MATLAB EXPO 2021

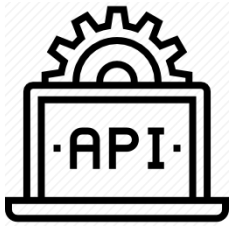
将外部仿真组件集成到Simulink平台中

董 淑成，MathWorks

MathWorks

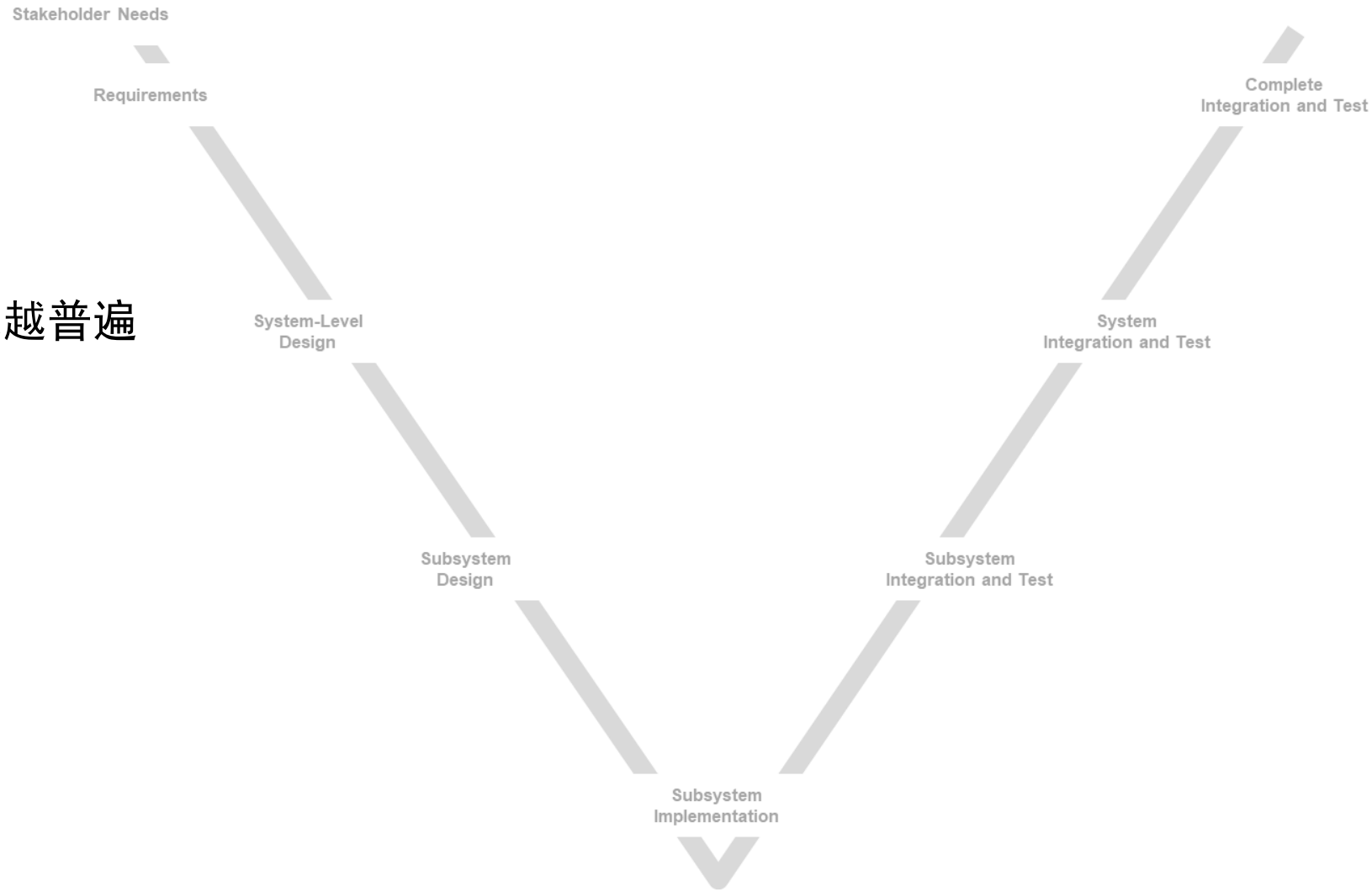
核心要点

Simulink 是开放的集成仿真平台，用于复杂多样的多域物理仿真。



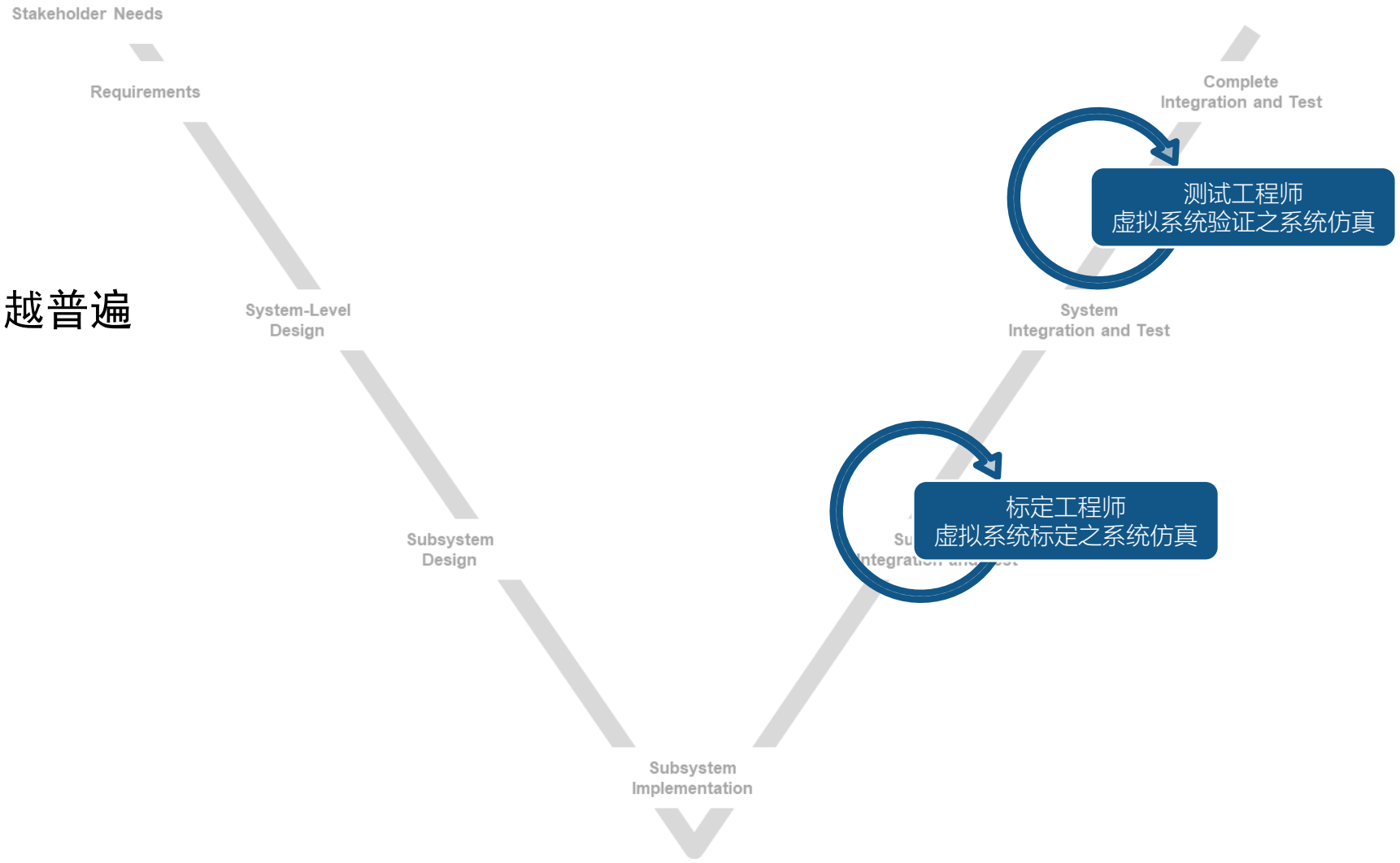
联合仿真动机所在

产品开发中，系统级仿真越来越普遍



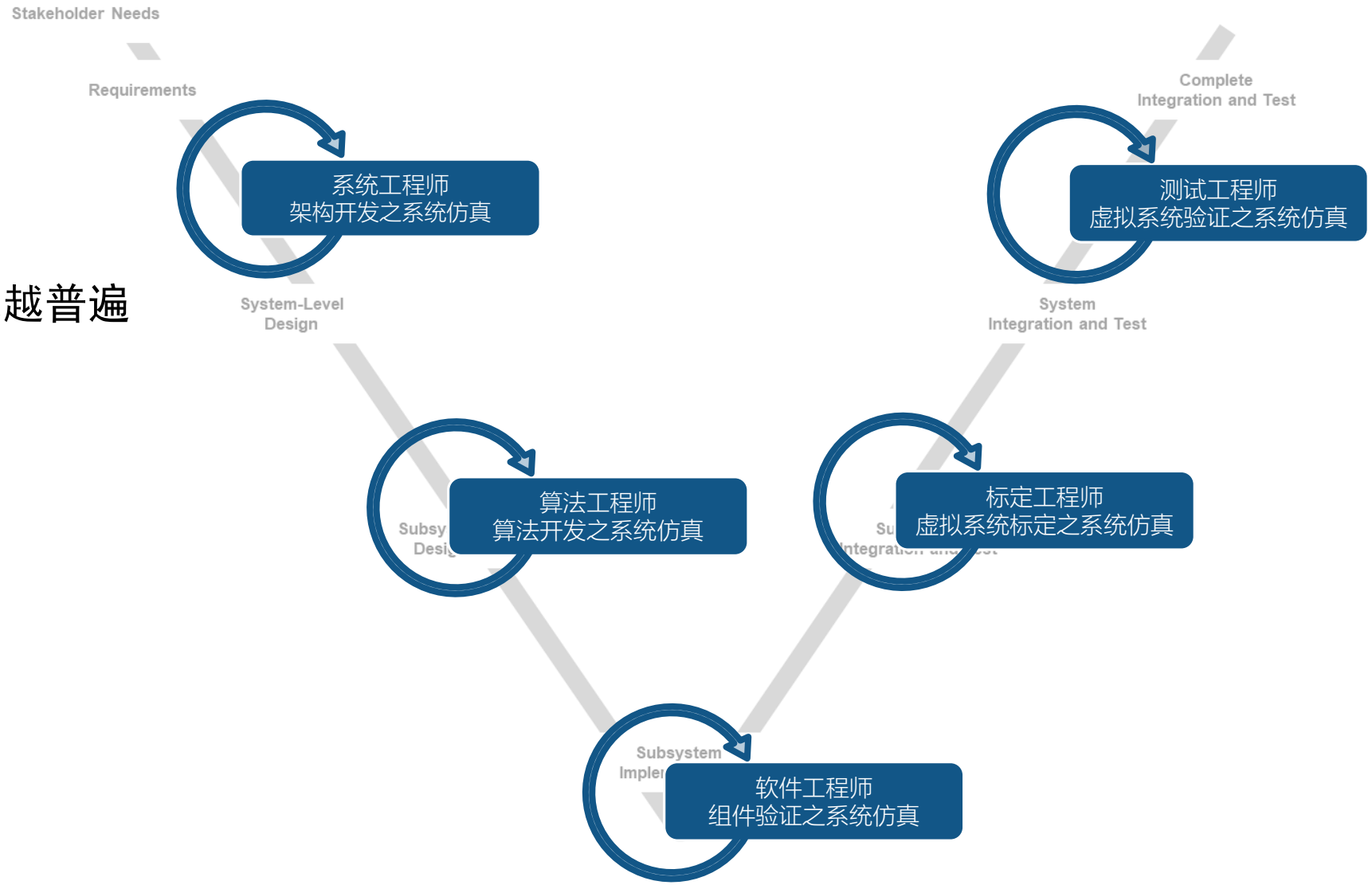
联合仿真动机所在

产品开发中，系统级仿真越来越普遍



联合仿真动机所在

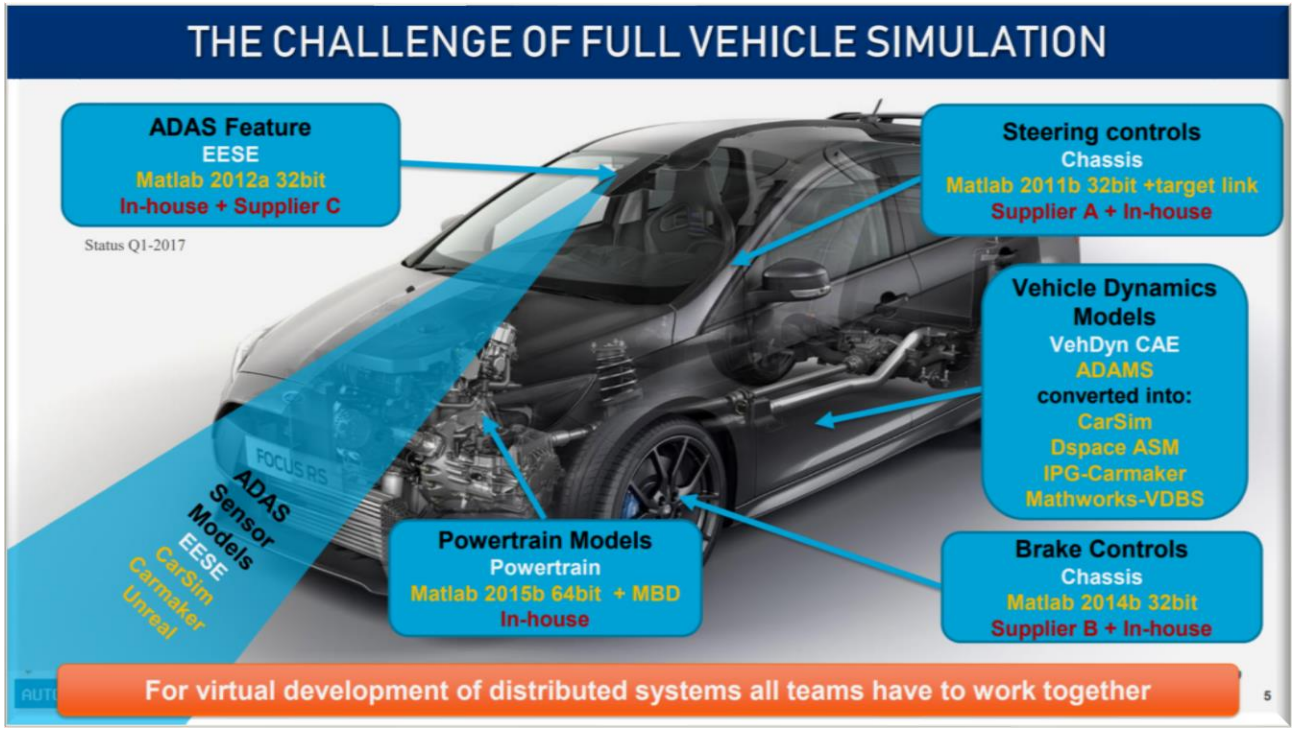
产品开发中，系统级仿真越来越普遍



原因所在

复杂系统仿真的挑战日渐增加

- 多物理域，跨学科设计
- 在不同供应商、客户和协作开发中，重用模型并隐藏详细设计
- 加速系统认知，必须加速仿真



[Model-Based Agility with Ford Automated System Simulation Toolchain (FASST)]

MathWorks 2020年汽车年会

演讲提纲

与外部仿真工具的接口

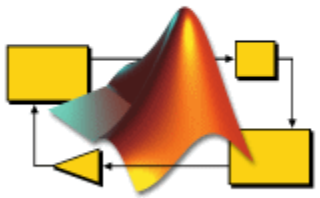
