

MATLAB EXPO 2021

将外部仿真组件集成到Simulink平台中

董淑成, 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。



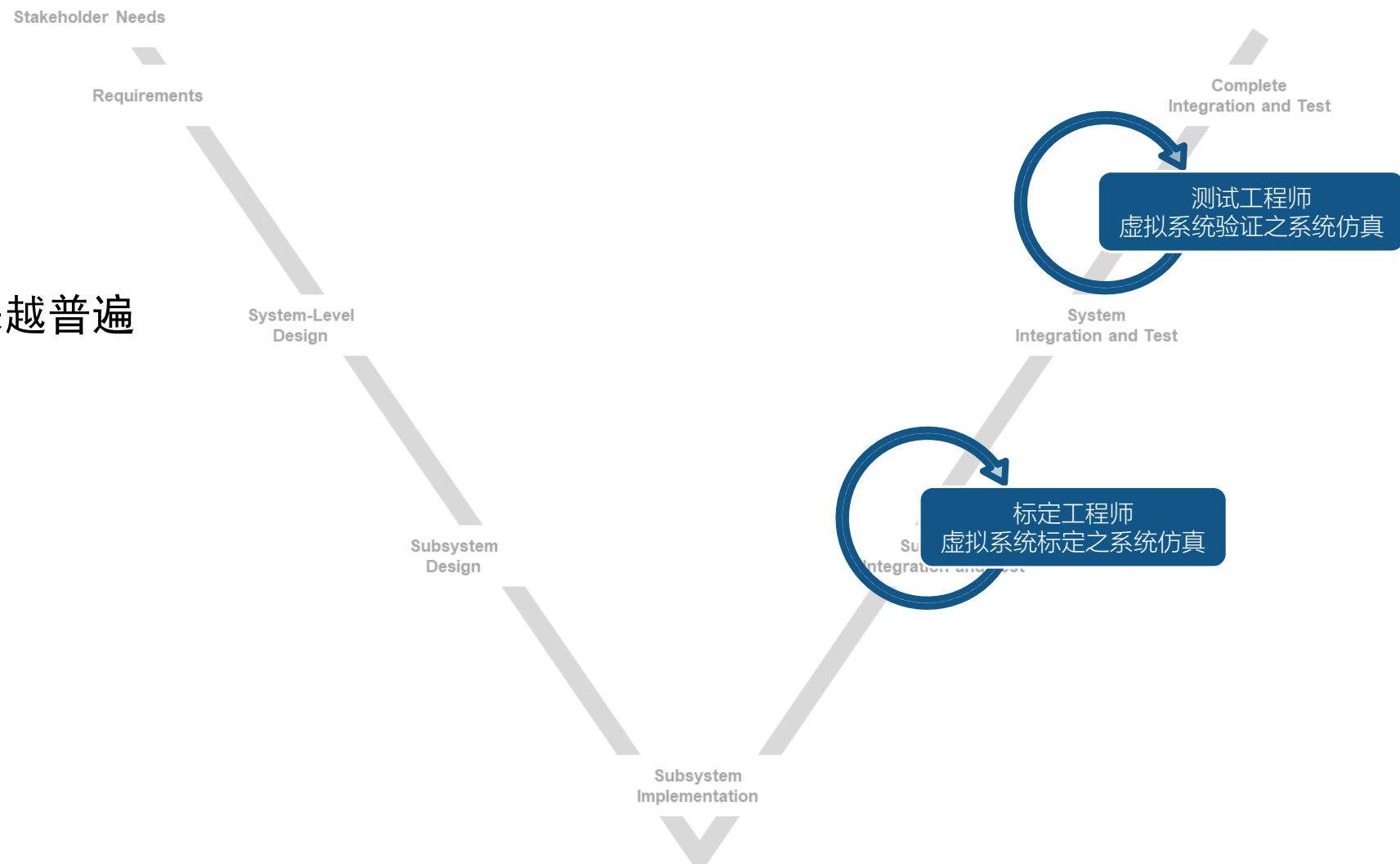
联合仿真动机所在

产品开发中，系统级仿真越来越普遍



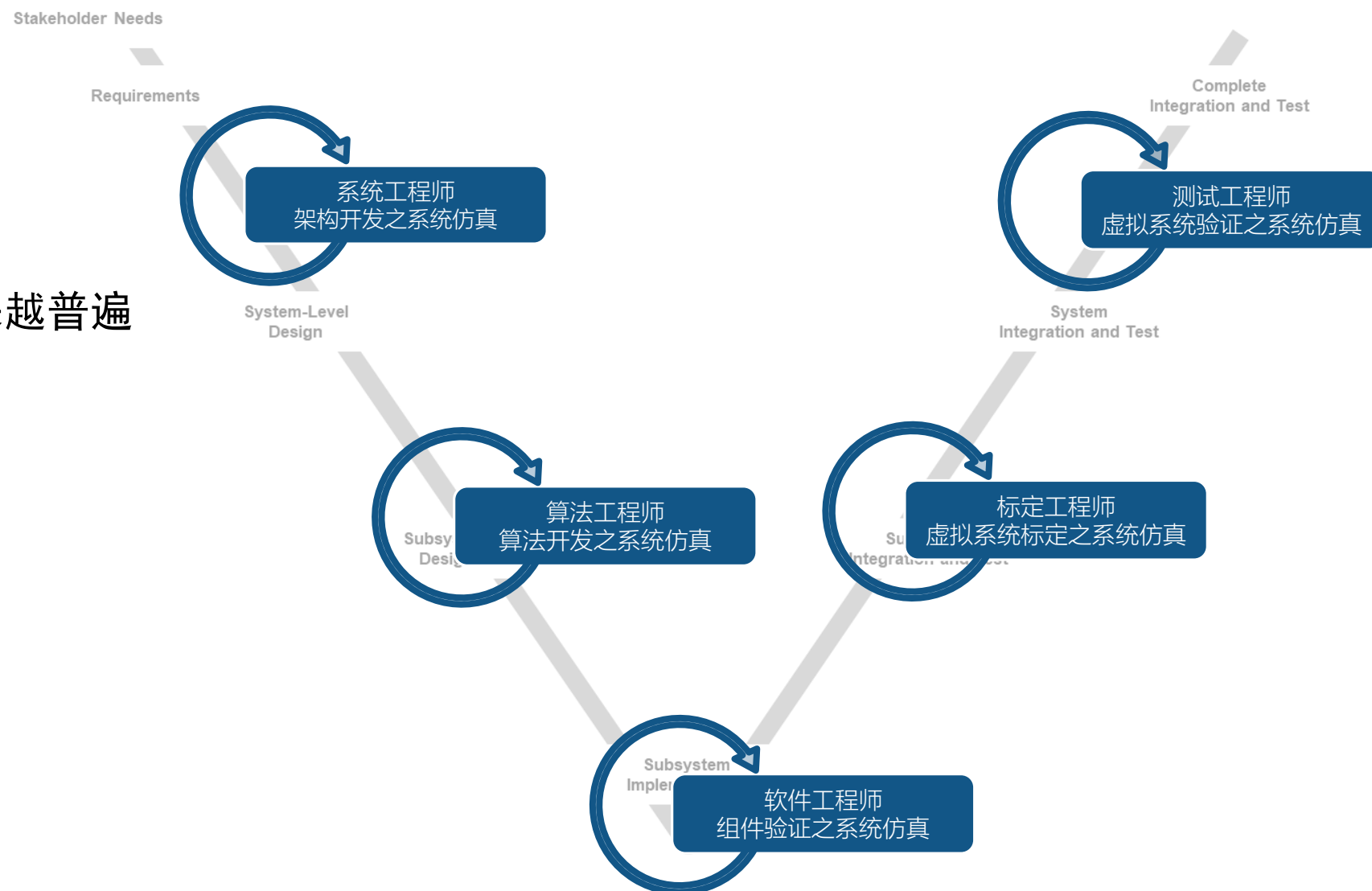
