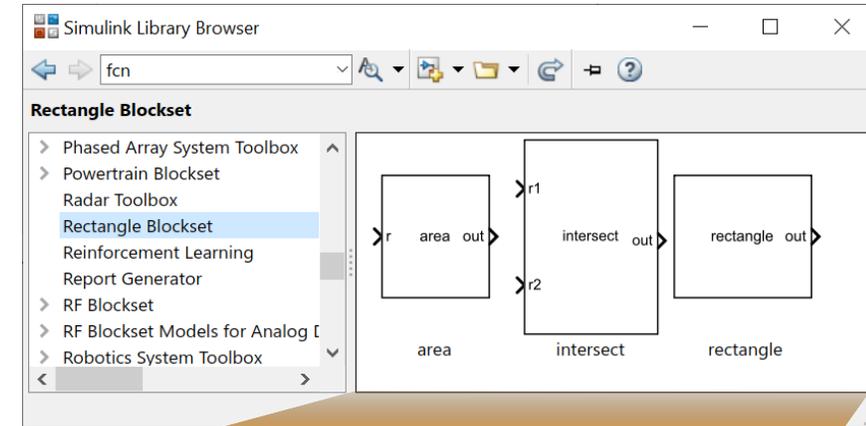
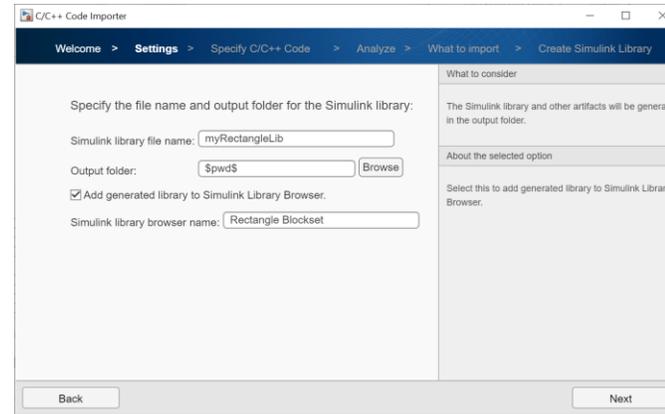
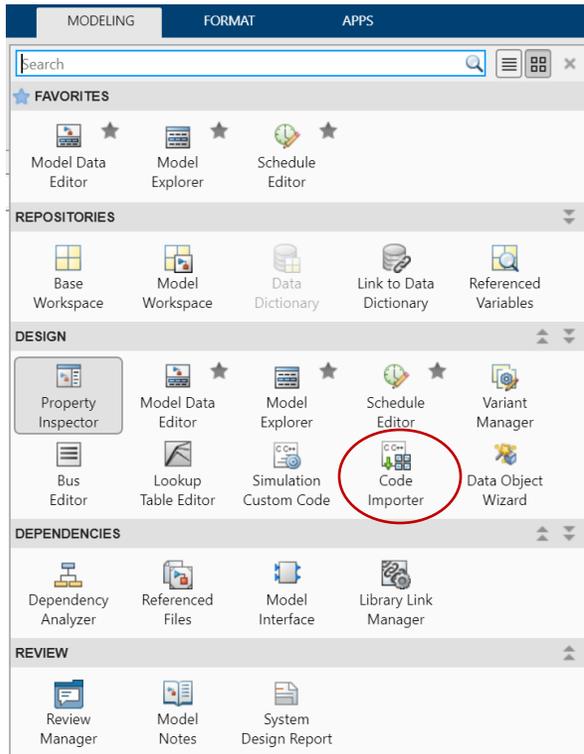


比 S 函数和
Legacy Code
Tool 操作更简单

支持仿真和代码生
成（集成）

Simulink 代码导入器

将自定义 C/C++ 代码转换为可复用的 Simulink 库

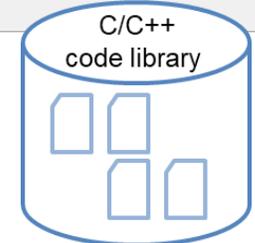


相关性 > 分析 > 匹配 > 导入

交互式向导提供了手把手的逐步指导
也提供 **MATLAB APIs** 的直观设置

<https://www.mathworks.com/help/simulink/code-importer.html>

将 C 代码转换为可复用的 **Simulink 库**
使用 C caller 来代表 C 代码算法的模块

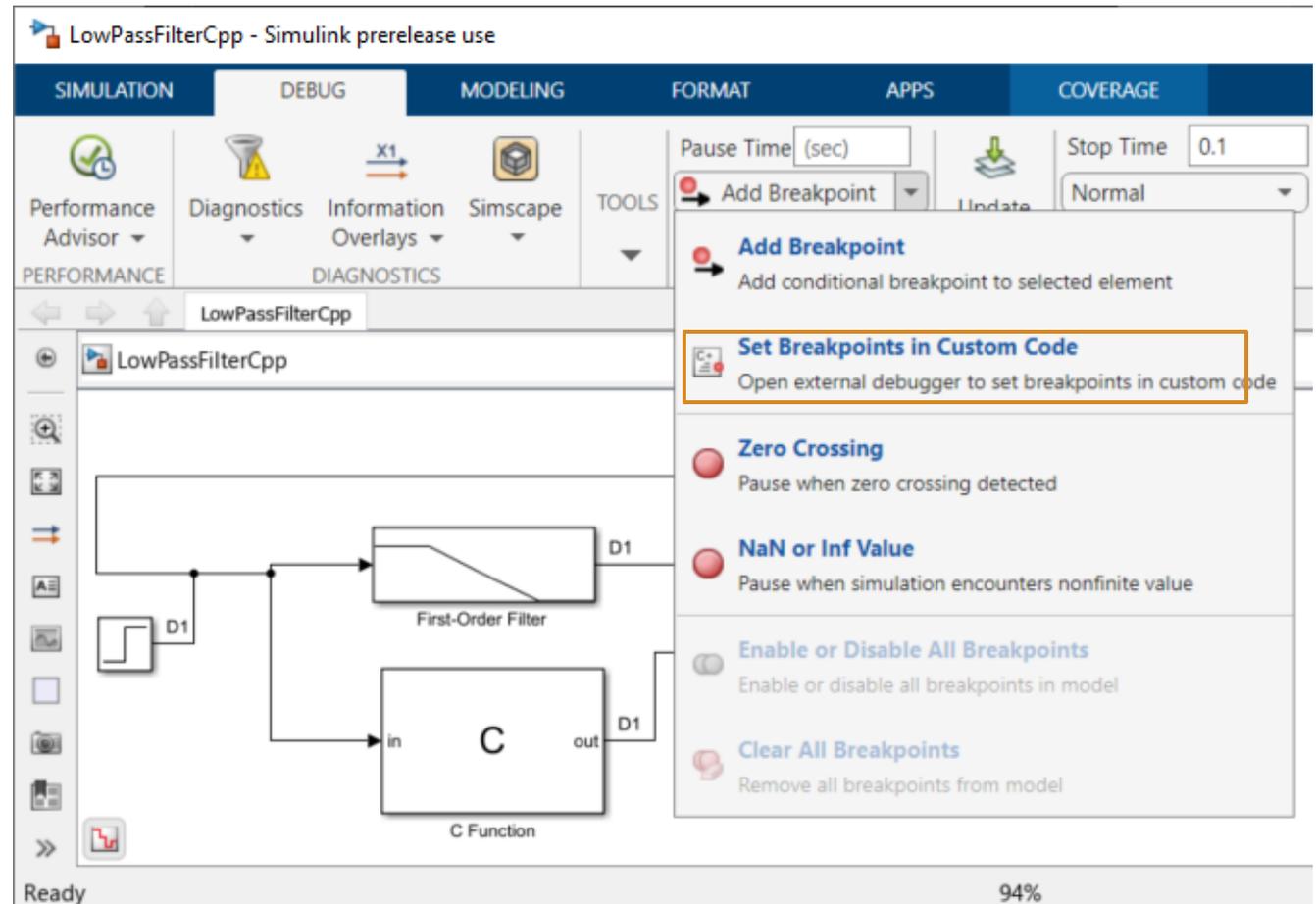


```

rectangle.h
1  typedef struct {
2     double x;
3     double y;
4  } Position;
5
6  typedef struct {
7     double w; // width
8     double h; // height
9     Position lower_left; // lower left coordinates
10 } Rectangle;
11
12 // Create a rectangle
13 extern Rectangle rectangle(double width, double height,
14                            double x, double y);
15
16 // Compute intersection of two rectangles
17 extern Rectangle intersect(Rectangle r1, Rectangle r2);
18
19 // Compute area of a rectangle
20 extern double area(Rectangle r);
  
```

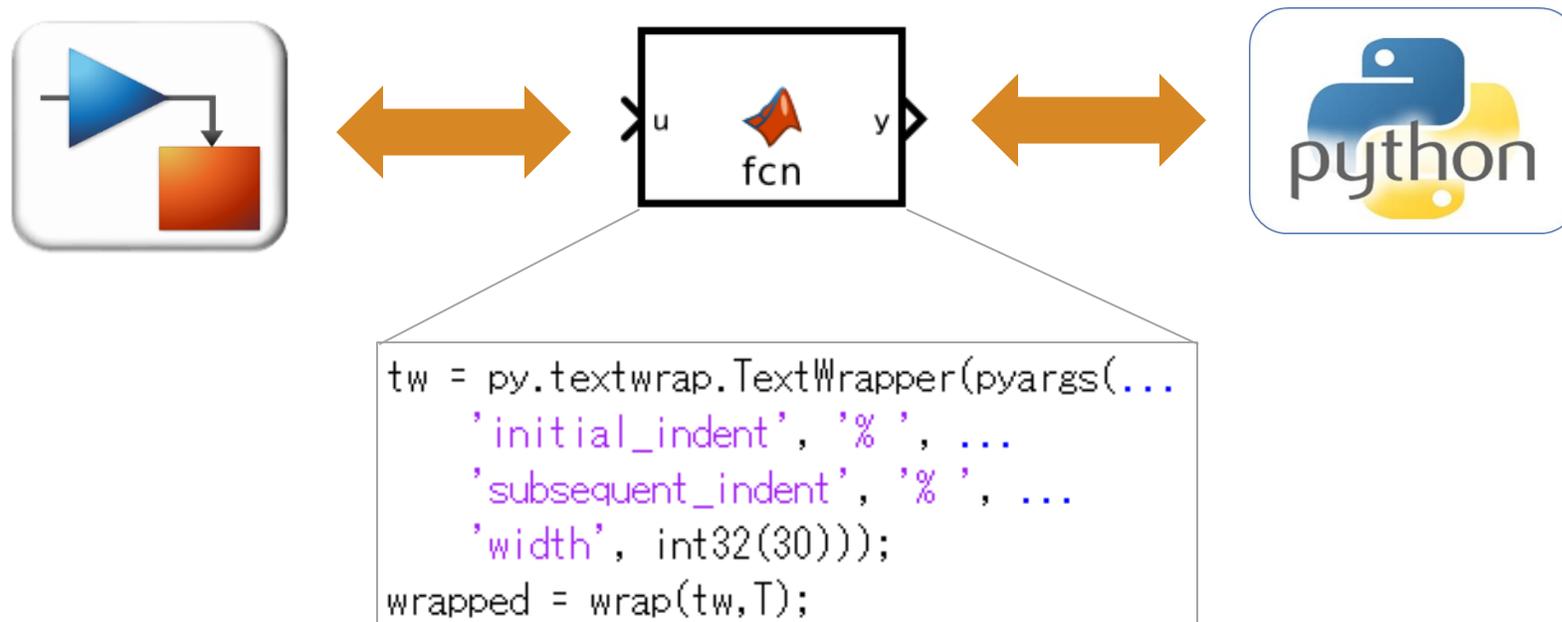
调试自定义代码

- 使用 Visual Studio 调试工具，将 MSVC 作为 MEX 的编译器
- Visual Studio code with
 - MINGW - gdb (Windows)
 - GCC - gdb (Linux)
 - Clang - LLDB(Mac)