联合仿真的鲁棒性

引入自定义C/C++代码

系统级仿真加速

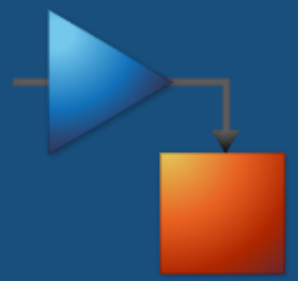
与外部仿真工具的接口

SIMULINK Enabled



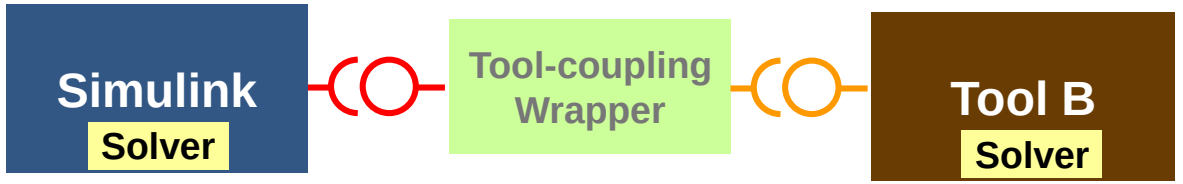
MathWorks Partner

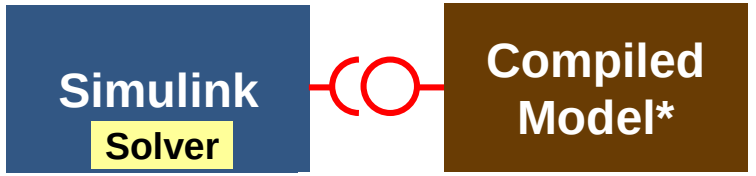
fmi Functional Mock-Up Interface



白盒集成和黑盒集成

- 白盒集成 (工具交联)
  - 在仿真时，Simulink和外部工具均运行
- 黑盒集成 (编译模型)
  - 在仿真时，仅 Simulink运行
  - 第三方模型是Simulink中的组件

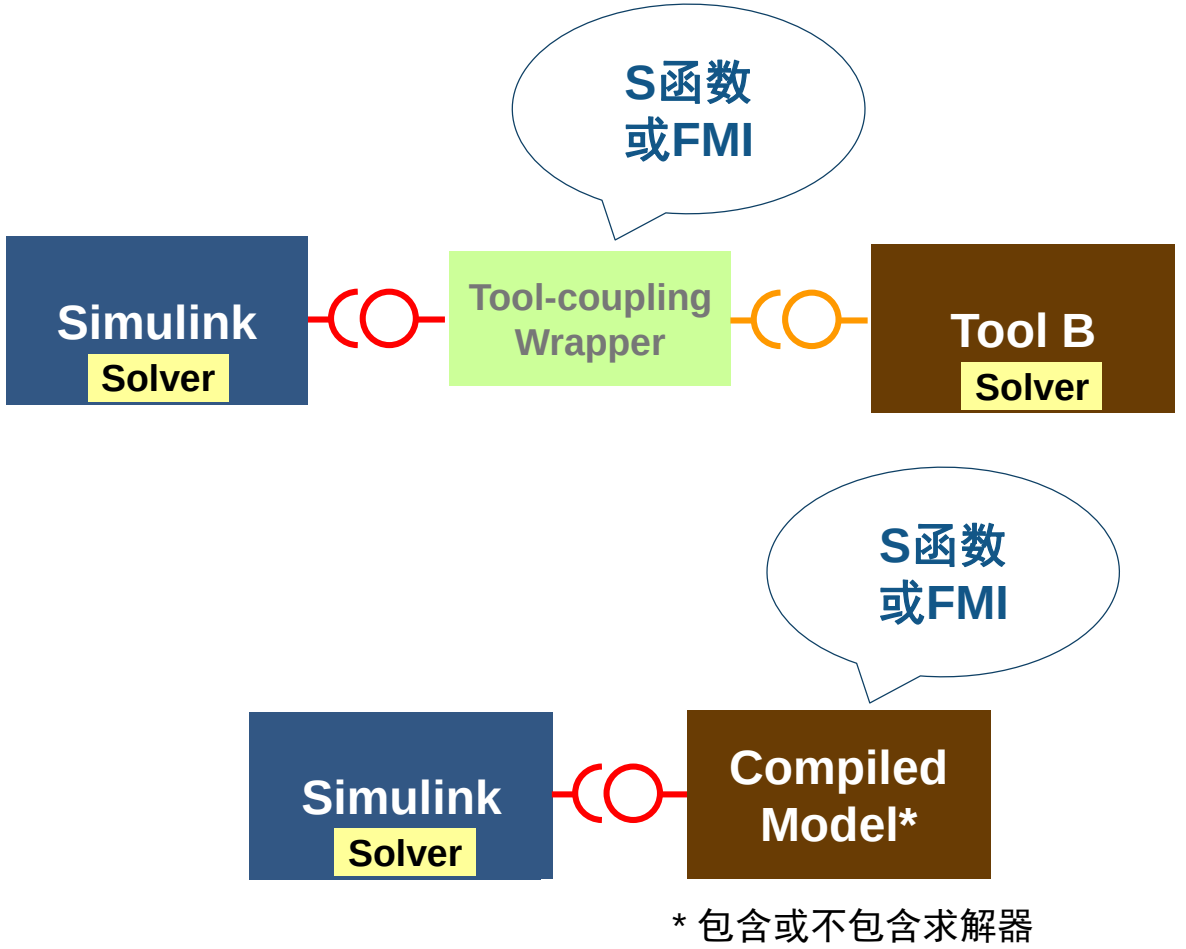




* 包含或不包含求解器

白盒集成和黑盒集成

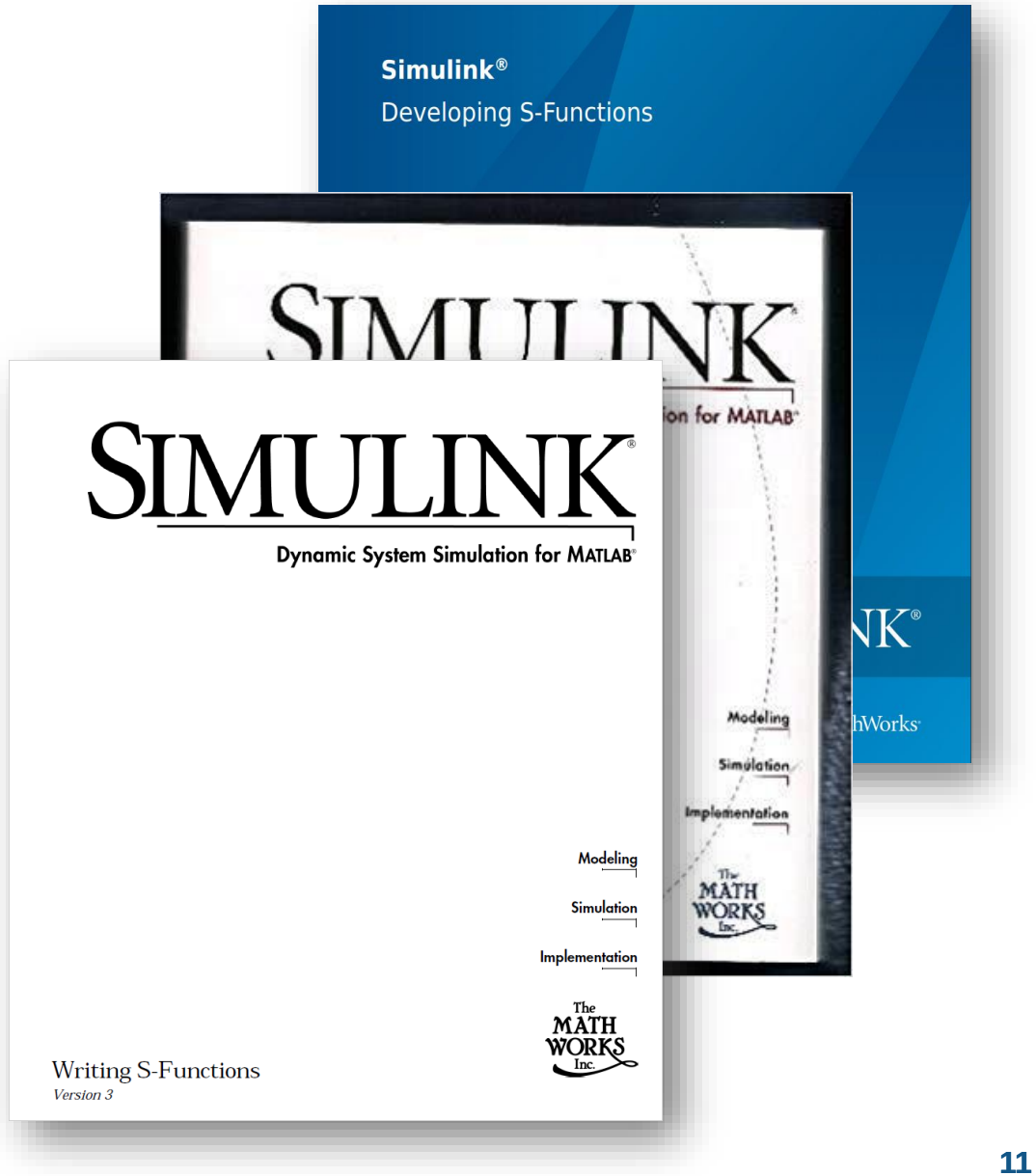
- 白盒集成 (工具交联)
  - 在仿真时，Simulink和外部工具均运行
- 黑盒集成 (编译模型)
  - 在仿真时，仅 Simulink运行
  - 第三方模型是Simulink中的组件
- 两种形式均使用标准接口
  - S函数
  - FMI (Functional Mockup Interface)