联合仿真动机所在

产品开发中，系统级仿真越来越普遍



联合仿真动机所在

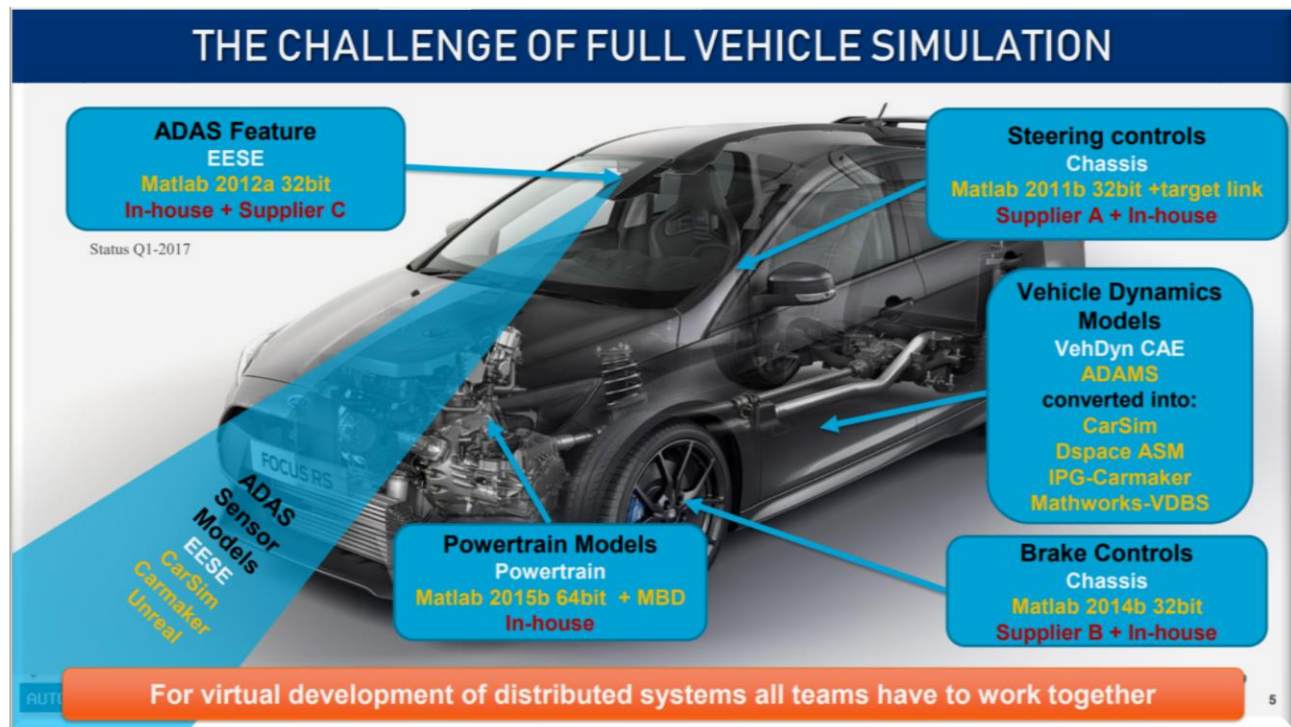
产品开发中，系统级仿真越来越普遍



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

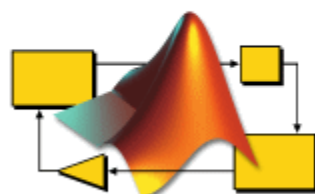