与外部程序的结合：MATLAB, Python

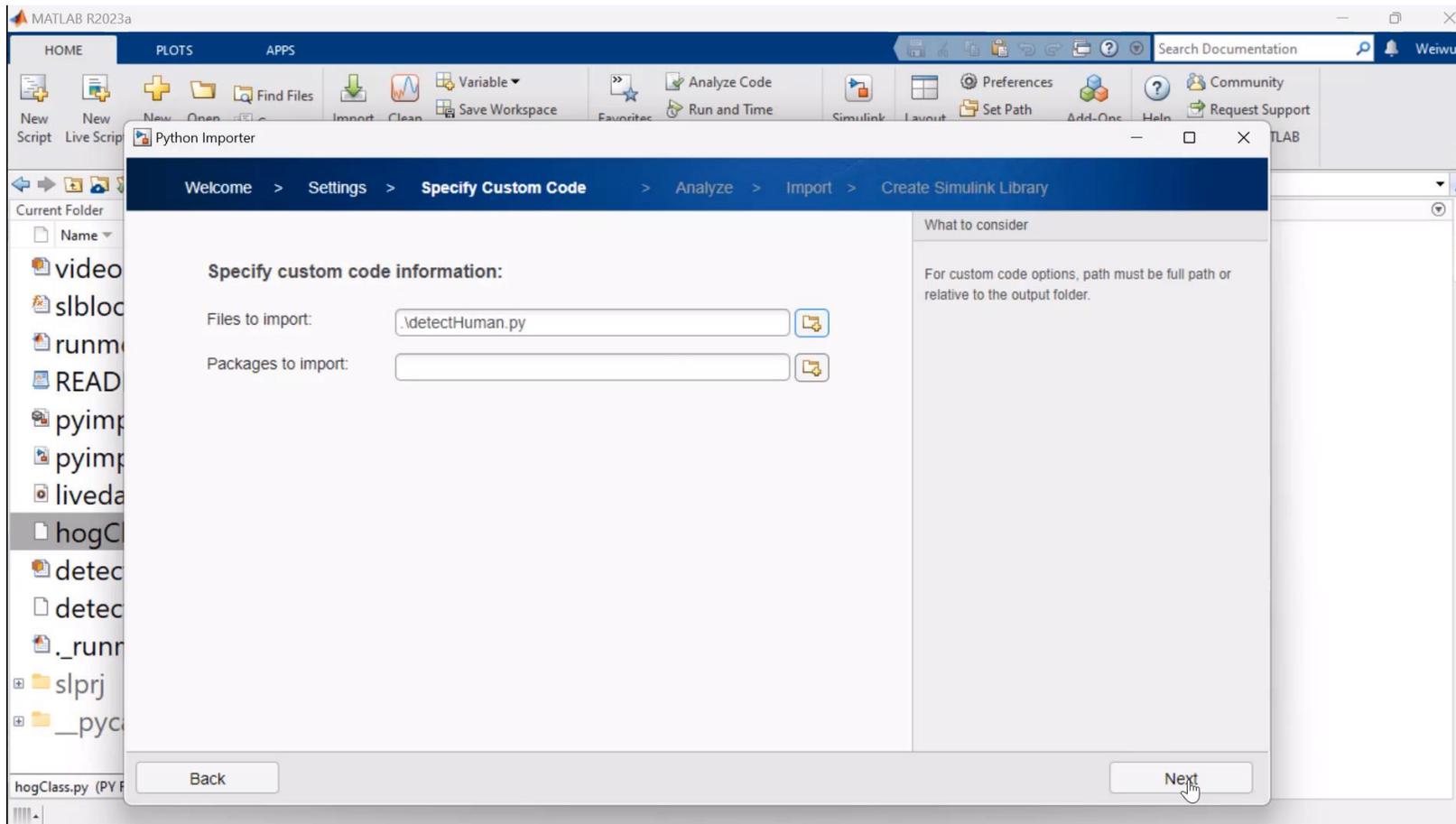
- 使用MATLAB Function、MATLAB System模块，可以在Simulink模型上记述、执行MATLAB程序或调用MATLAB函数
- 通过在同一模块内利用MATLAB的Python库函数调用功能，可以使Simulink模型和Python联合起来。
 - 使用时，需要将MATLAB/Python之间的数据相互转换。



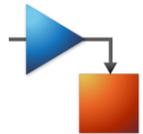
Python 集成 - 使用 Python 导入器从 Simulink 调用 Python 代码



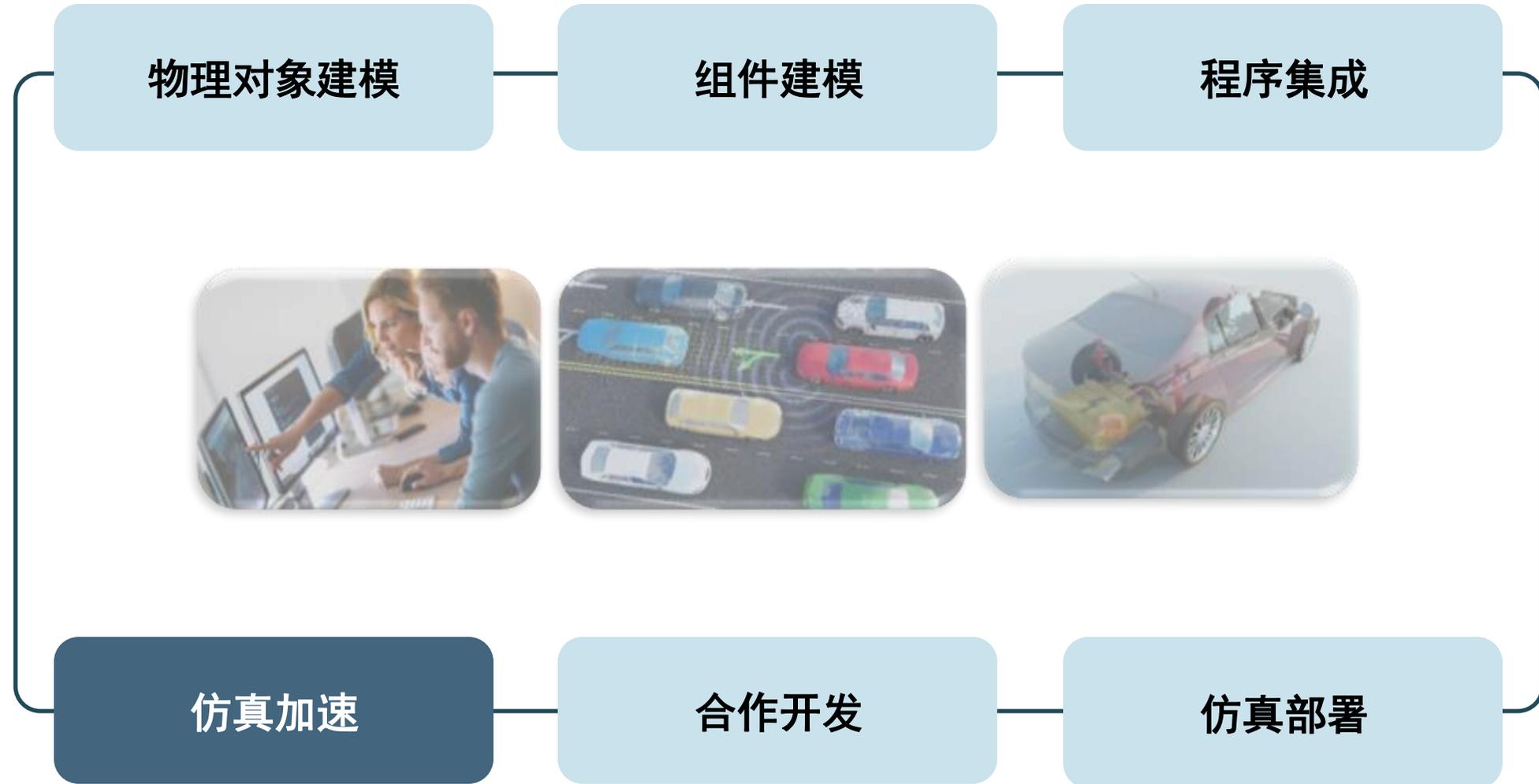
导入



到模块



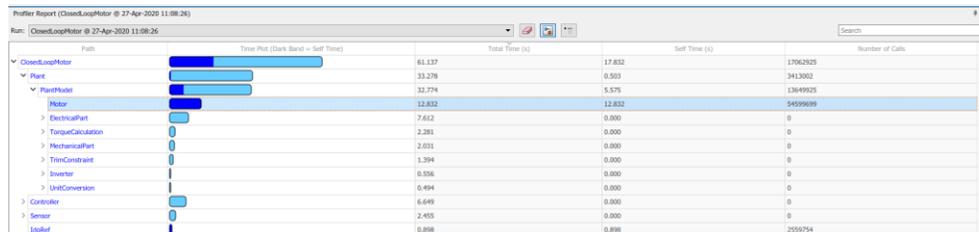
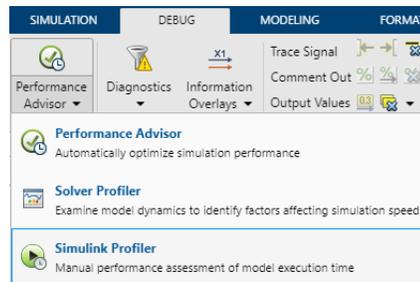
- 图形化向导提供了逐步的操作指导，最小化手工代码编写
- 将单个 Python 函数组成的 Python 函数包集成到一个相关的库模块中
- 方便复用和构建自定义工具集



仿真速度瓶颈分析

■ Simulink Profiler

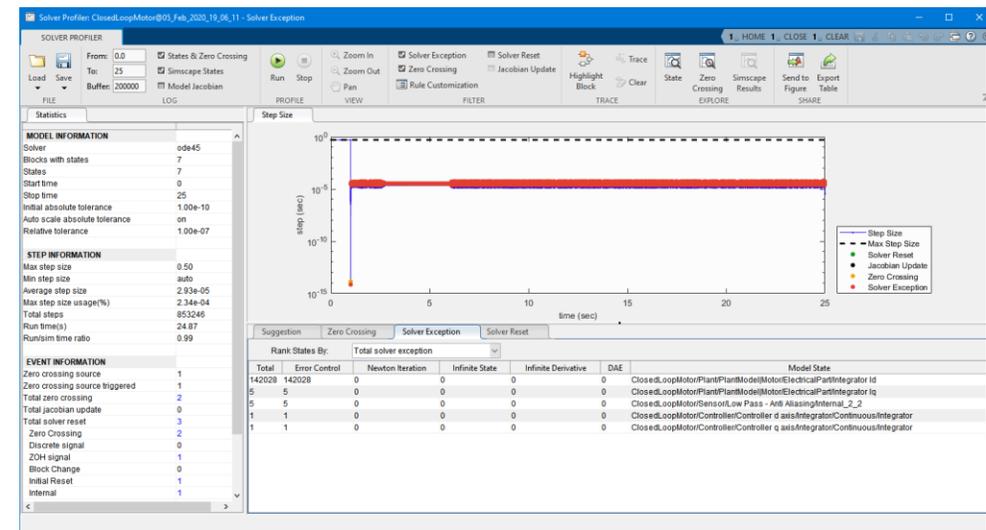
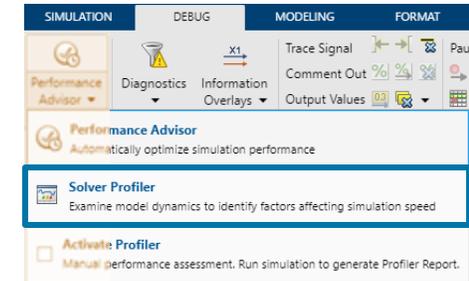
- 确定模型中需要最多时间来仿真的部分。
- 显示执行每个功能所花费的时间（系统和块级别）