* 包含或不包含求解器

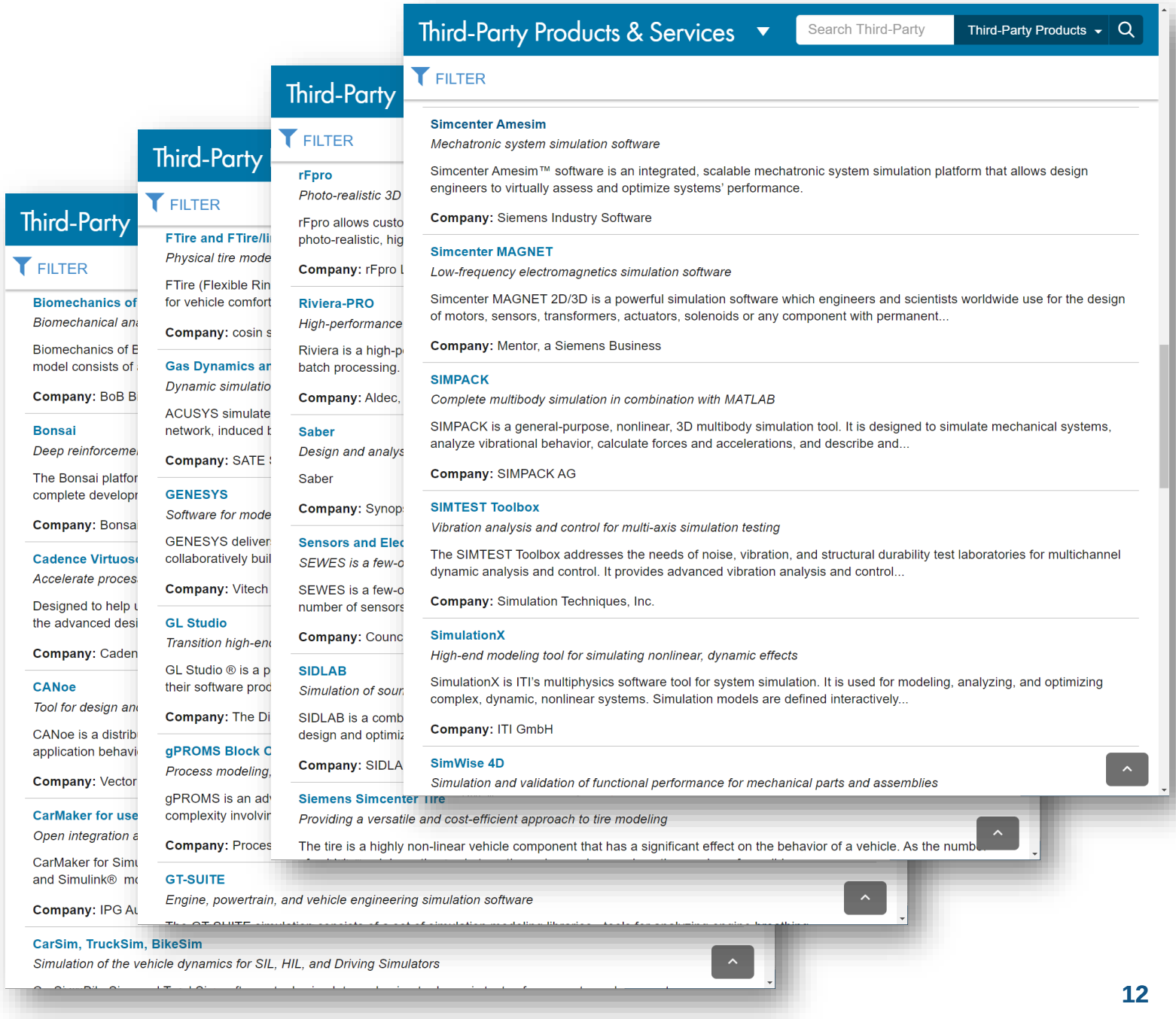
S函数接口

- 使用 MATLAB 和 C/C++语言自定义动态系统
- 支持Simulink所有特性
- 经过20+年行业验证
- Simulink与将外部工具联合的事实标准



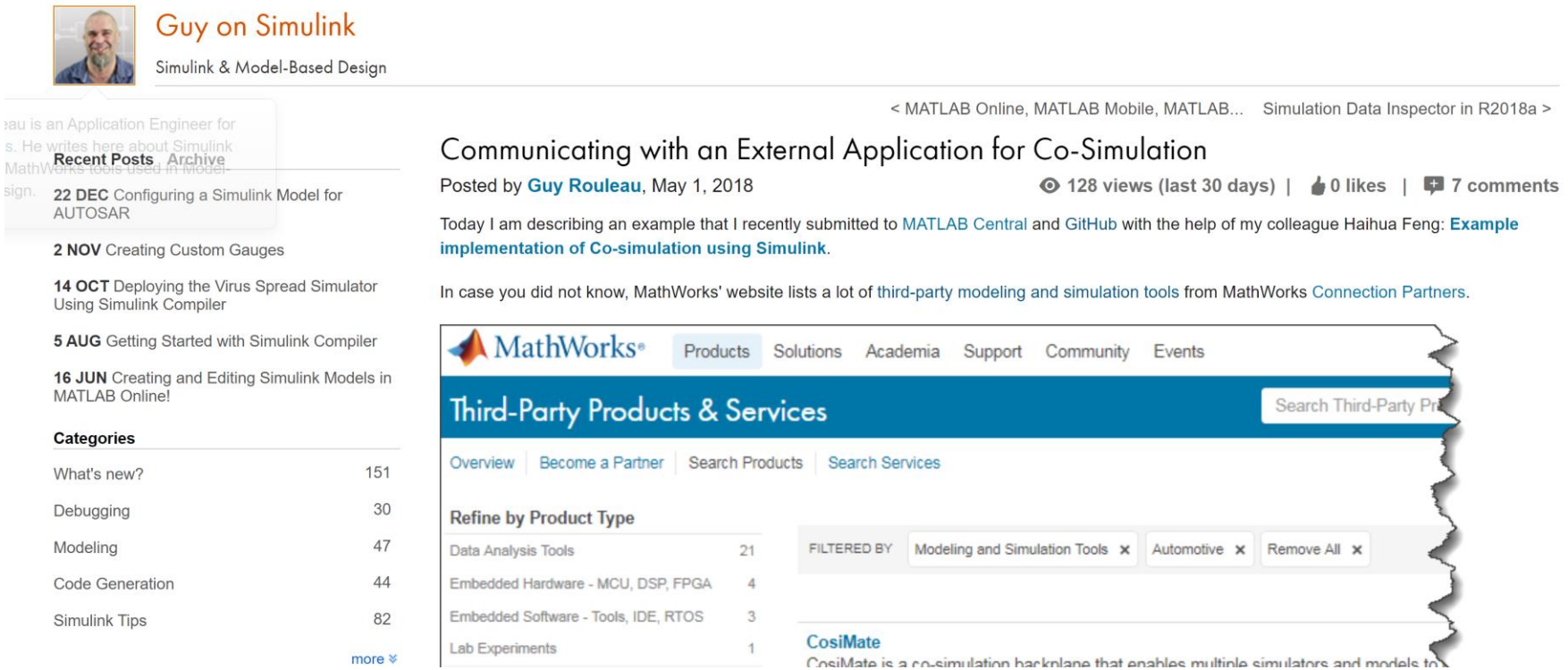
S函数接口

- 多达150+家MathWorks合作伙伴提供预置的S函数联合仿真接口
- 非合作伙伴工具也通过S函数提供联合仿真接口



更多关于S函数与外部应用的通信介绍

- 使用S函数用于多工具交互接口的模板
  - 链接到博客文章



FMI (Functional Mock-up Interface)

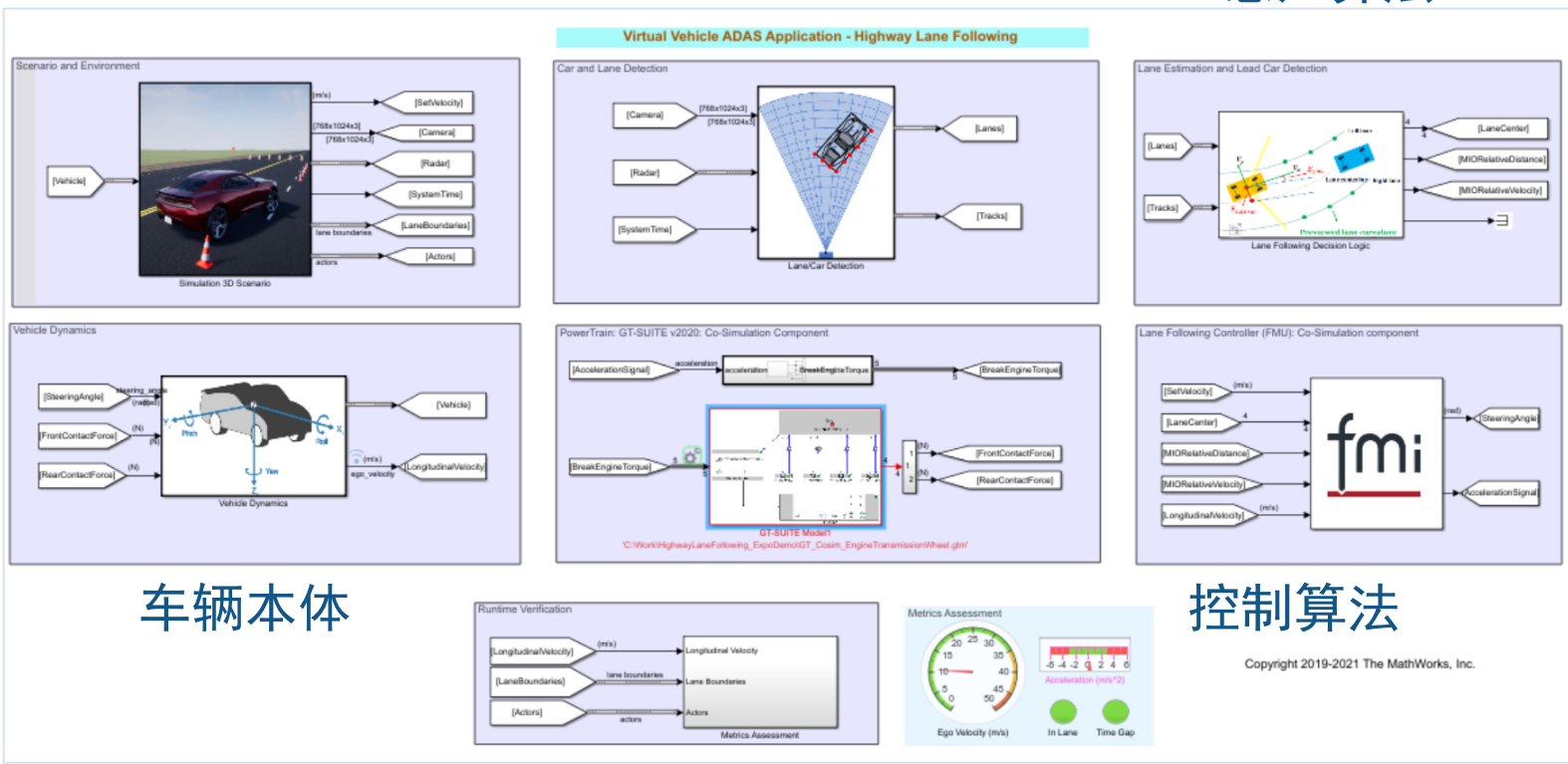
- FMI是与工具无关的系统动态模型格式
  - FMU是一个将模型以FMI格式导出打包的压缩文件
- 支持FMU导入导出的工具在持续增长
- Simulink支持FMI V1.0和V2.0版
- 支持联合仿真和模型交换形式导入FMU模型