系统级仿真加速



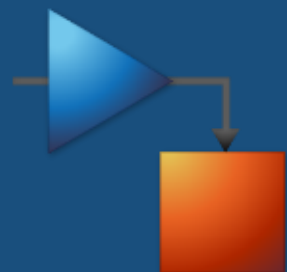
与外部仿真工具的接口

SIMULINK[®]
Enabled



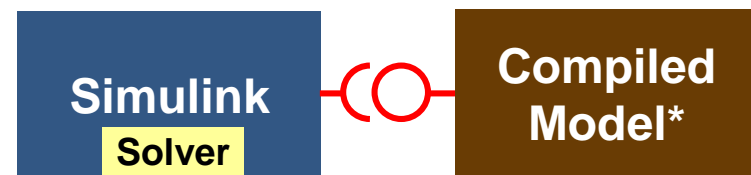
MathWorks Partner

fmi: Functional
Mock-Up
Interface



白盒集成和黑盒集成

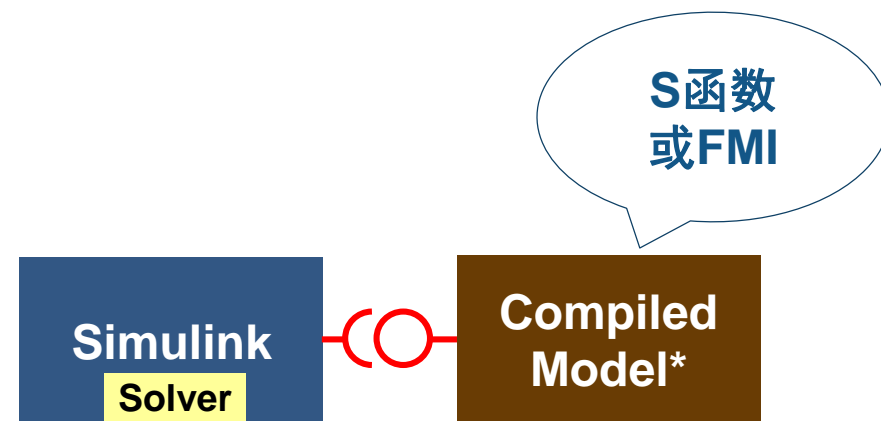
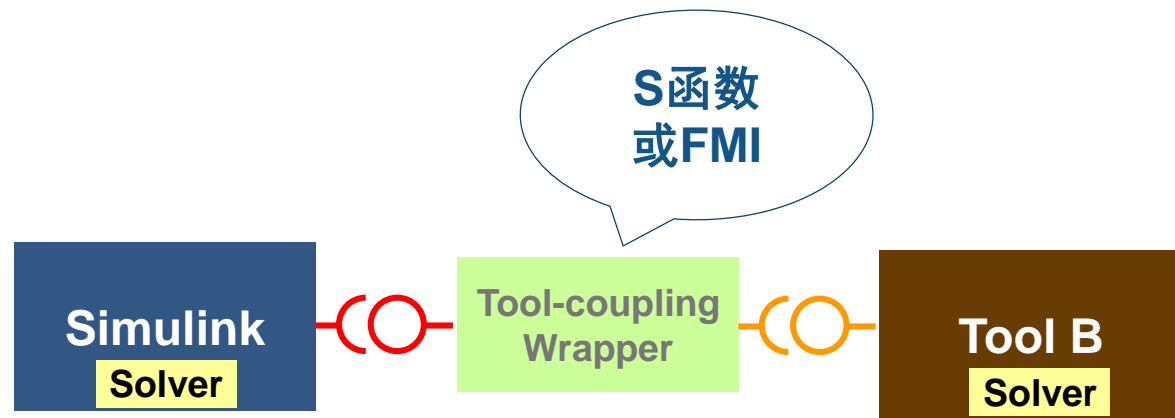
- 白盒集成 (工具交联)
 - 在仿真时, Simulink和外部工具均运行
- 黑盒集成 (编译模型)
 - 在仿真时, 仅 Simulink运行
 - 第三方模型是Simulink中的组件