<https://ww2.mathworks.cn/help/releases/R2024a/simulink/ug/how-profiler-captures-performance-data.html> -

■ Solver Profiler

- 确定求解器性能瓶颈及其在模型中的位置
- Zero Crossing/Solver Exceptions Events/Solver Reset Events/Jacobian Computation Events

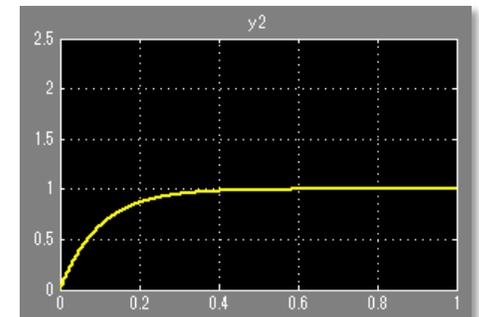
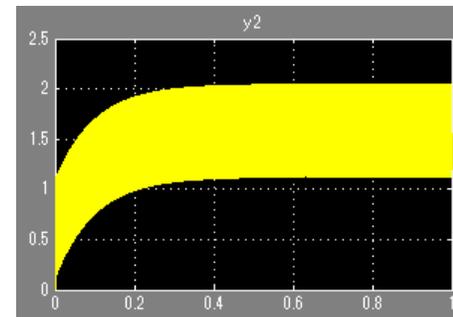
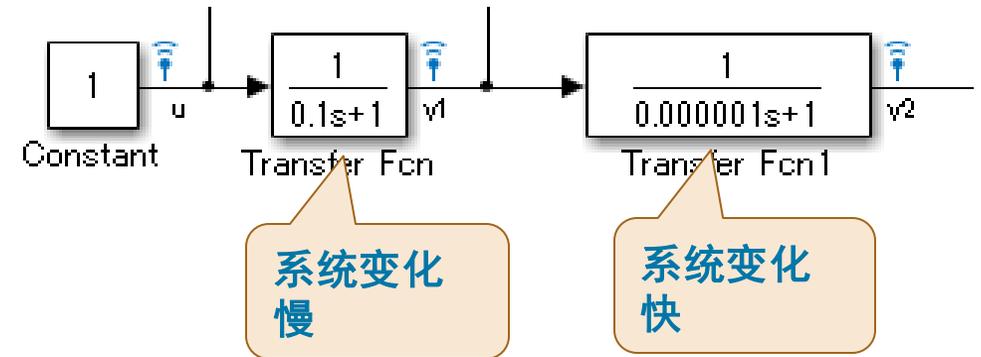


<https://ww2.mathworks.cn/help/releases/R2024a/simulink/ug/examine-solver-behavior-using-solver-profiler.html> -

导致仿真变慢的技术因素 (1)

刚性系统的求解器设置不正确

- 刚性系统是指
 - 包含变化缓慢的部分和变化快速的的部分的系统
 - 包含时常数差异很大的行为的系统。
 - 例如，机械系统与电气系统的混合系统。
 - 刚性系统，如果使用显式求解器，需要减小时间步长才能得到解，从而导致仿真时间长。
- 针对刚性系统的对策
 - 选择隐式求解器
 - 可变步长求解器：求解器名称中标有 "stiff"。
 - 固定步长求解器：ode14x。
- 推荐自动（求解器的自动选择）选项。

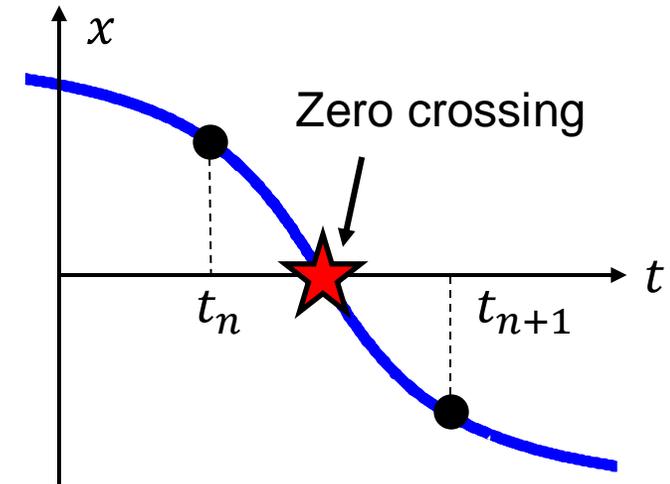


使用 ODE45 求解器对上述模型的仿真结果
左：最大步长 $1e-5$ ，右： $1e-6$

导致仿真变慢的技术因素（2）

频繁的过零检测（抖动）

- 过零检测是什么
 - 增加过零点前后的计算步骤
 - 用于准确确定过零时间
 - 对于模拟特性不连续切换的系统或物理现象非常有用，通常不需要更改设置
 - 如果有很多过零检测，计算速度将大大降低
- 过零检测的对策
 - 检查过零是否符合预期的物理行为。如果不符合，修改模型以解决问题。
 - 考虑 Simulink 文档中提到的对策，推荐使用自适应算法。

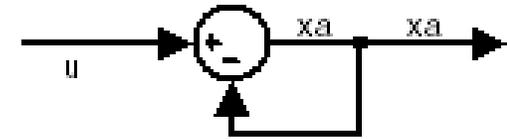


导致仿真变慢的技术因素 (3)

代数环

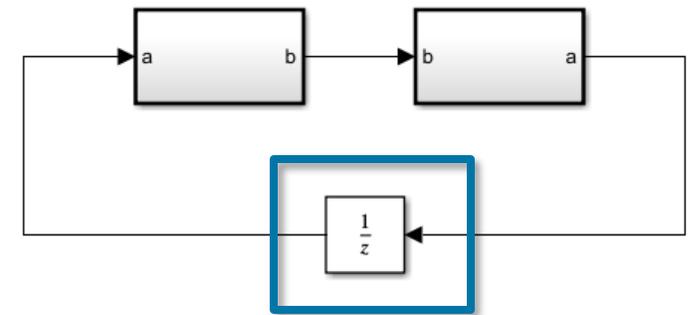
代数环是指？

- 指在计算输出时需要输入的计算循环。右图是一个简单的代数环示例
- 代数环通常出现在由代数微分方程（如守恒定律等）表示的系统中，或者在没有延迟元素的反馈控制系统模型中。
- 由于在每个时刻都需要额外计算来解决代数循环，仿真会变慢
 - 默认的诊断设置会在存在代数循环时显示警告，并指出其所在位置。



代数环的对策

- 直接求解代数方程并建模方程
 - 在右上图中循环的情况下， $x_a = u - x_a$ ，所以 $x_a = u / 2$
- 在循环中插入带有延迟元素的模块
 - 对于离散时间信号，Unit Delay 模块
 - 对于连续时间信号，Memory 模块或 Transfer Function 模块（如一阶迟滞等）



<https://www.mathworks.com/help/releases/R2020b/simulink/ug/algebraic-loops.html>

通过改变整个模型的仿真模式来加速

正常模式

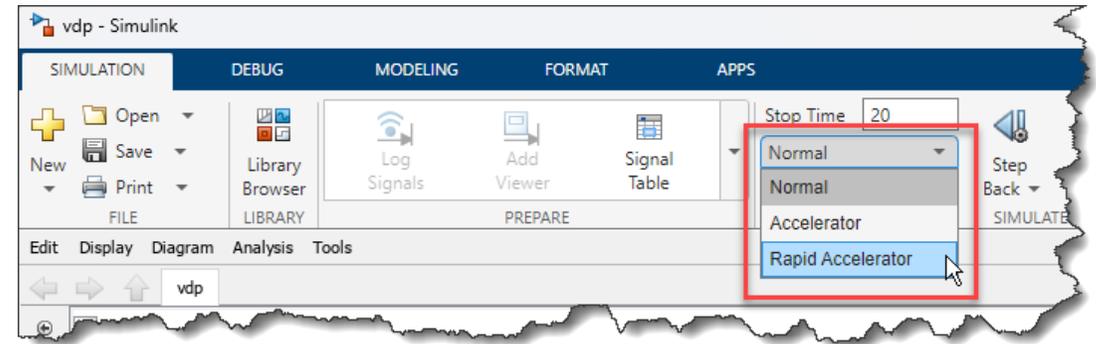
- 正常仿真

加速模式

- 与普通模式相比，仿真速度有望提高2到10倍。
- 不支持运行时诊断

快速加速模式

- 将模型转换为C代码并在独立进程中执行，模拟速度最快
 - 第一次运行时需要时间进行代码生成和构建。
 - 仅适用于可以生成C代码的模型。
- 不支持调试器和分析器。
- 速度可能因C编译器而异（如MinGW比Visual Studio更快等）



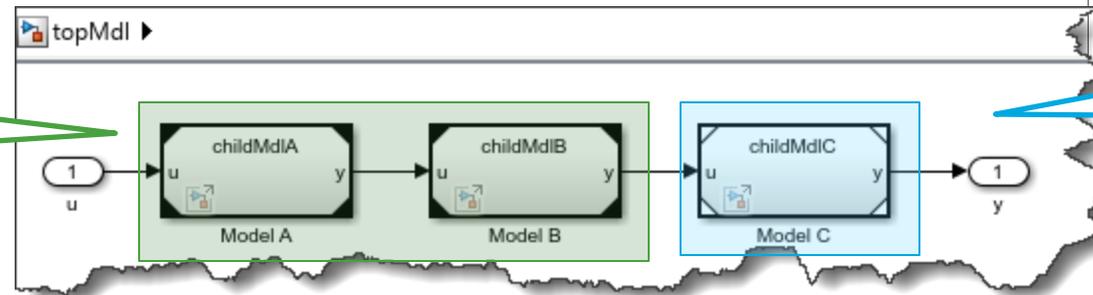
※从 R2016 开始，accelerator 模式使用 JIT 编译器，模型初始化时间（用于生成 C 代码缩短了）。

※从 R2020b 开始，Rapid Accelerator 等模式生成的代码开始利用 SIMD 运算，从而实现了进一步的加速。

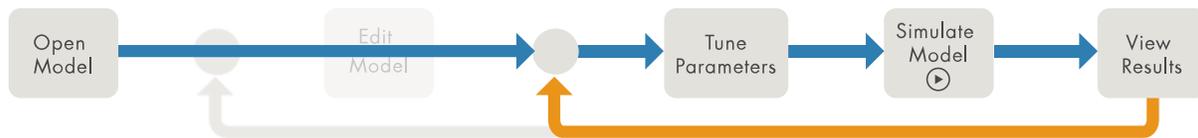
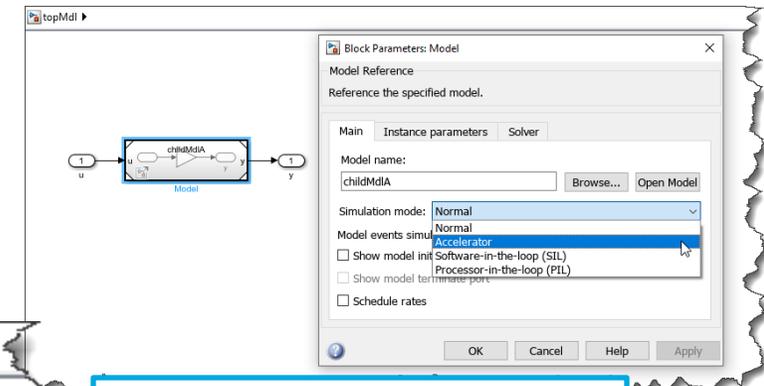
通过更改模型引用的仿真模式来加速。