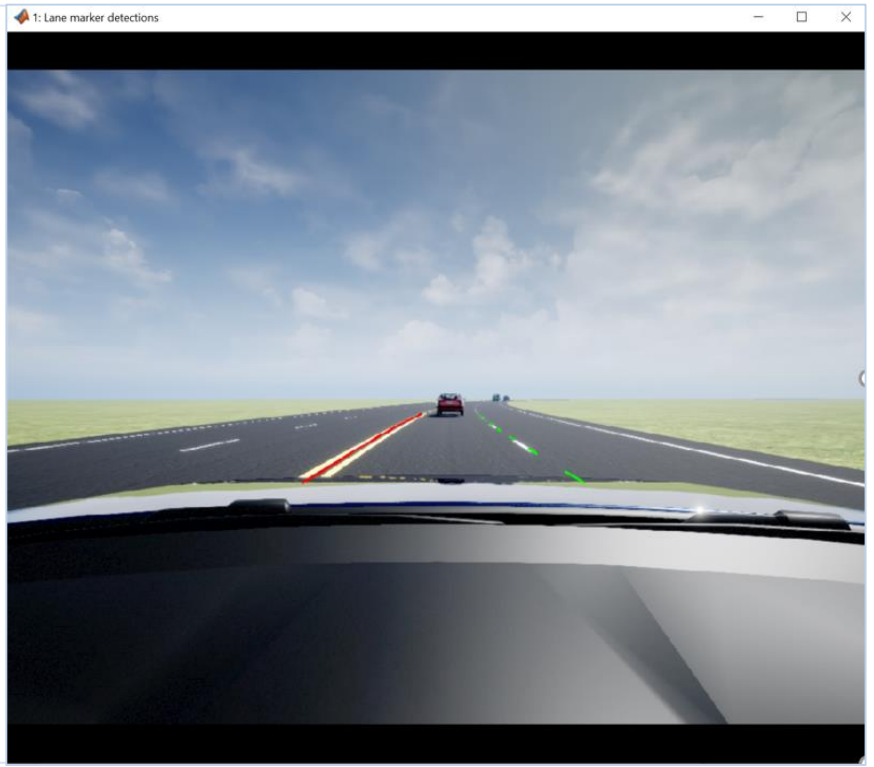
演示示例 – 虚拟车辆的ADAS应用

- 使用S函数和FMU将外部组件集成进来

环境模型

感知算法



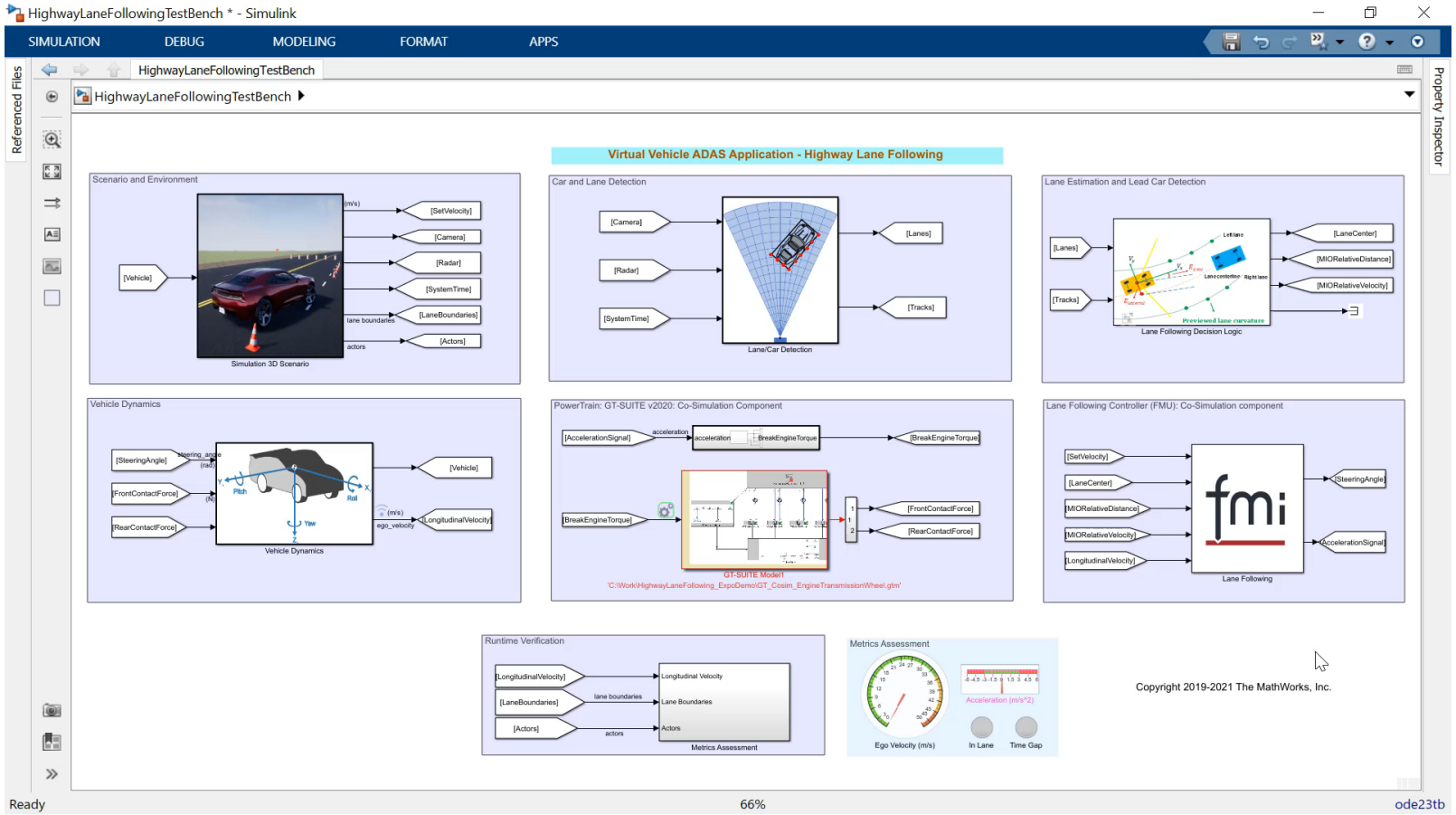


车辆本体

控制算法

演示示例 – 虚拟车辆的ADAS应用

- 使用S函数和FMU将外部组件集成进来



演讲提纲

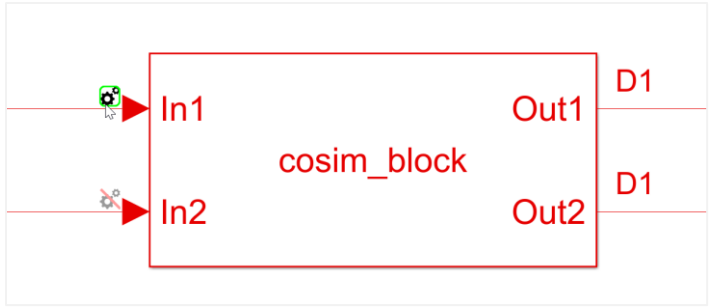
与外部仿真工具的接口

联合仿真的鲁棒性

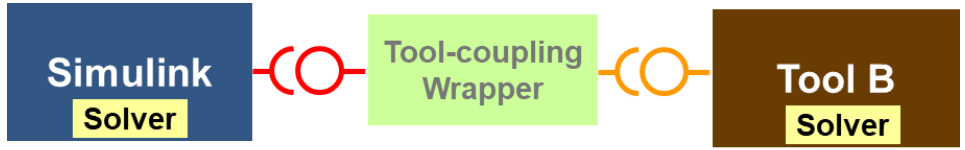
引入自定义C/C++代码

系统级仿真加速

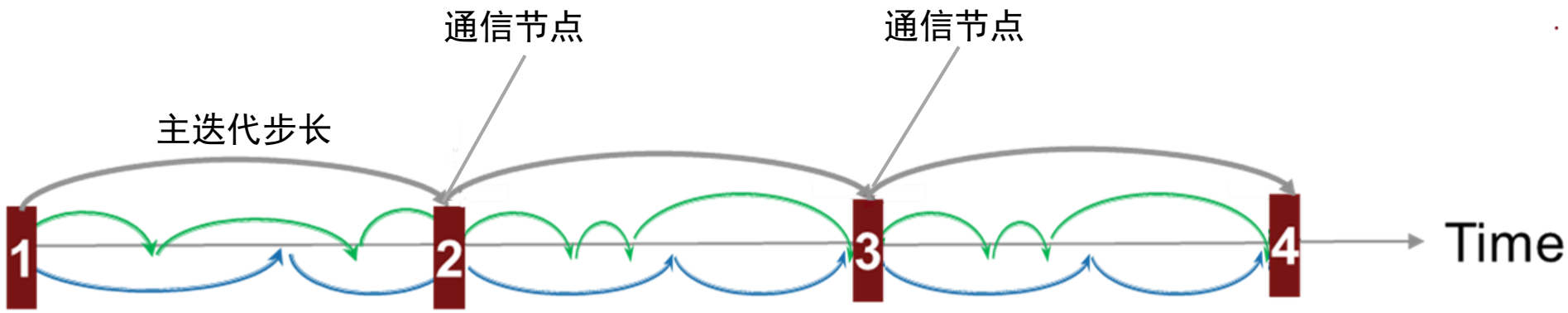
联合仿真的鲁棒性



联合仿真

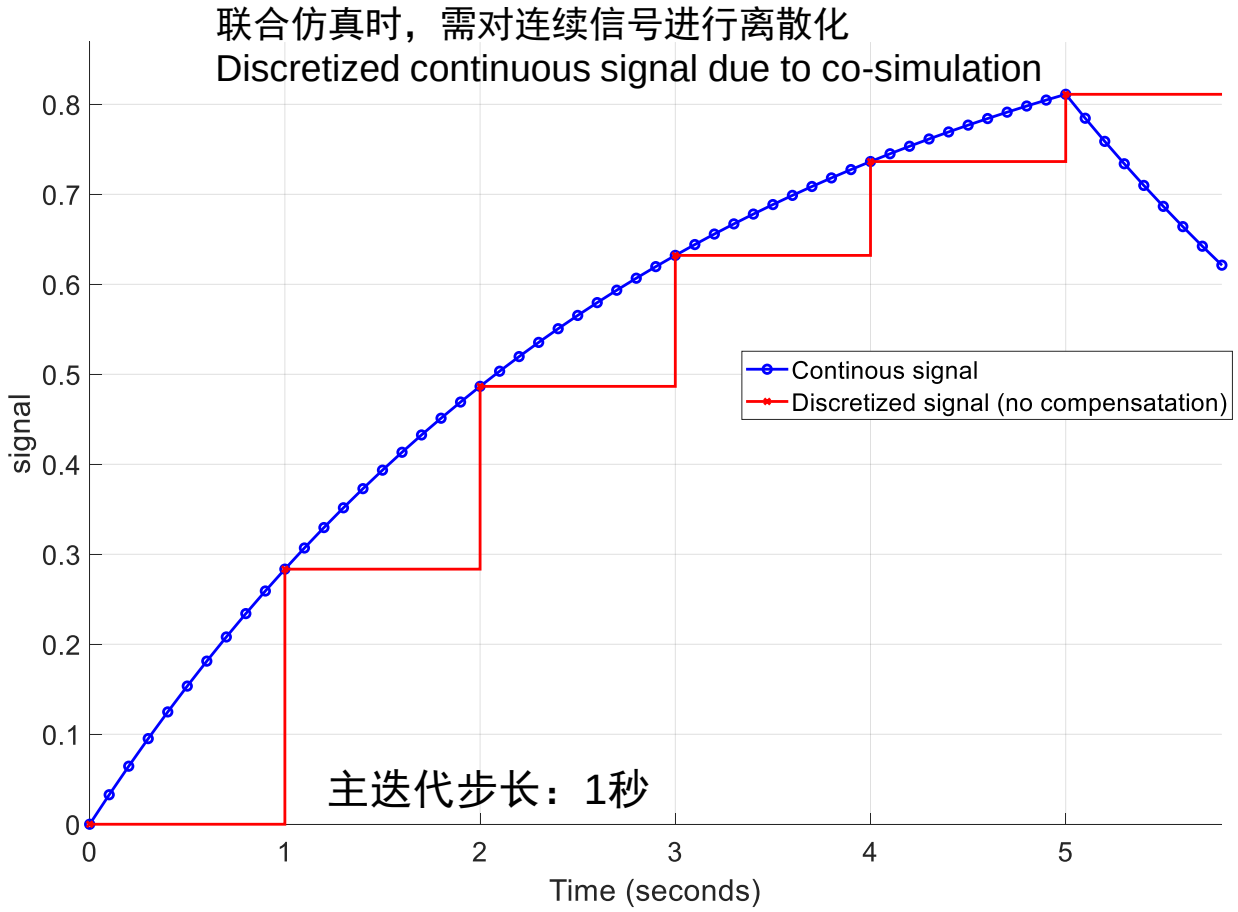


- 将外部工具模型导入的Simulink的常用方法
  - 联合仿真的各组件均有自己的求解器
  - 可以通过白盒或黑盒形式实现
- 在通信节点之间，联合仿真的组件可自由并发运行



联合仿真中的数值行为

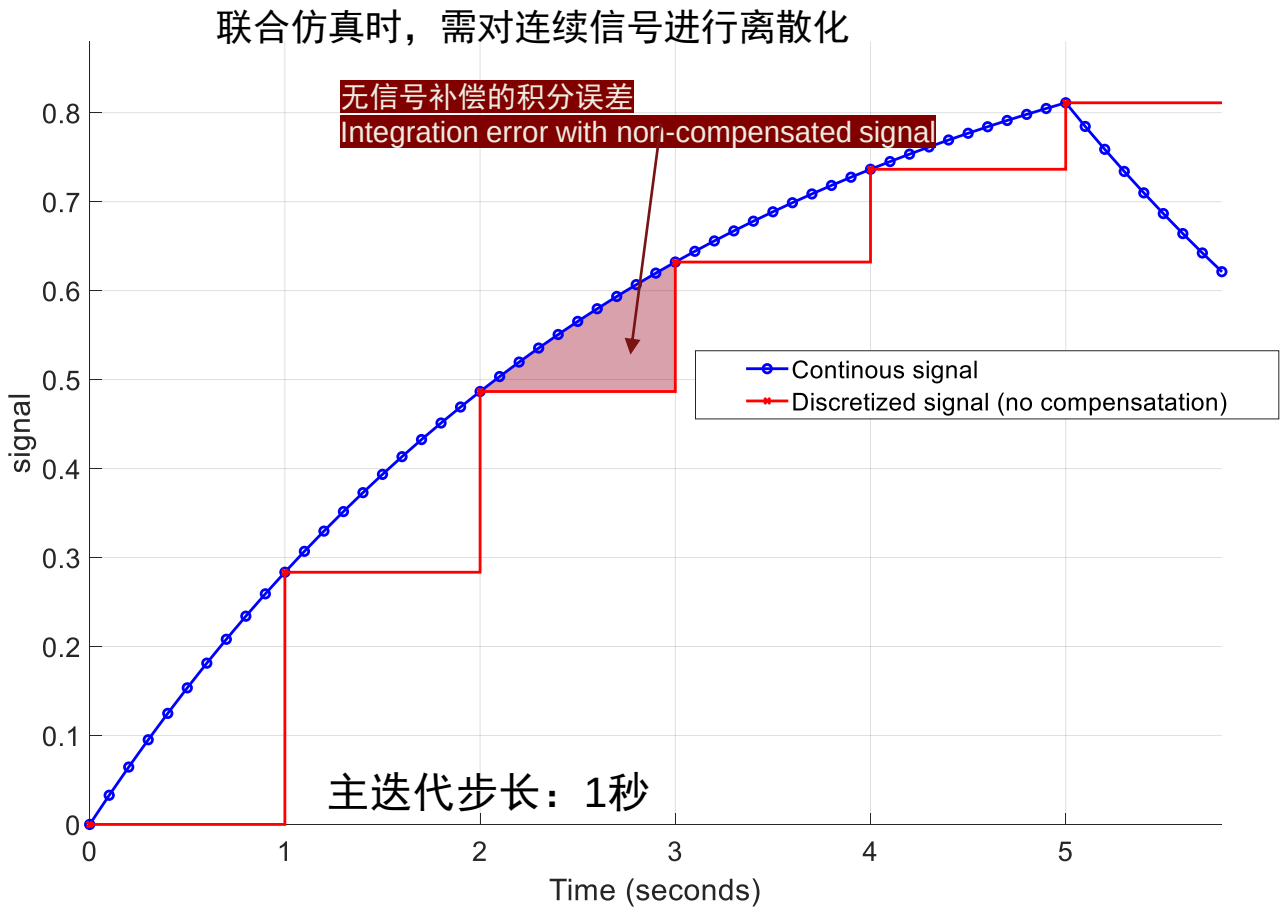