* 包含或不包含求解器

白盒集成和黑盒集成

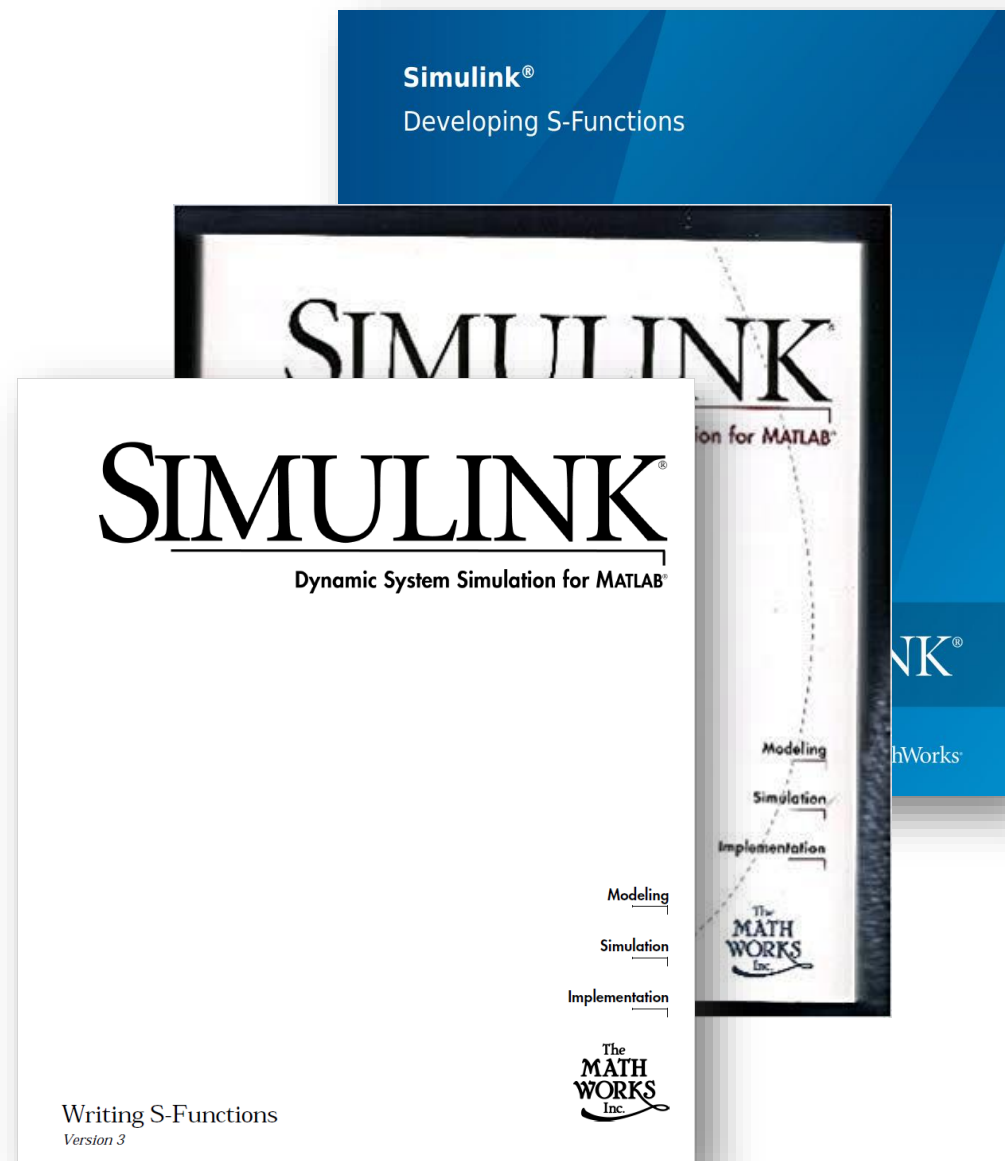
- 白盒集成 (工具交联)
 - 在仿真时, Simulink和外部工具均运行
- 黑盒集成 (编译模型)
 - 在仿真时, 仅 Simulink运行
 - 第三方模型是Simulink中的组件
- 两种形式均使用标准接口
 - S函数
 - FMI (Functional Mockup Interface)



* 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

The screenshot displays the 'Third-Party Products & Services' page on the MATLAB website. It features a search bar at the top right and a 'FILTER' button. The main content is a list of various simulation software products, each with a brief description and the name of the company that provides it. The products listed include:

- Biomechanics of E...** (Biomechanical analysis model) - Company: BoB B...
- Bonsai** (Deep reinforcement learning) - Company: Bonsai...
- Cadence Virtuoso** (Accelerate process design) - Company: Caden...
- CANoe** (Tool for design and analysis) - Company: Vector...
- CarMaker for use...** (Open integration and simulation) - Company: IPG A...
- CarSim, TruckSim, BikeSim** (Simulation of vehicle dynamics) - Company: IPG A...
- FTire and FTire/li...** (Physical tire model) - Company: cosin s...
- Gas Dynamics an...** (Dynamic simulation) - Company: Vitech...
- GL Studio** (Transition high-end modeling) - Company: Coun...
- gPROMS Block C...** (Process modeling) - Company: Proces...
- GT-SUITE** (Engine, powertrain, and vehicle engineering simulation software) - Company: IPG A...
- rFpro** (Photo-realistic 3D simulation) - Company: rFpro L...
- Riviera-PRO** (High-performance batch processing) - Company: Aldec...
- Saber** (Design and analysis) - Company: Synop...
- Sensors and Elec...** (SEWES is a few-o number of sensors) - Company: Coun...
- SIDLAB** (Simulation of sour...) - Company: SIDLA...
- Siemens Simcenter TIRE** (Providing a versatile and cost-efficient approach to tire modeling) - Company: Proces...
- Simcenter Amesim** (Mechatronic system simulation software) - Company: Siemens Industry Software
- Simcenter MAGNET** (Low-frequency electromagnetics simulation software) - Company: Mentor, a Siemens Business
- SIMPACK** (Complete multibody simulation in combination with MATLAB) - Company: SIMPACK AG
- SIMTEST Toolbox** (Vibration analysis and control for multi-axis simulation testing) - Company: Simulation Techniques, Inc.
- SimulationX** (High-end modeling tool for simulating nonlinear, dynamic effects) - Company: ITI GmbH
- SimWise 4D** (Simulation and validation of functional performance for mechanical parts and assemblies) - Company: ITI GmbH

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](#)



Guy on Simulink
Simulink & Model-Based Design

... is an Application Engineer for ... s. He writes here about Simulink MathWorks tools used in Model... sign.

Recent Posts [Archive](#)

- 22 DEC** Configuring a Simulink Model for AUTOSAR
- 2 NOV** Creating Custom Gauges
- 14 OCT** Deploying the Virus Spread Simulator Using Simulink Compiler
- 5 AUG** Getting Started with Simulink Compiler
- 16 JUN** Creating and Editing Simulink Models in MATLAB Online!

Categories

- What's new? 151
- Debugging 30
- Modeling 47
- Code Generation 44
- Simulink Tips 82

[more](#) ▾

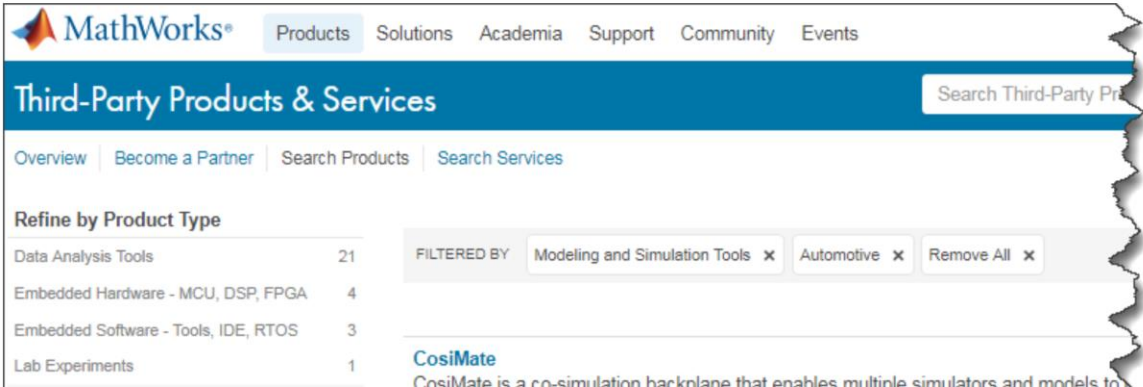
< MATLAB Online, MATLAB Mobile, MATLAB... Simulation Data Inspector in R2018a >

Communicating with an External Application for Co-Simulation

Posted by [Guy Rouleau](#), May 1, 2018 👁 128 views (last 30 days) | 👍 0 likes | 💬 7 comments

Today I am describing an example that I recently submitted to [MATLAB Central](#) and [GitHub](#) with the help of my colleague Haihua Feng: [Example implementation of Co-simulation using Simulink](#).

In case you did not know, MathWorks' website lists a lot of [third-party modeling and simulation tools](#) from MathWorks [Connection Partners](#).