Model Reference支持在一个仿真中实现多个工作流

我不需要修改模型
A and B



我正在编辑模型C



对未修改的模型应用 **Accelerator** 模式：

- 零初始化时间
- 更快速的仿真



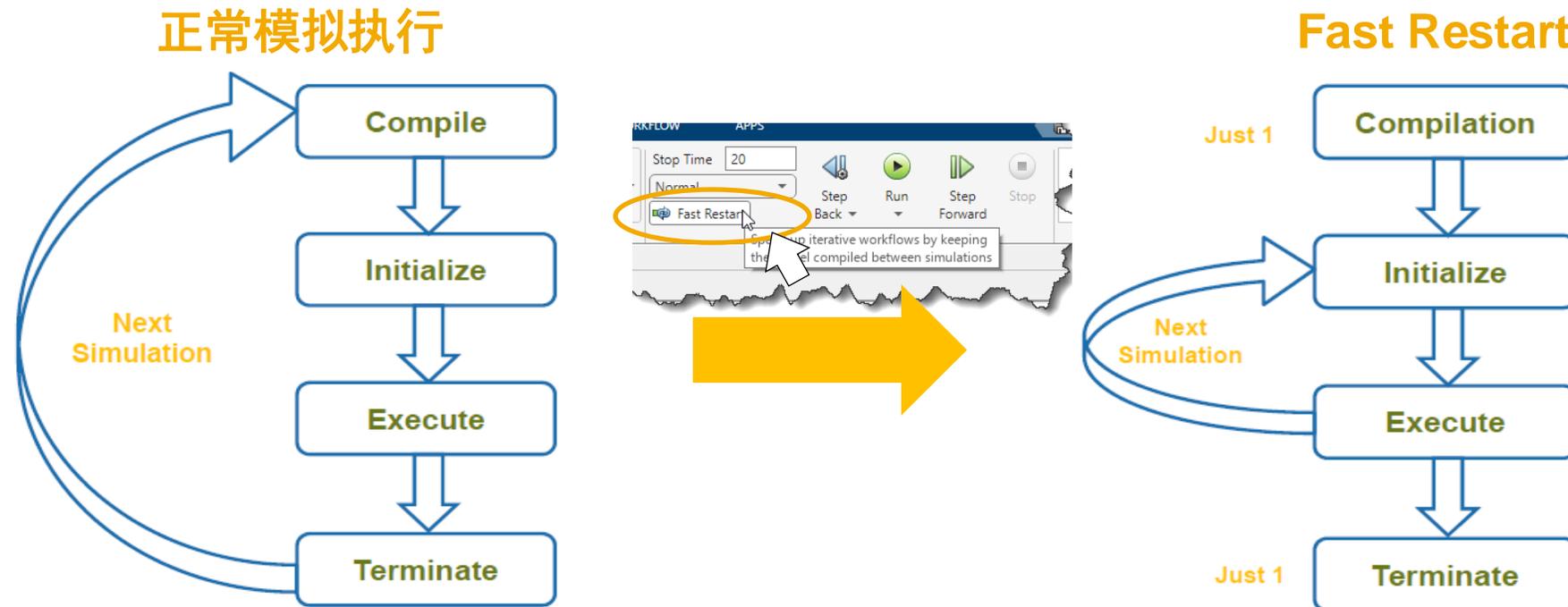
对正在开发的模型应用**Normal**模式：

- 结构更改后可以最快初始化

使用“Fast Restart”实现更快的重复仿真

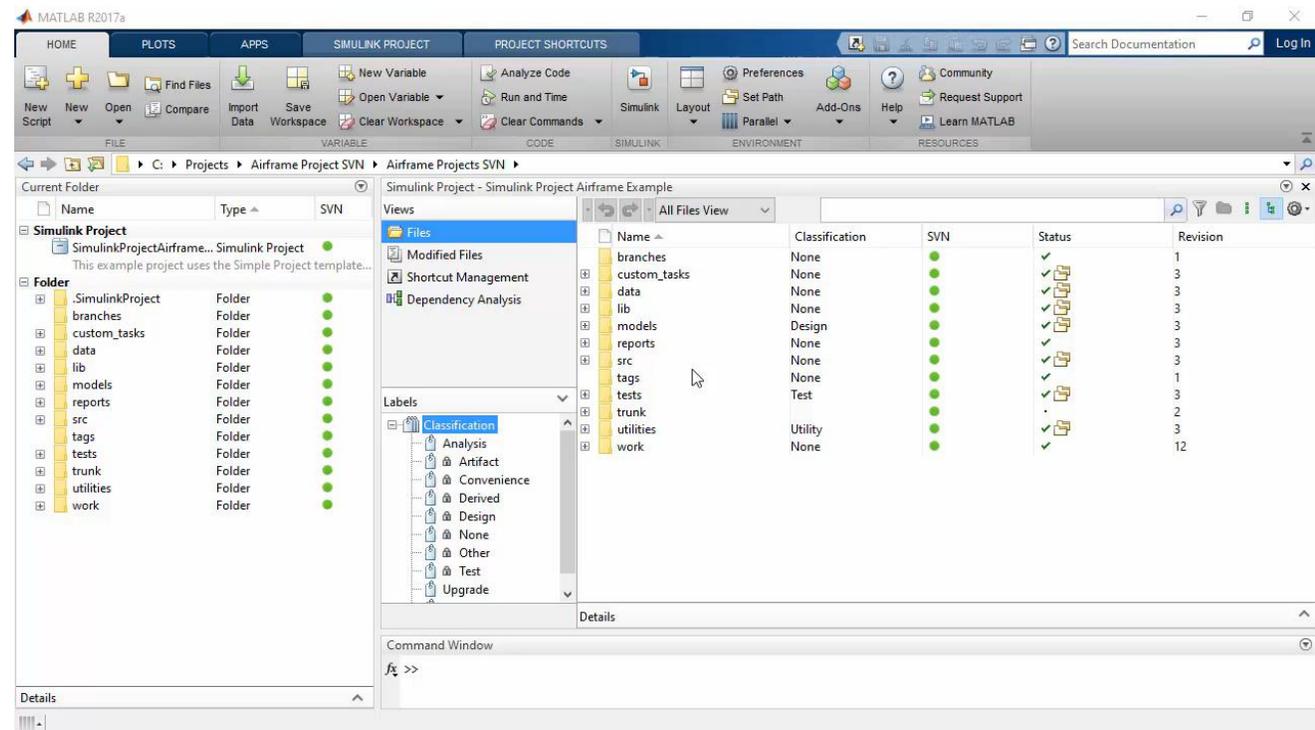
对于优化计算和参数研究非常有效

- 启用后，从第二次模拟开始跳过初始化过程以加速模拟。
- 如果进行参数更改以外的模型修改，则会重新进行初始化。



通过 Simulink 缓存文件（SLXC）缩短初始化时间

- Simulink 缓存文件（扩展名为 slxc）在模型更新时自动创建。
- 缓存中存储了各个目标的构建生成物，以防止模型重新构建。
- 可以复制缓存文件以在不同环境中重用。

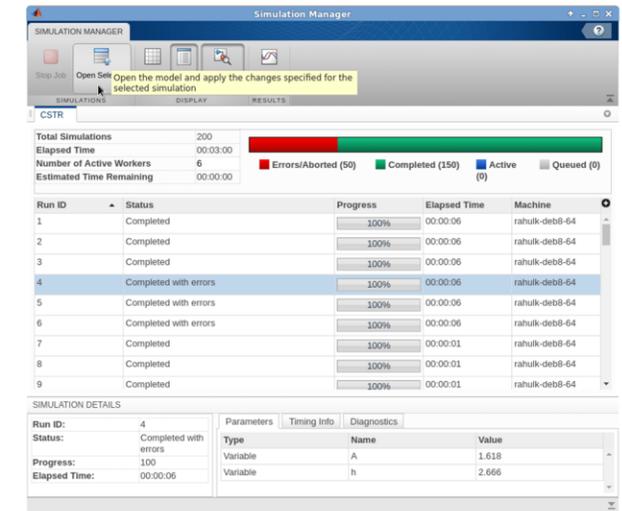
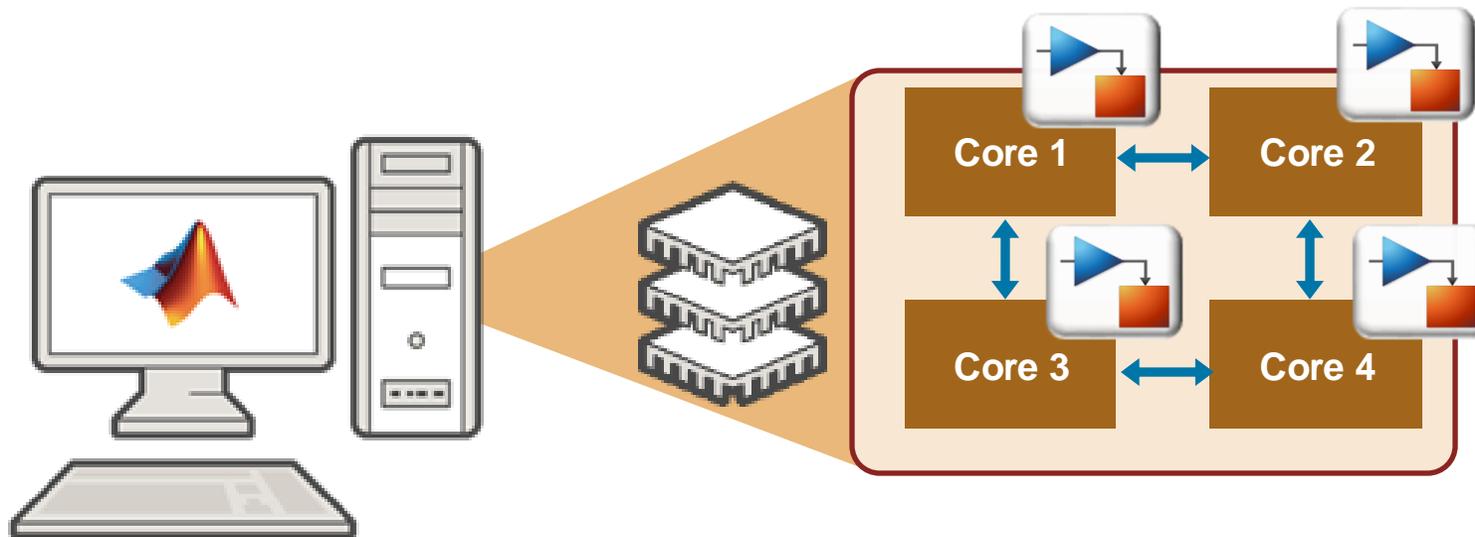


通过并行和分布式处理减少计算时间

加速多个仿真，例如优化

- **parsim**
 - 比传统的 parfor 仿真更方便的并行和分布式仿真执行命令。
 - 执行命令的 MATLAB 会话将变为忙碌状态。
- **batchsim**
 - 用于离线执行并行和分布式仿真的命令。
 - MATLAB 会话不会变为忙碌状态。