- 模型集成不仅仅是连接信号线
- 连续信号对接时，潜在错误可能大增
  - 离散化和延迟
  - 无补偿的信号，可能导致精度损失，甚至系统不稳定



The un-compensated signal (red line) deviates from the ideal, continuous signal (blue line) due to discretization
因为离散化的原因，无补偿的信号（红线）与理想的连续信号（蓝色）发生偏差

联合仿真中的数值行为

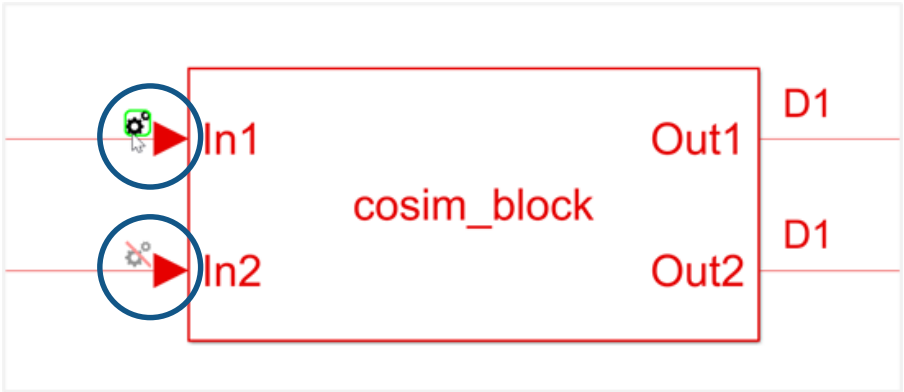
- 模型集成不仅仅是连接信号线
- 连续信号对接时，潜在错误可能大增
  - 离散化和延迟
  - 无补偿的信号，可能导致精度损失，甚至系统不稳定

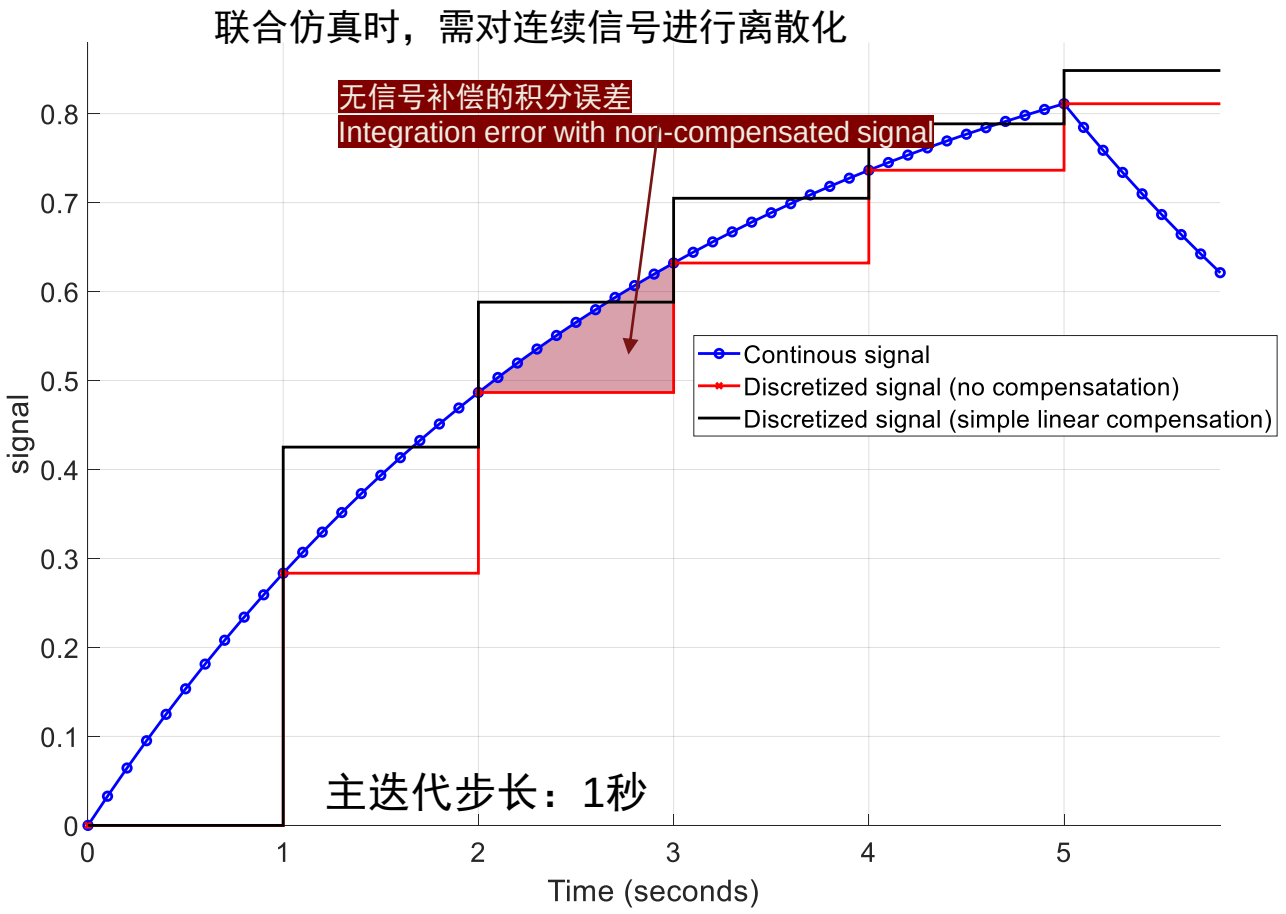


The un-compensated signal (red line) deviates from the ideal, continuous signal (blue line) due to discretization
因为离散化的原因，无补偿的信号（红线）与理想的连续信号（蓝色）发生偏差

联合仿真的鲁棒性

- 对离散化的连续信号，提供自动和手动的补偿机制
  - 提供线性或高阶外推补偿方法
- 相比无信号补偿的仿真，结果鲁棒性更好

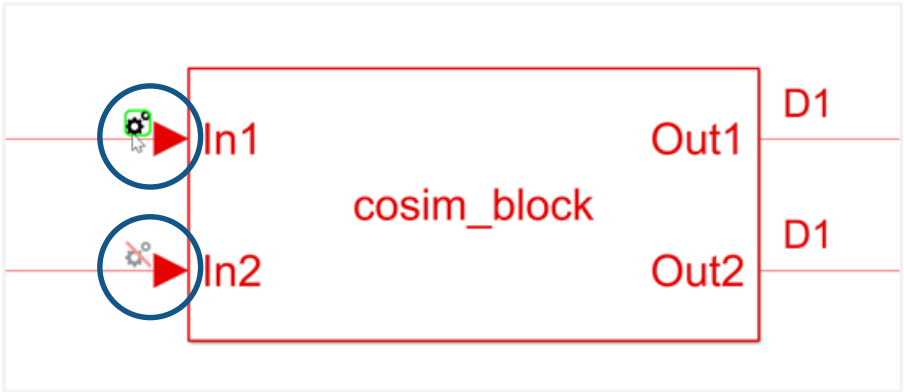


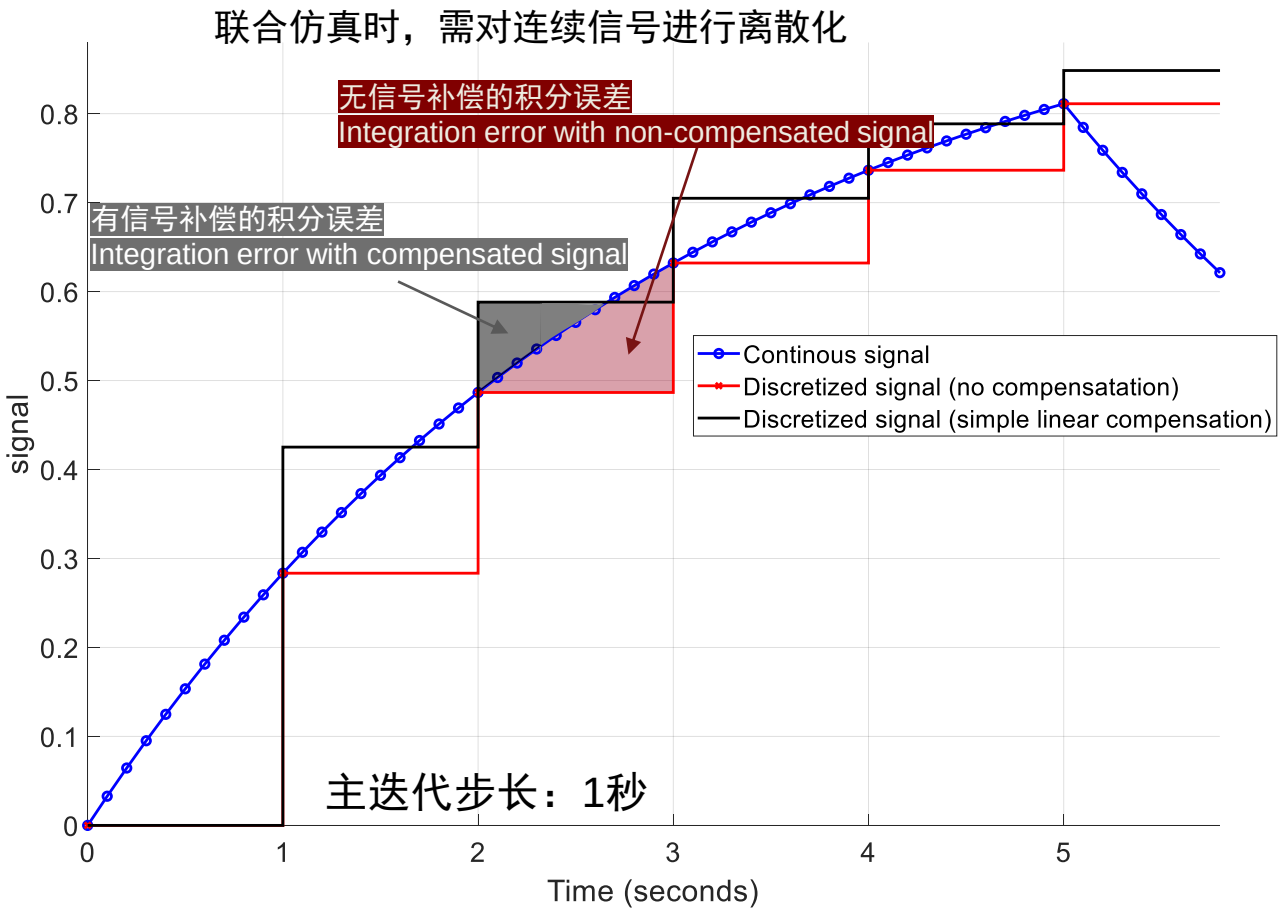


经过简单线性外推的补偿信号(黑线)比没有补偿的离散信号(红线)更接近理想的连续信号(蓝线)

联合仿真的鲁棒性

- 对离散化的连续信号，提供自动和手动的补偿机制
  - 提供线性或高阶外推补偿方法
- 相比无信号补偿的仿真，结果鲁棒性更好





经过简单线性外推的补偿信号(黑线)比没有补偿的离散信号(红线)更接近理想的连续信号(蓝线)

演讲提纲

与外部仿真工具的接口

联合仿真的鲁棒性

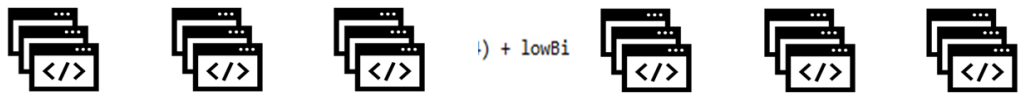
引入自定义C/C++代码

系统级仿真加速

引入自定义C/C++代码

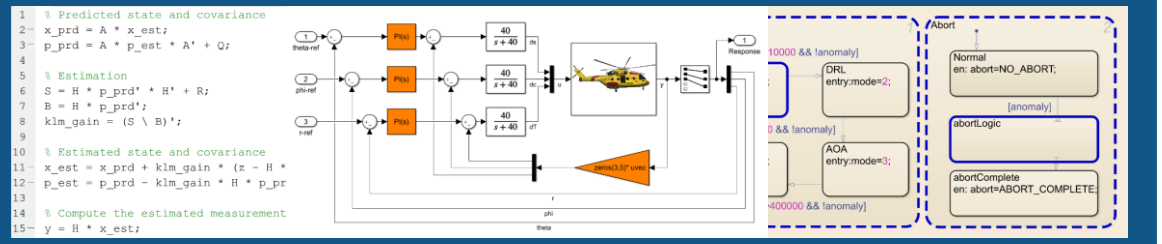
```
template<class InputString, class OutputString>
bool unhexlify(const InputString& input, OutputString& output) {
  if (input.size() % 2 != 0) {
    return false;
  }
  output.resize(input.size() / 2);
  int j = 0;
  auto unhex = [](char c) -> int {
    return c >= '0' && c <= '9' ? c - '0' :
           c >= 'A' && c <= 'F' ? c - 'A' + 10 :
           c >= 'a' && c <= 'f' ? c - 'a' + 10 :
           -1;
  };

  for (size_t i = 0; i < input.size(); i += 2) {
```