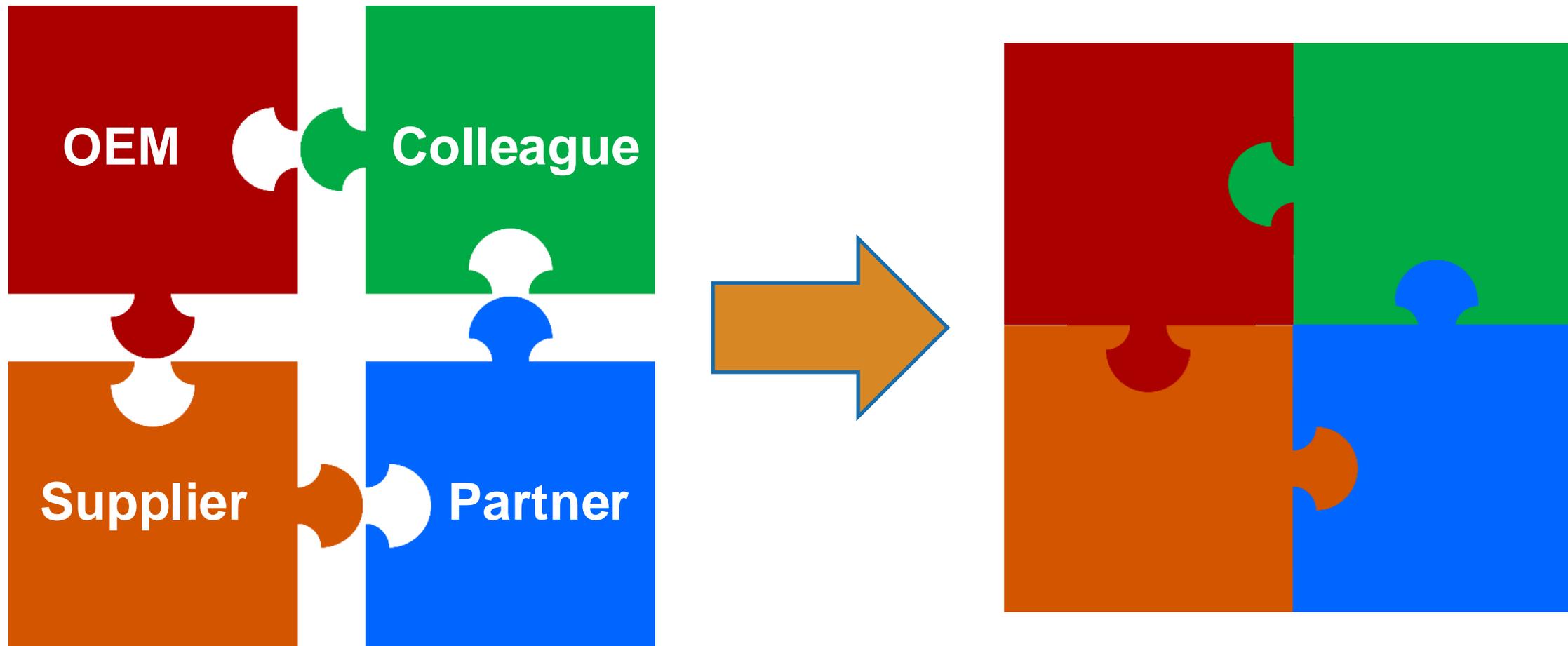
※并行处理需要 Parallel Computing Toolbox, 分布式处理需要 MATLAB Parallel Server.





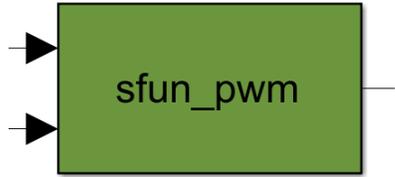
合作开发

- 各个团队模型的IP 以多种形式存在于许多位置，使集成变得困难



合作开发保护IP方法

- 单平台仿真IP保护



S-Function



Protected
Model

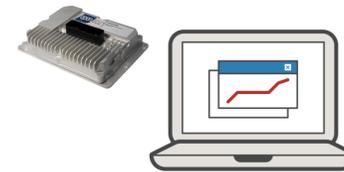


FMU

- 分布式仿真方法



使用ROS网络
节点



桌面Simulink
外接通讯



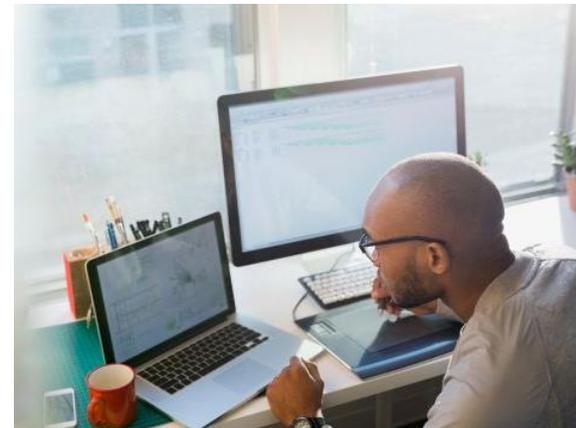
硬件在环

模型保护

- 创作者希望在不暴露知识产权的情况下，与接收者共享模型以进行仿真



Creator



Recipient



Protect



Protect



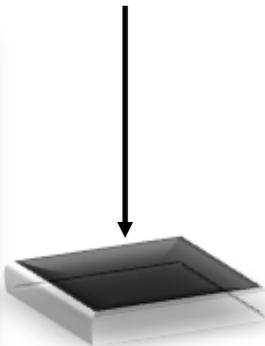
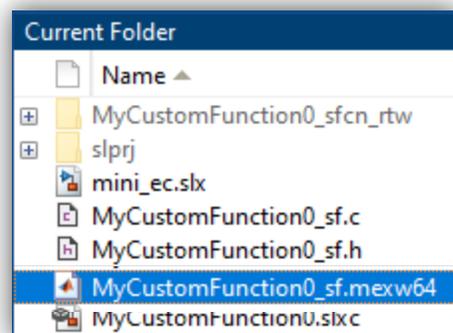
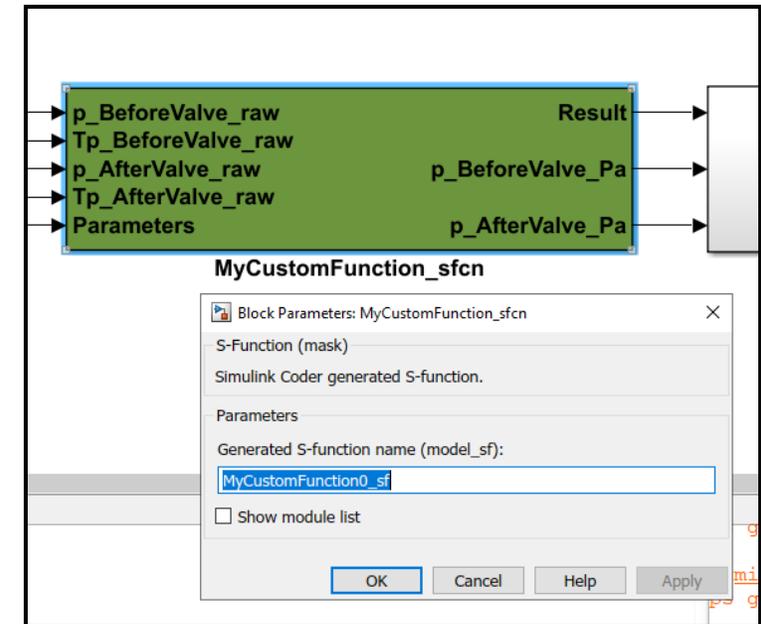
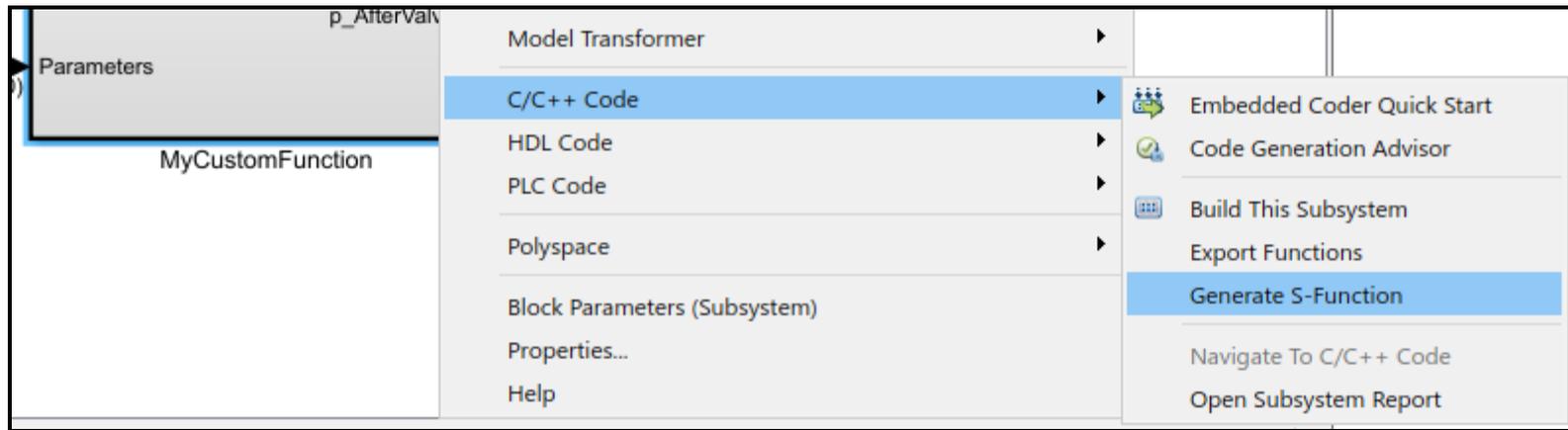
使用S-function 进行模型保护



Creator



Recipient



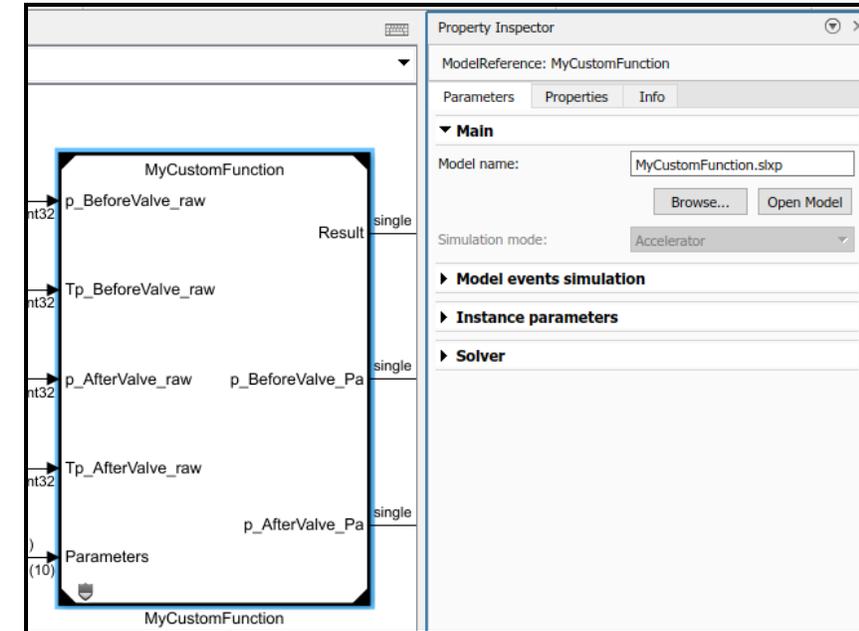
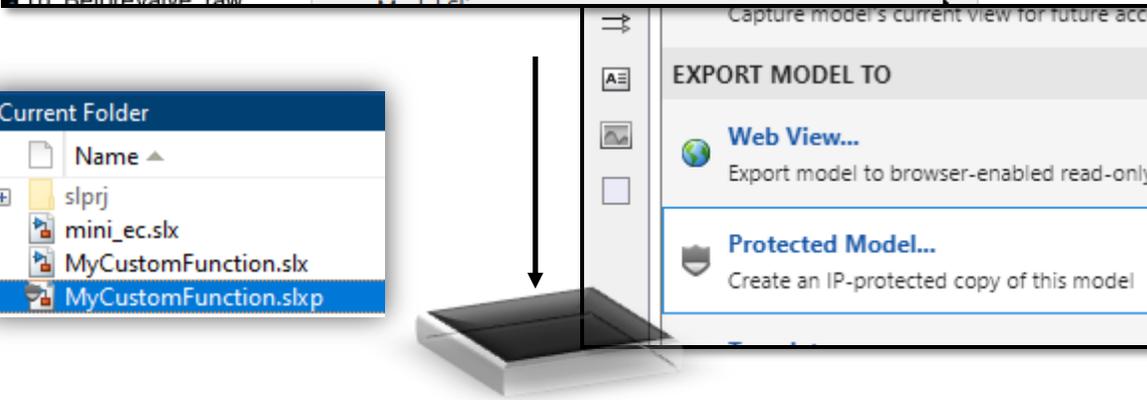
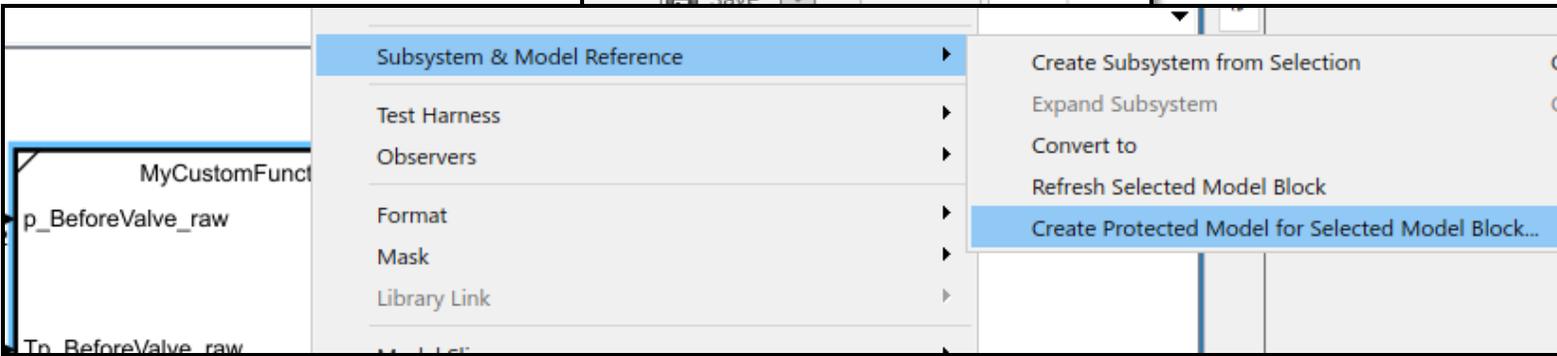
使用受保护模型



Creator



Recipient



从 Simulink 模型导出独立 FMU

生成的 FMU 可以导入回 Simulink 中

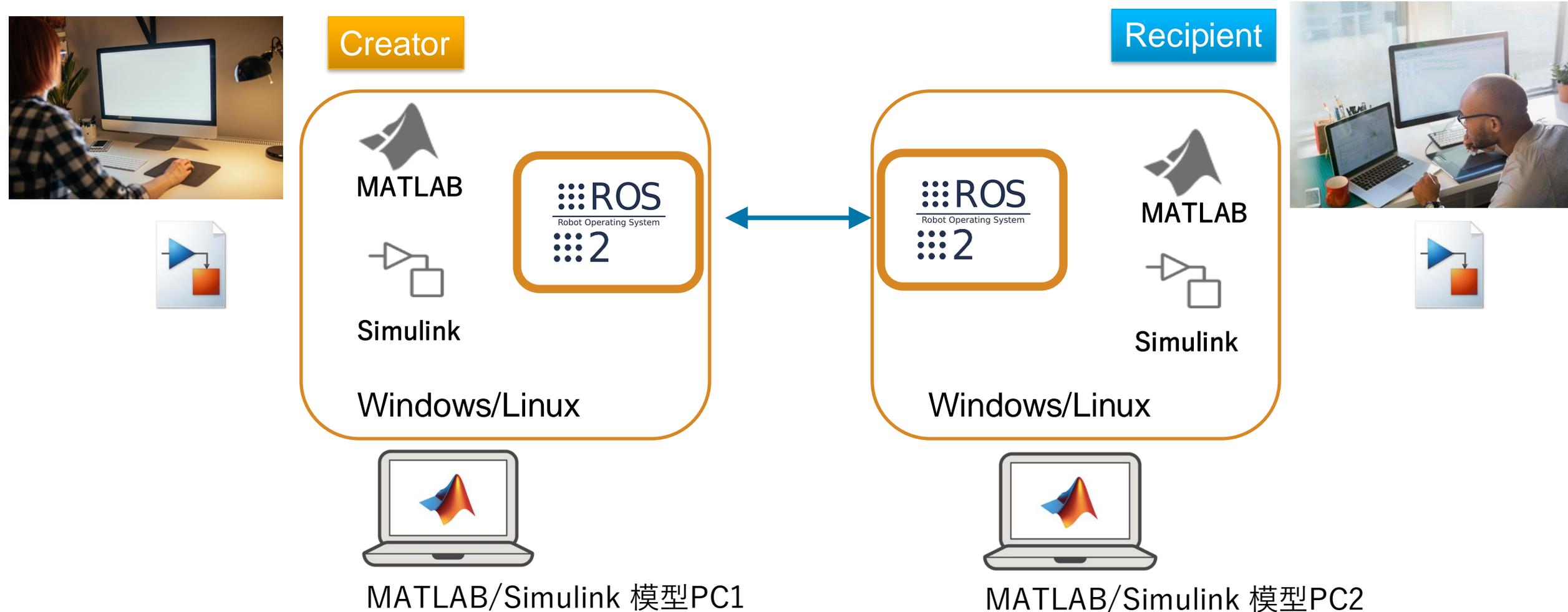
Parameter	Value	Unit	Description
PumpCon [1,1]	-0.0107684200000001		PumpCon.Value[1,1]: PumpCon.Value[1,1]
PumpCon [1,2]	0.03075046		PumpCon.Value[1,2]: PumpCon.Value[1,2]
PumpCon [1,3]	0.052019292		PumpCon.Value[1,3]: PumpCon.Value[1,3]
PumpCon [1,4]	0.06570069999999999		PumpCon.Value[1,4]: PumpCon.Value[1,4]
PumpCon [1,5]	0.07607053999999999		PumpCon.Value[1,5]: PumpCon.Value[1,5]
PumpCon [1,6]	0.084118476		PumpCon.Value[1,6]: PumpCon.Value[1,6]
PumpCon [1,7]	0.09144319599999999		PumpCon.Value[1,7]: PumpCon.Value[1,7]
PumpCon [1,8]	0.09761330799999998		PumpCon.Value[1,8]: PumpCon.Value[1,8]
PumpCon [1,9]	0.1032759		PumpCon.Value[1,9]: PumpCon.Value[1,9]
PumpCon [1,10]	0.10860226		PumpCon.Value[1,10]: PumpCon.Value[1,10]
PumpCon [1,11]	0.11345282		PumpCon.Value[1,11]: PumpCon.Value[1,11]
PumpCon [1,12]	0.11813420666666667		PumpCon.Value[1,12]: PumpCon.Value[1,12]
PumpCon [1,13]	0.12264642000000002		PumpCon.Value[1,13]: PumpCon.Value[1,13]

运行时可调参数保留在生成的 FMU 中

生成的 FMU 可以导入到其他仿真工具中。查看完整列表，请访问：
<https://fmi-standard.org/tools/>



通过ROS网络的协作



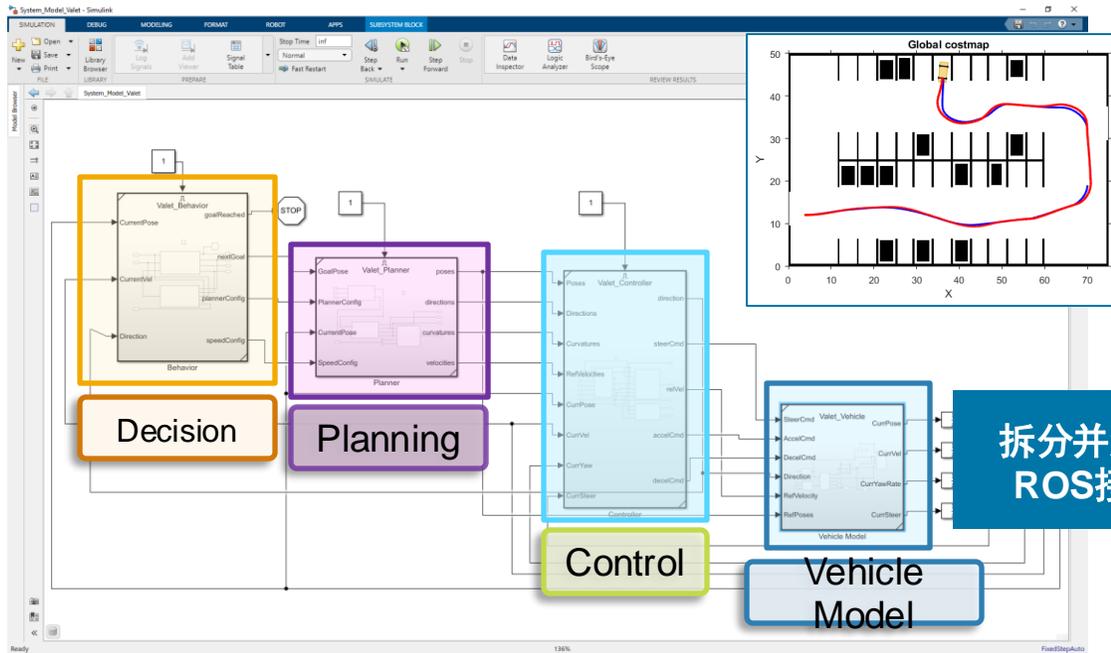
ROS = Robot Operating System,也被用于自动驾驶仿真和项目开发环境

应用实例 自动泊车

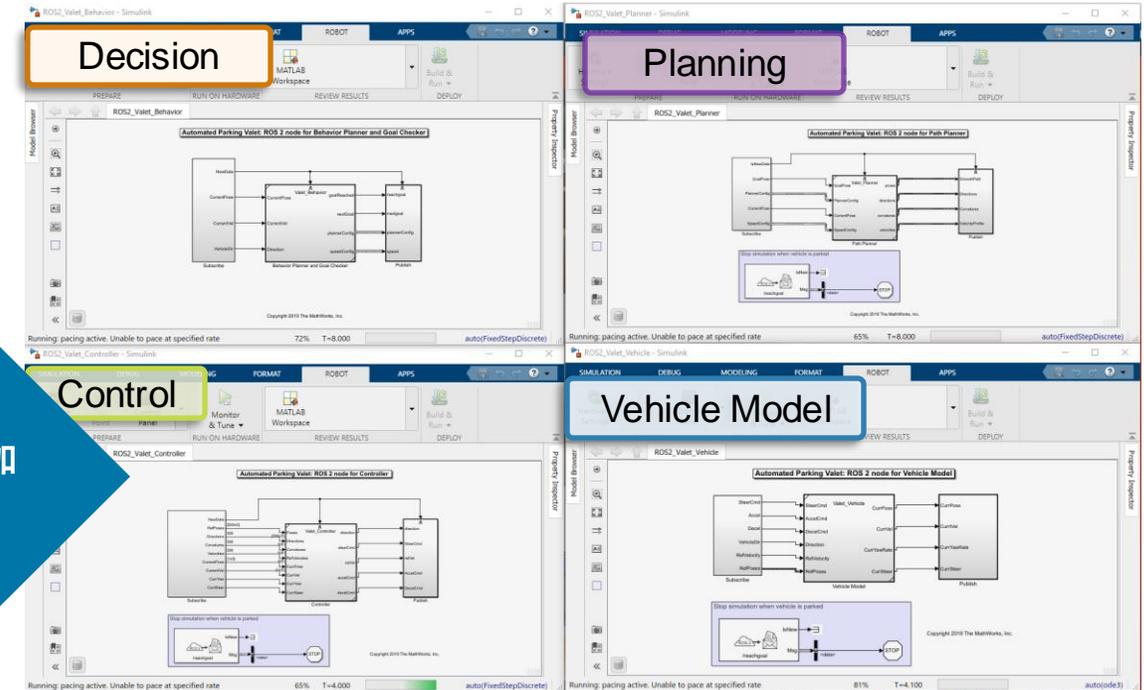
ROS Toolbox
Automated Driving Toolbox™
Simulink Coder™
Embedded Coder™