Internal Libraries **Vendor Libraries**



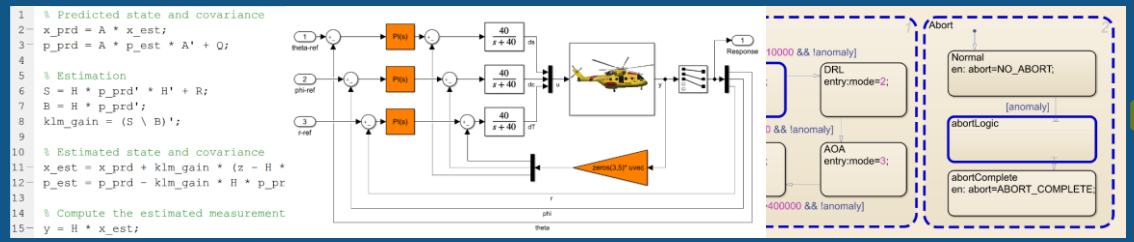
引入自定义C/C++代码

模型

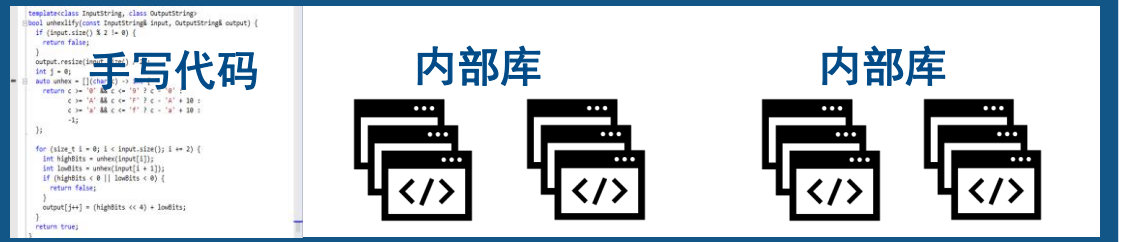


引入自定义C/C++代码

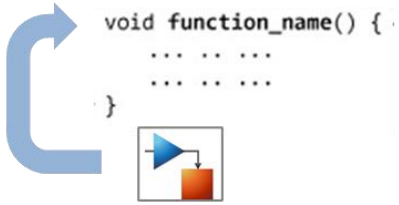
模型

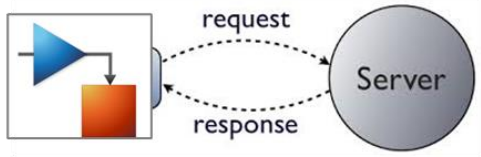


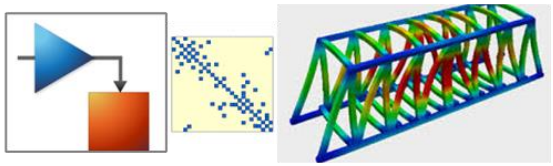
C/C++ 库

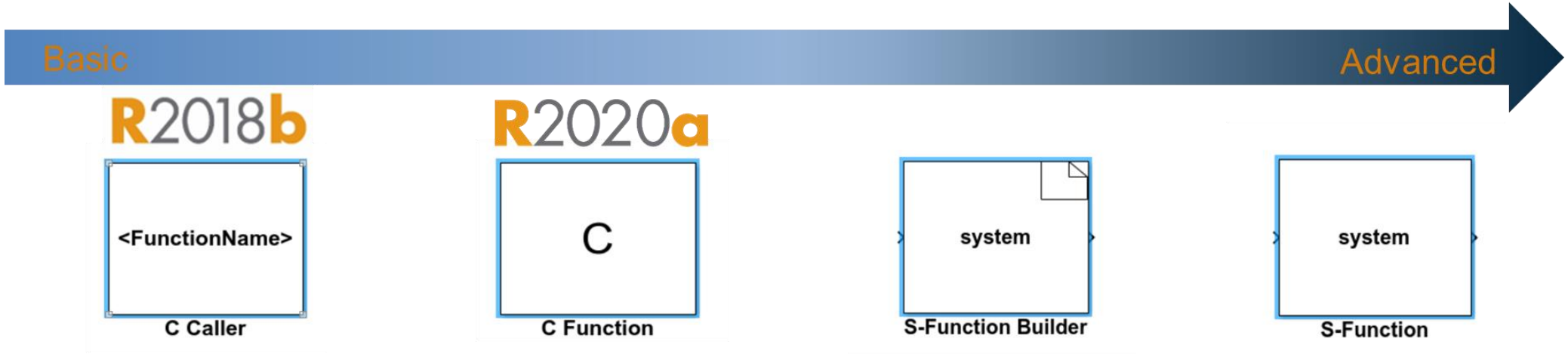


引入自定义C/C++代码





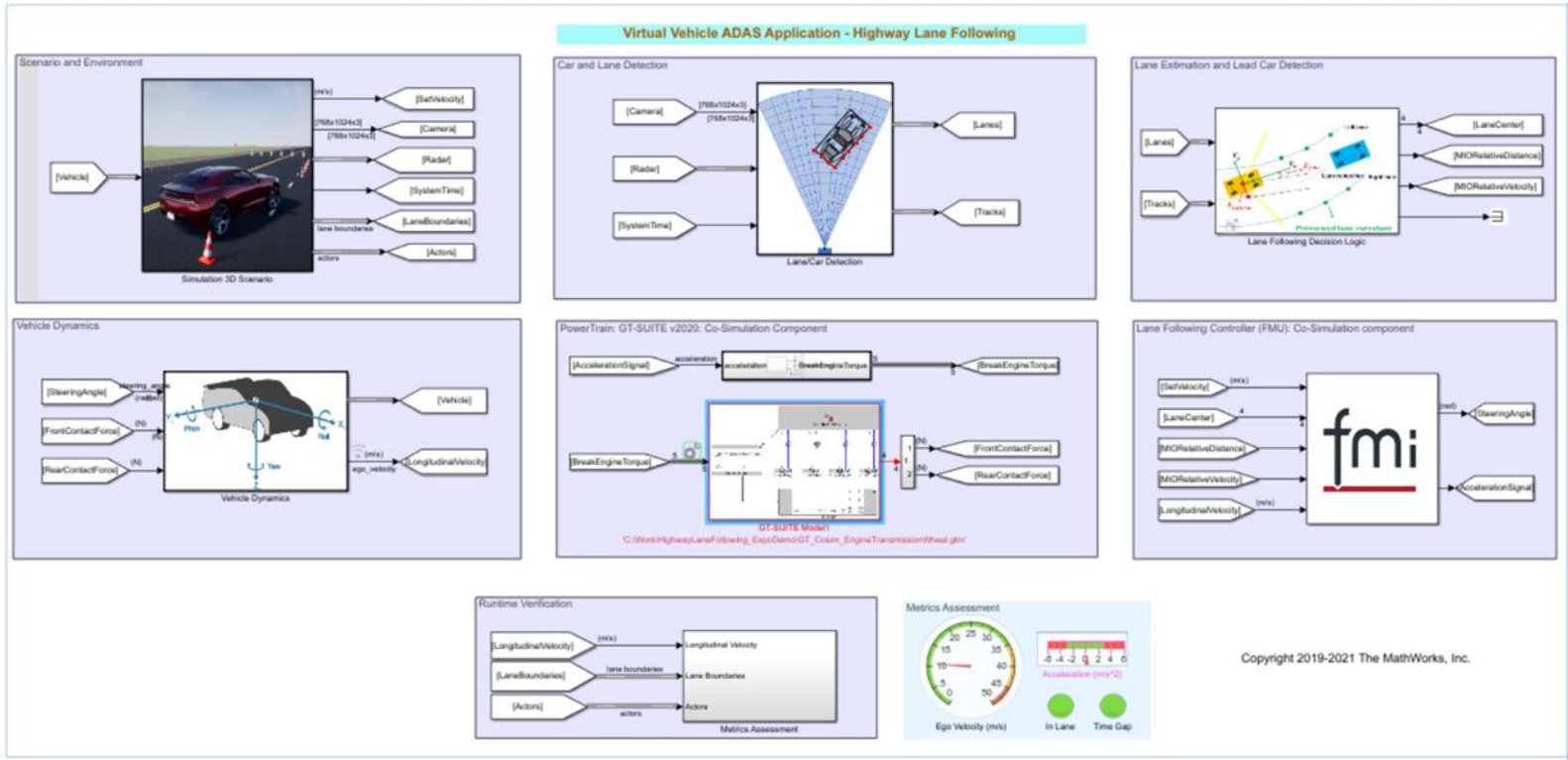


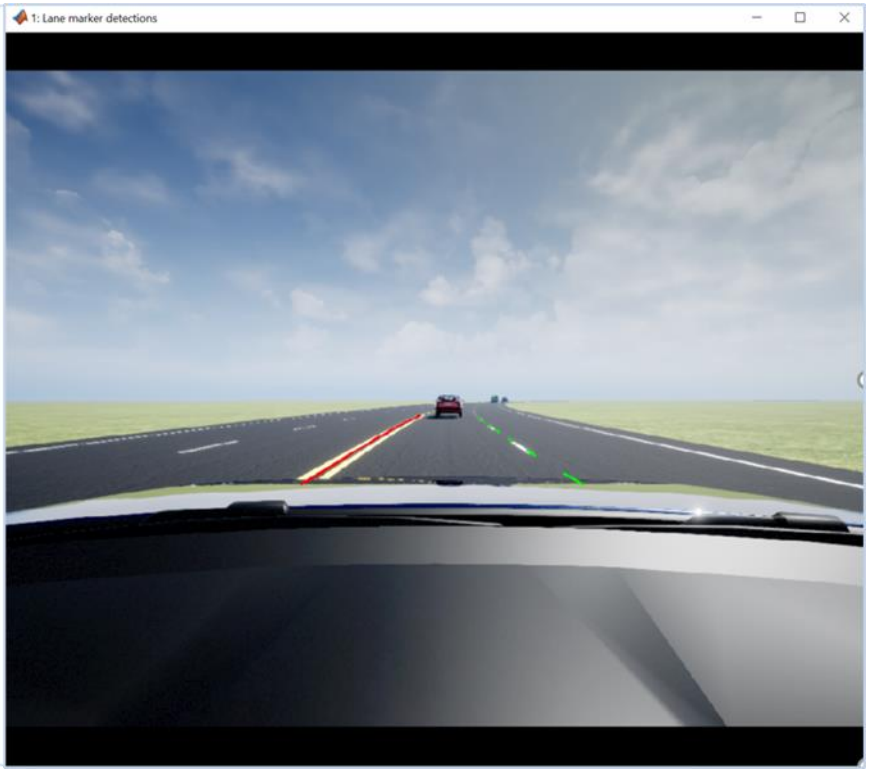


- 集成自定义代码的方法多样
  - 简单调用函数
  - 以Simulink模型库重用代码
  - 编写包含离散状态的算法
  - 创建动态系统

演示示例 – 虚拟车辆的ADAS应用

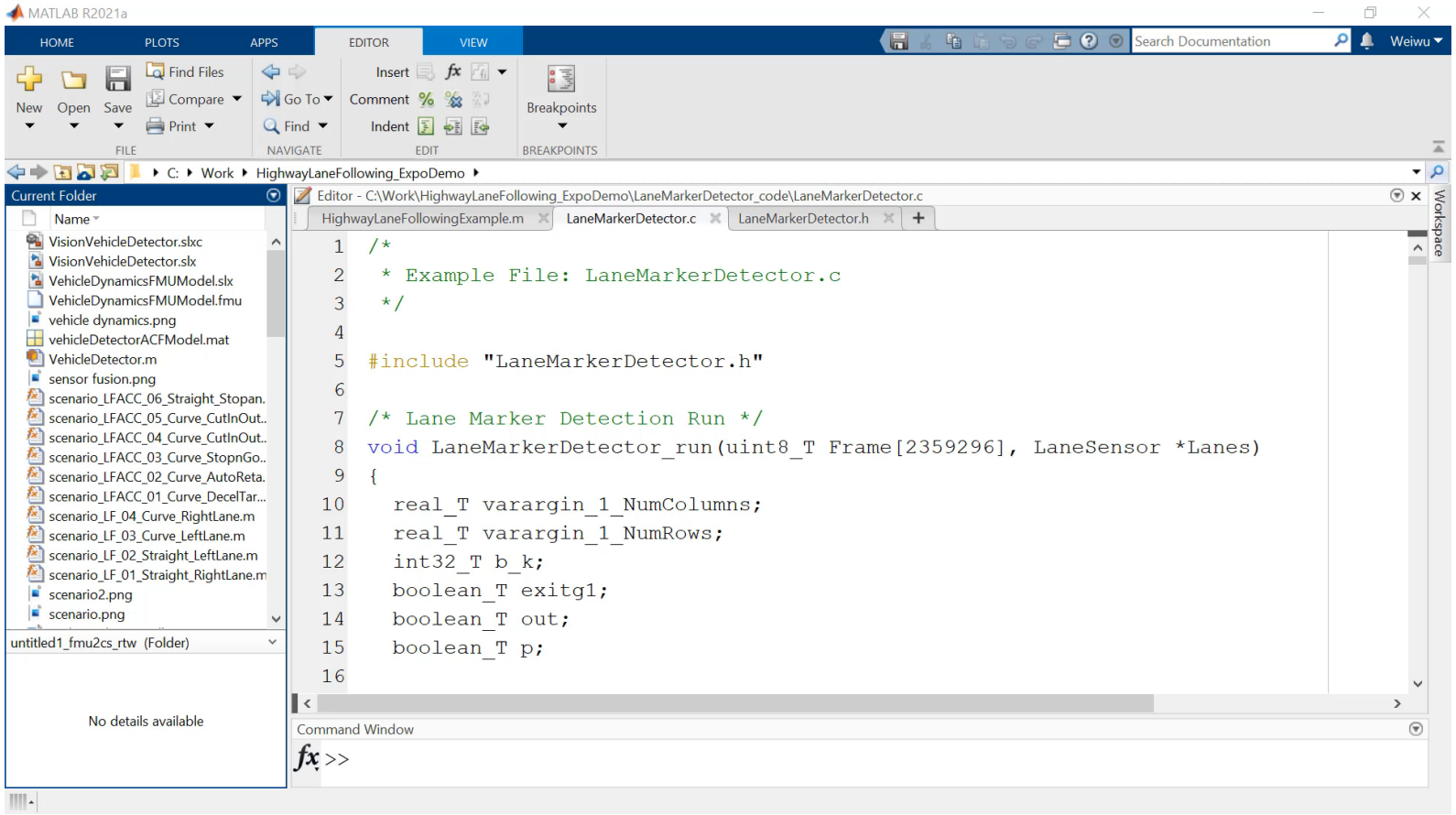
- 集成自定义的车道线检测 C 代码





演示示例 – 虚拟车辆的ADAS应用

- 集成自定义的车道线检测 C 代码



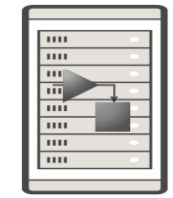
演讲提纲

与外部仿真工具的接口

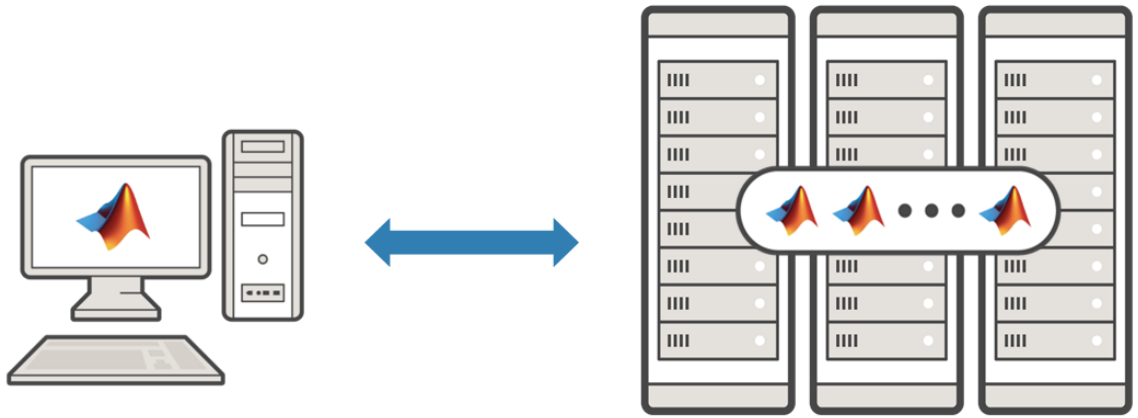
联合仿真的鲁棒性

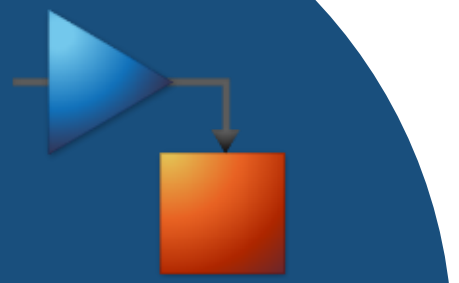
引入自定义C/C++代码

系统级仿真加速



系统级仿真加速





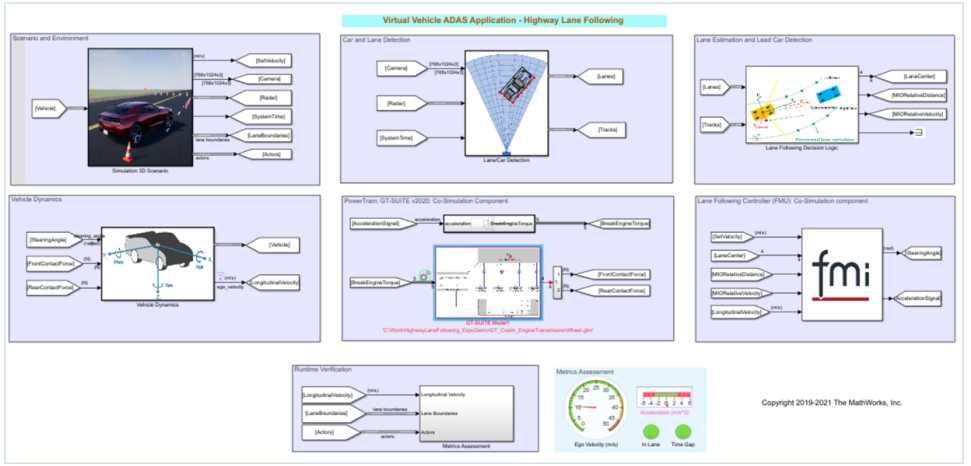
大规模系统仿真