- 作为 ROS/ROS2 节点实现自动泊车的示例
- 在 ROS 2 网络上测试模型
- 生成 C++ ROS 2 节点以快速执行

[Automated Parking Valet with ROS in MATLAB](#)
[Automated Parking Valet with ROS in Simulink](#)
[Automated Parking Valet with ROS 2 in MATLAB](#)
[Automated Parking Valet with ROS 2 in Simulink](#)



拆分并添加
ROS接口



每个算法串行执行

原始仿真模型

<https://www.mathworks.com/help/ros/ug/ros-automated-valet.html>

ROS 2 发布/订阅,
作为单个节点的分布式执行

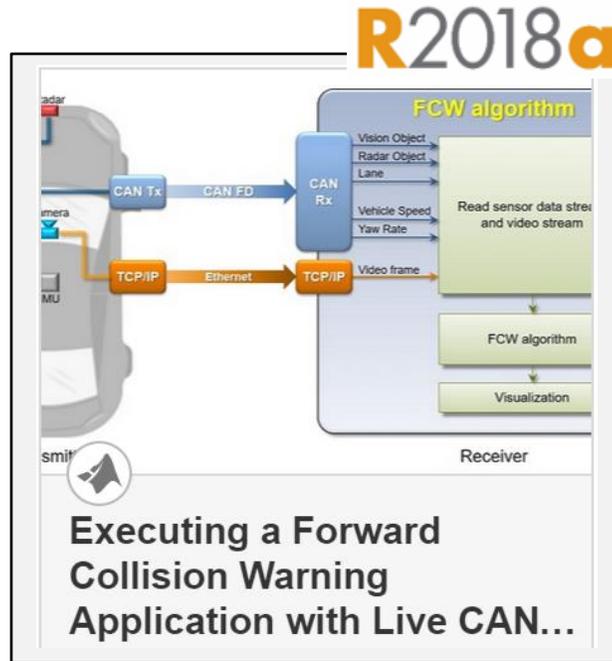
Vehicle Network Toolbox

提供对车辆测试所需的通信协议接口文件的访问

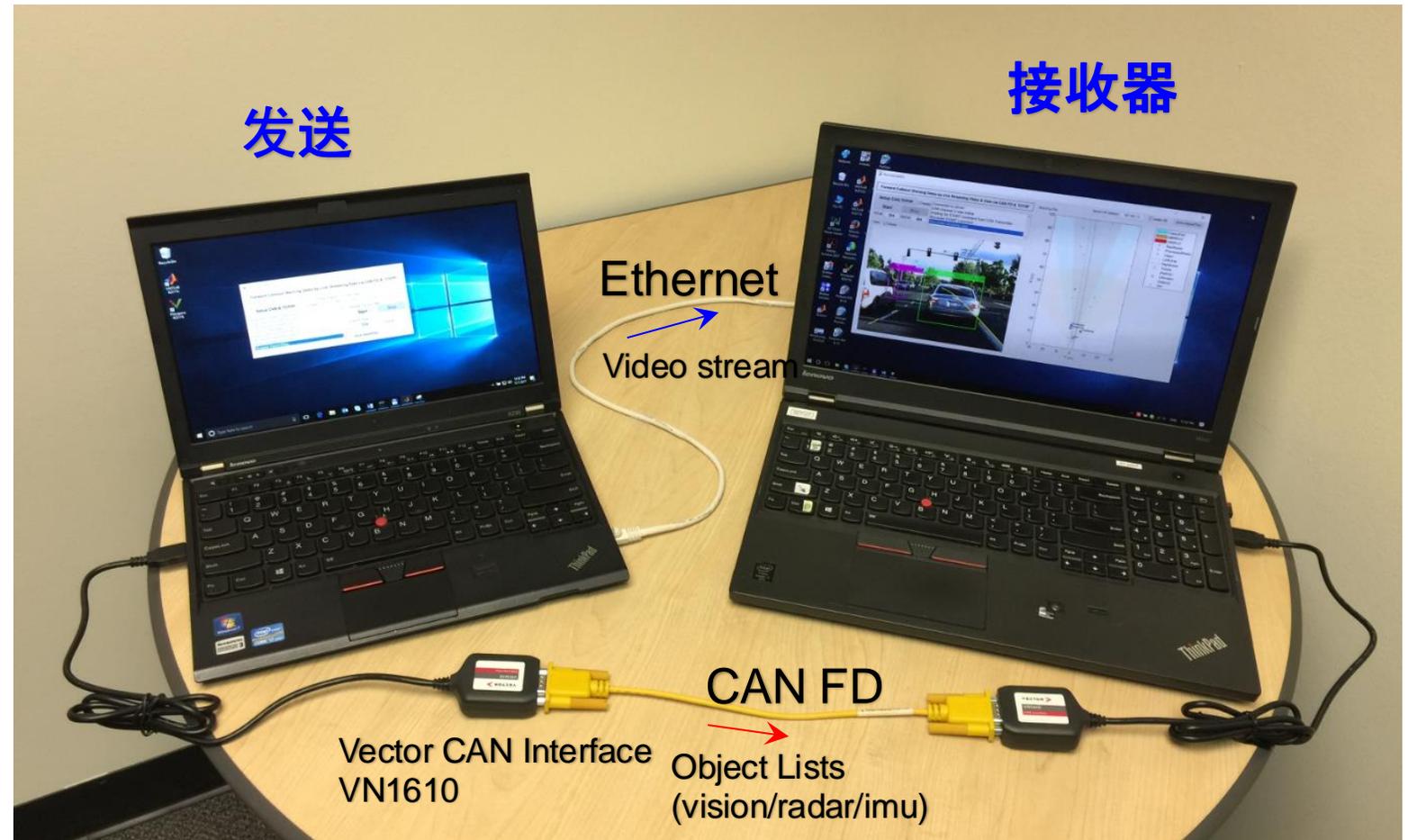


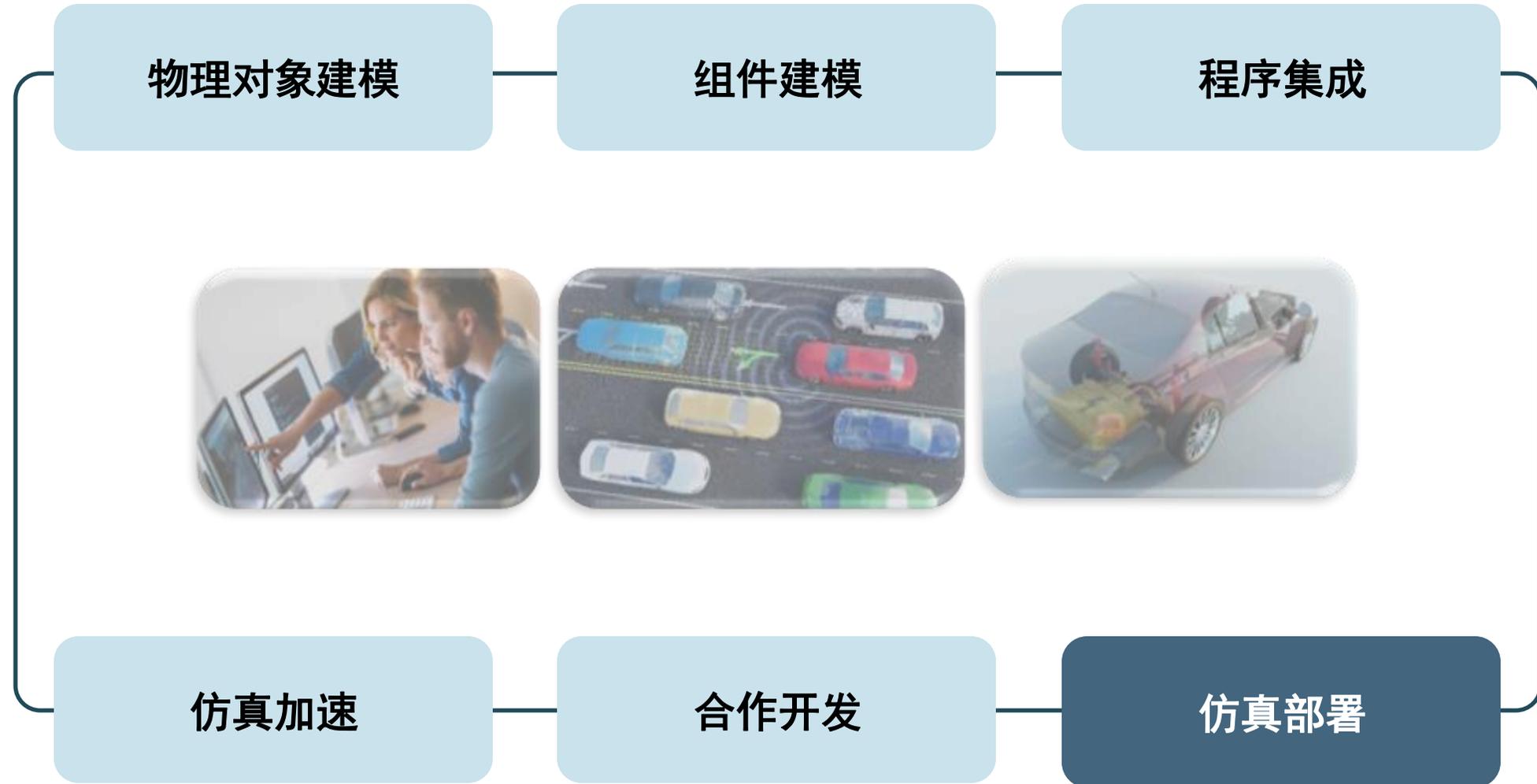
使用CAN FD数据开发融合算法

Automated Driving System Toolbox™
Instrument Control Toolbox™
Vehicle Network Toolbox™

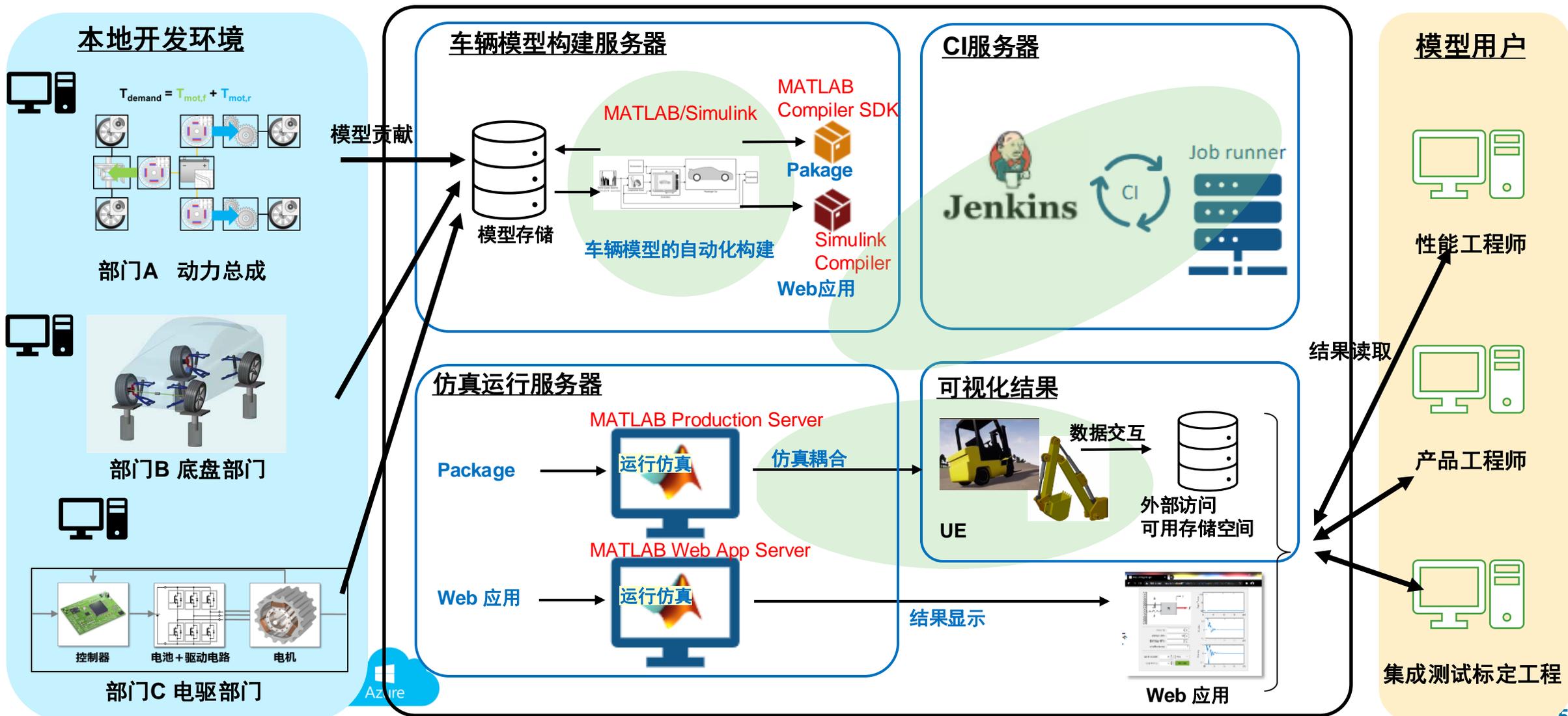


- **CAN FD 数据流和原型设计算法**





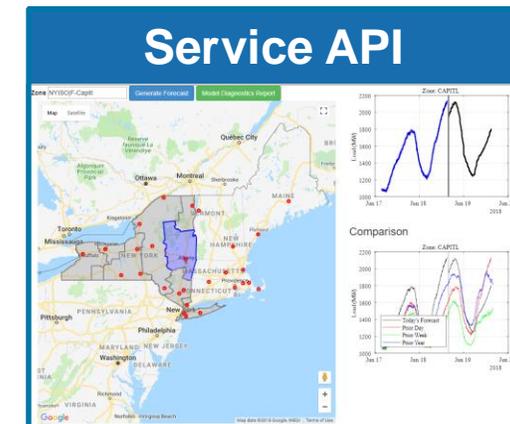
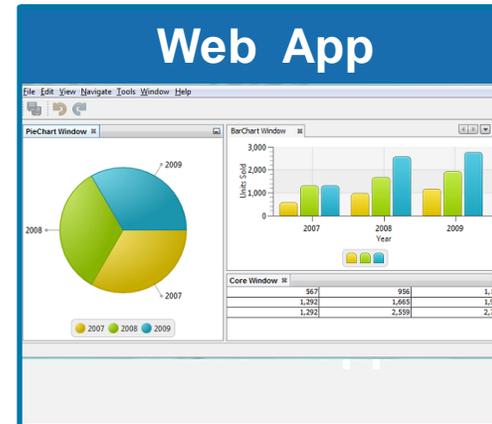
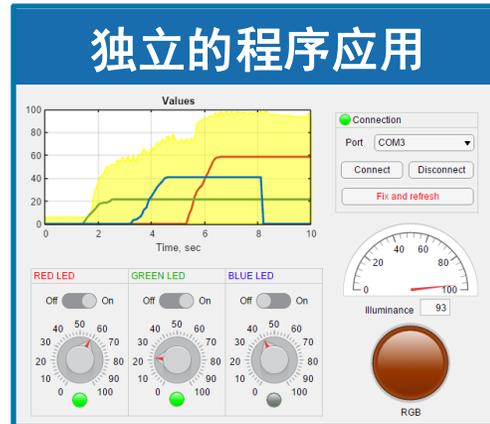
大规模仿真环境



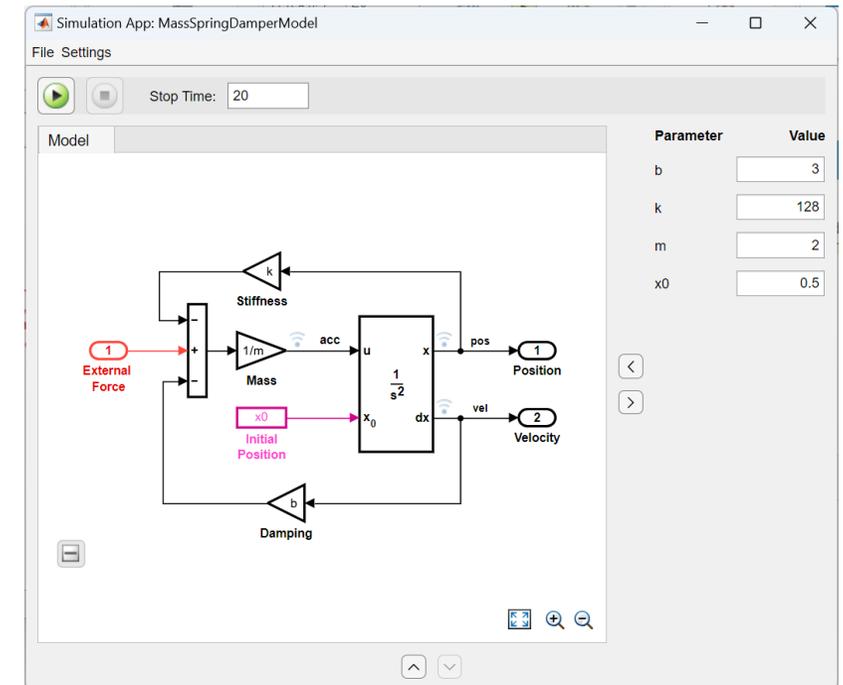
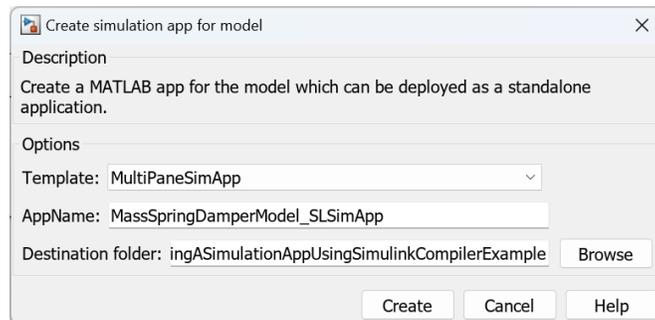
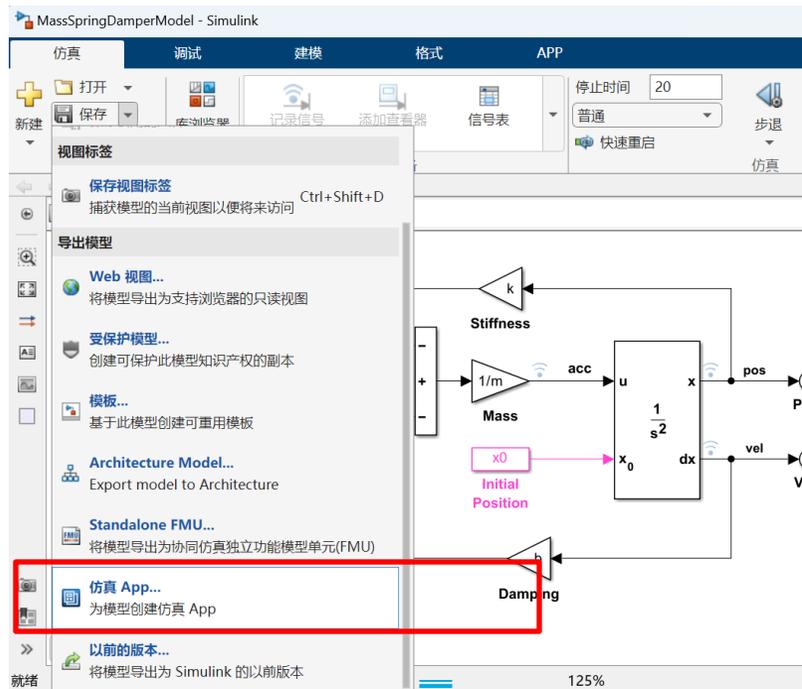
让模型脱离MATLAB环境运行

基于Simulink Compiler的模型应用

- 以可执行文件的形式发布Simulink仿真
- 将仿真集成到各种企业级生产应用程序中，供大量用户访问。
- 基于浏览器的 Web App
- 独立的仿真应用程序
- 支持跨系统编译

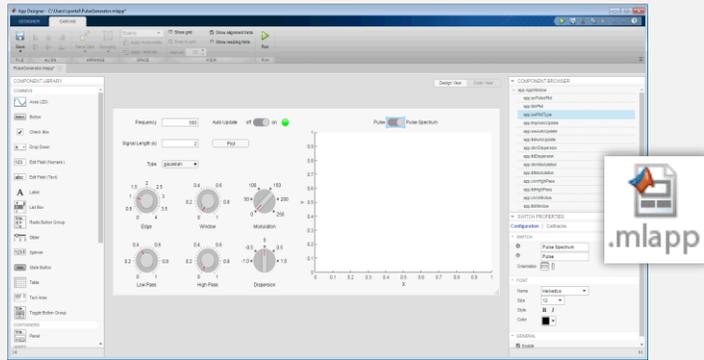


Simulink 自动生成MATLAB App

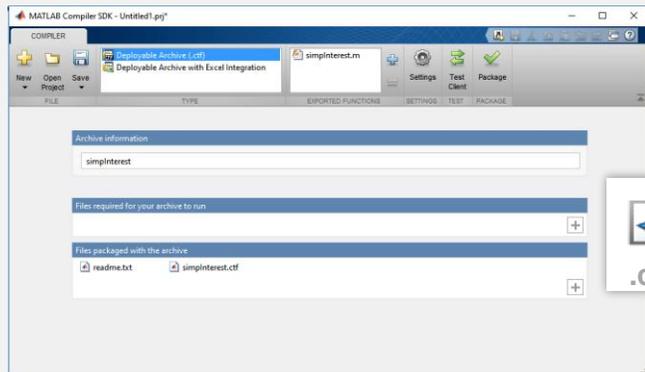
R2022b

Web 部署

Authoring

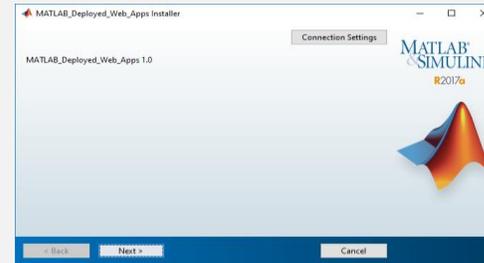


1 使用MATLAB App Designer创建 Application



2 作为 web app分享应用 (创建 .ctf)

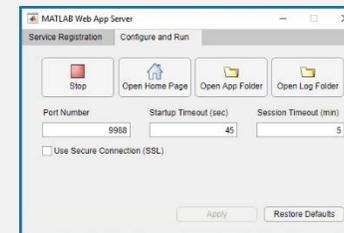
One-time server installation



3 任何人可以安装 Web Apps server



4 Copy .ctf to Web Apps server



5 管理 server 如start/stop

Access

6 通过浏览器运行web app 程序(IE 11 or Chrome)





总结： 基于Simulink平台的综合仿真环境

跨物理域的对象建模
跨工具平台及程序

Simulink可以通过整合系统来仿真整个系统



2024 MathWorks 中国汽车年会

Thank you