- 系统级仿真可能因多种设计组合，需要大量仿真迭代
- 复杂系统仿真花费时间长
- 可扩展性是集成平台的必备能力，能加速仿真，加速对系统理解

整车仿真

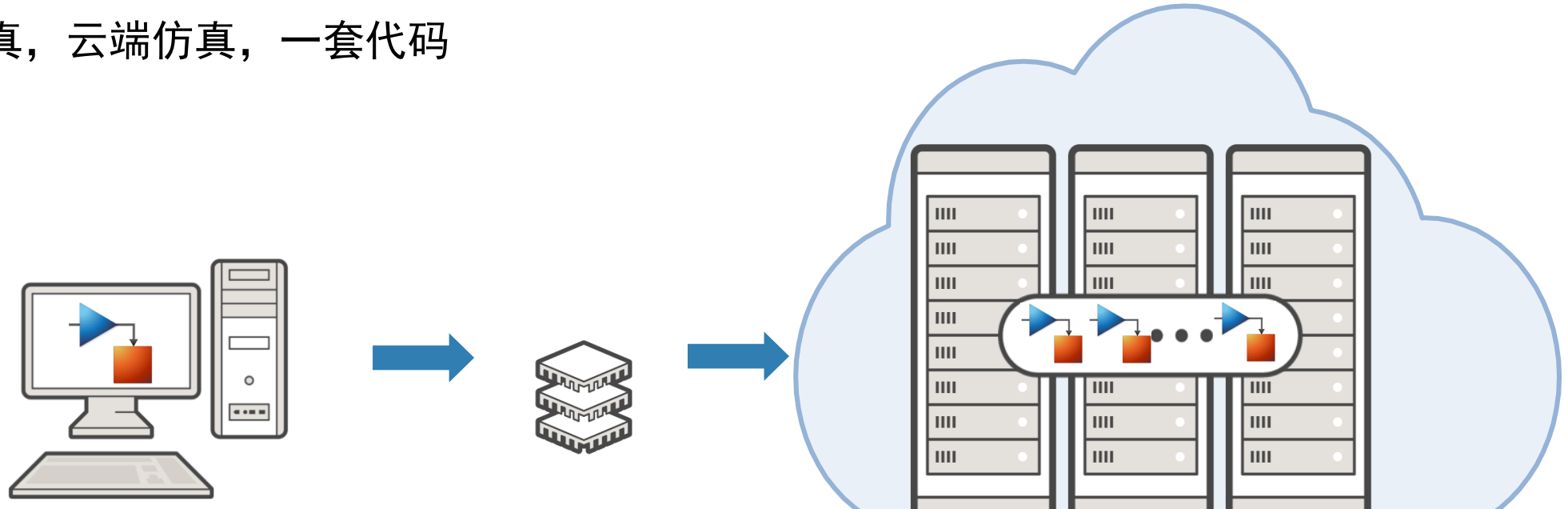
10个驾驶循环
10种天气条件
10种车辆载荷
10种传动比
10个轮胎尺寸

-> 100,000 仿真



大规模系统仿真

- 桌面仿真，云端仿真，一套代码

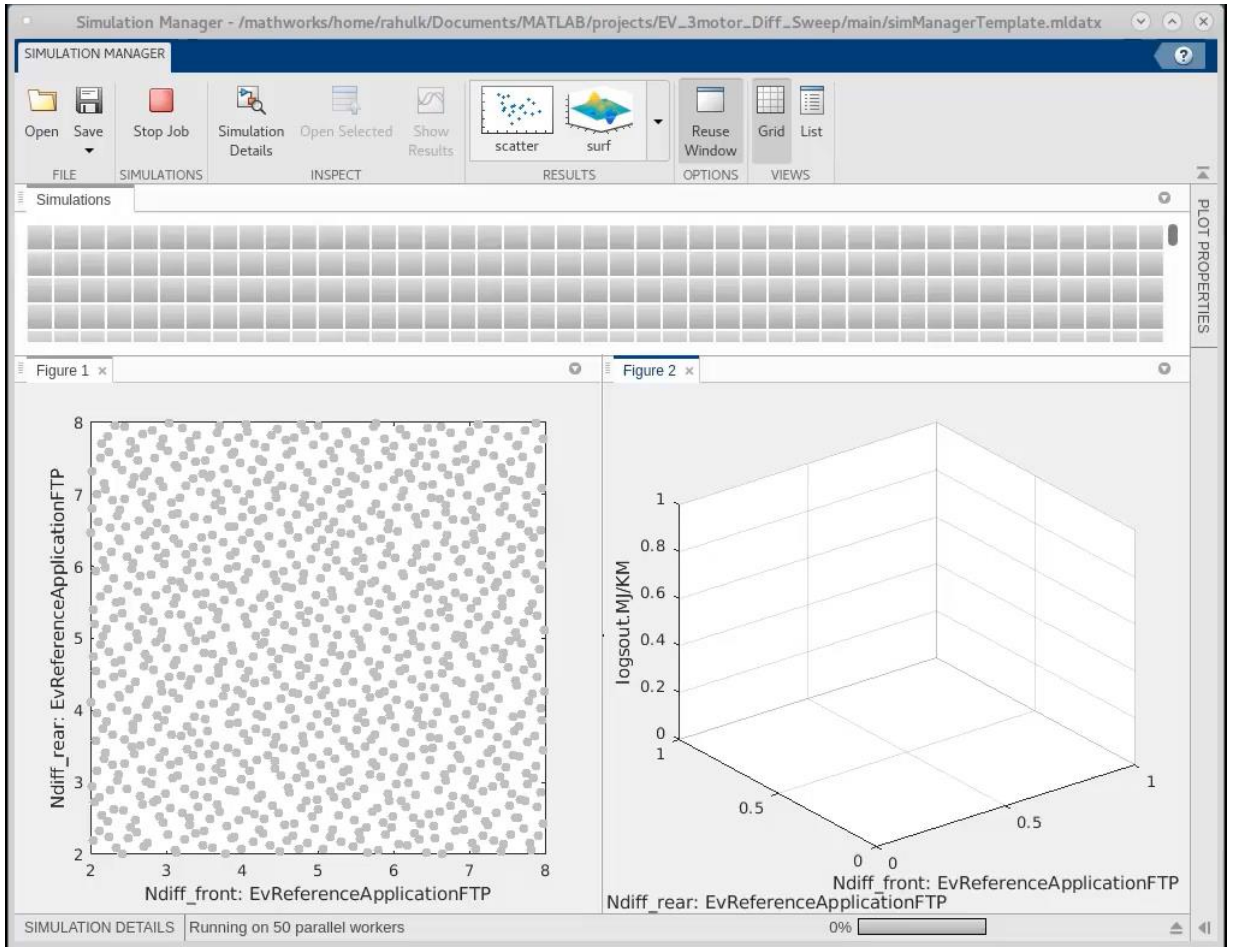


桌面　　多核　　集群

```
for i = 10000:-1:1
    in(i) = Simulink.SimulationInput('my_model');
    in(i) = in(i).setVariable('my_var', i);
end
out = parsim(in);
```

大规模系统仿真

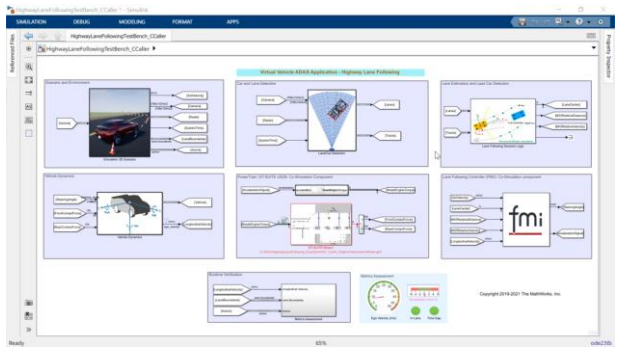
- 仿真管理和可视化

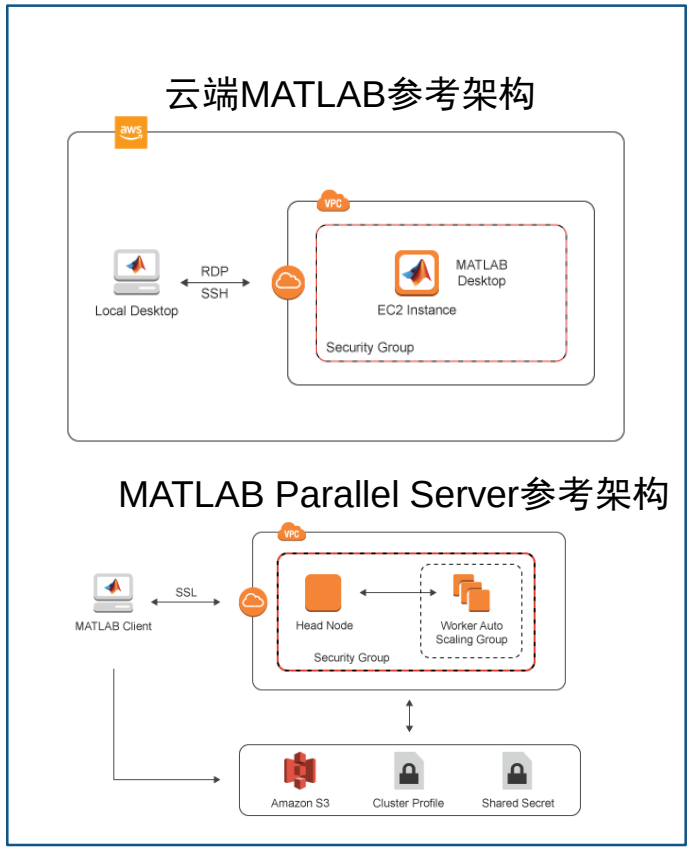


大规模系统仿真

- 通过参考架构并利用预构建的云配置，将仿真迁移到云端

系统仿真





Microsoft Azure

amazon web services

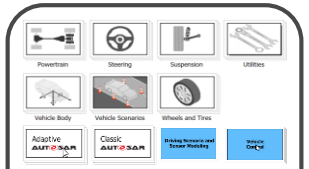
使用云端MATLAB/Simulink

要点总结

Simulink 是开放的集成仿真平台，用于复杂多样的多域物理仿真。

- 通过标准接口与第三方工具和模型集成
- 使用自动信号补偿提高联合仿真的数值鲁棒性
- 快速简单的引入自定义C/C++代码
- 使用并行计算加速系统级的仿真

DDS



Block libraries

MATLAB

C++

fmi

OpenCV

python

Simulink

ROS

UNREAL ENGINE

NVIDIA CUDA

MATLAB EXPO 2021

感谢您的聆听