The screenshot shows the MathWorks website navigation bar with links for Products, Solutions, Academia, Support, Community, and Events. Below the navigation bar is a search bar for 'Third-Party Products & Services'. The page content includes a table of product categories and a filter section. The filter section shows 'Modeling and Simulation Tools' and 'Automotive' selected, with a 'Remove All' button. The table lists the following categories and counts:

Product Type	Count
Data Analysis Tools	21
Embedded Hardware - MCU, DSP, FPGA	4
Embedded Software - Tools, IDE, RTOS	3
Lab Experiments	1

Below the table, the 'CosiMate' product is listed with a brief description: 'CosiMate is a co-simulation backbone that enables multiple simulators and models to...'

FMI (Functional Mock-up Interface)



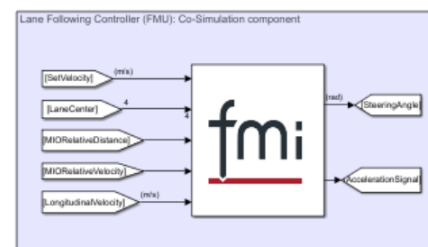
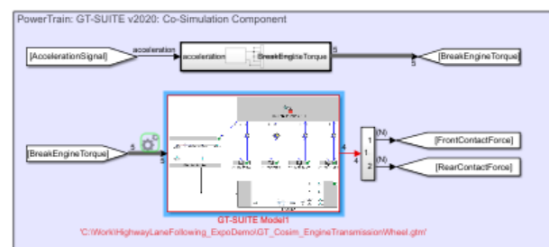
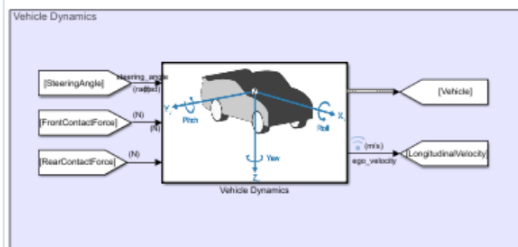
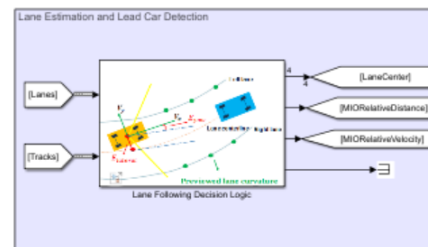
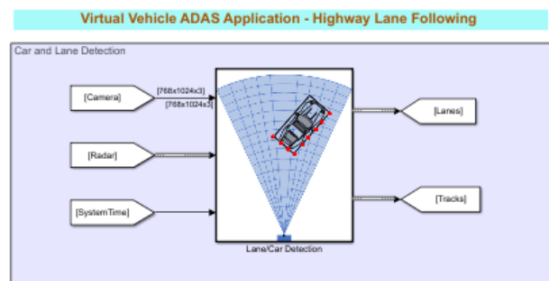
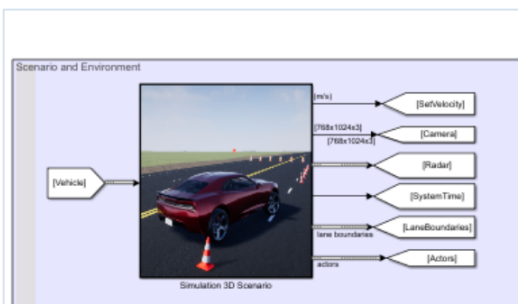
- FMI是与工具无关的系统动态模型格式
 - FMU是一个将模型以FMI格式导出打包的压缩文件
- 支持FMU导入导出的工具在持续增长
- Simulink支持FMI V1.0和V2.0版
- 支持联合仿真和模型交换形式导入FMU模型

		FMU Export				FMU Import	
Name	License	Platforms	Co-Simulation	Model Exchange	Co-Simulation	Model Exchange	
MATLAB® Simulink®	\$	Apple, Windows	1.0, 2.0	1.0, 2.0	1.0, 2.0	1.0, 2.0	

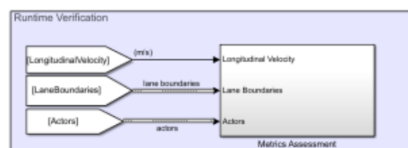
演示示例 – 虚拟车辆的ADAS应用

- 使用S函数和FMU将外部组件集成进来

环境模型



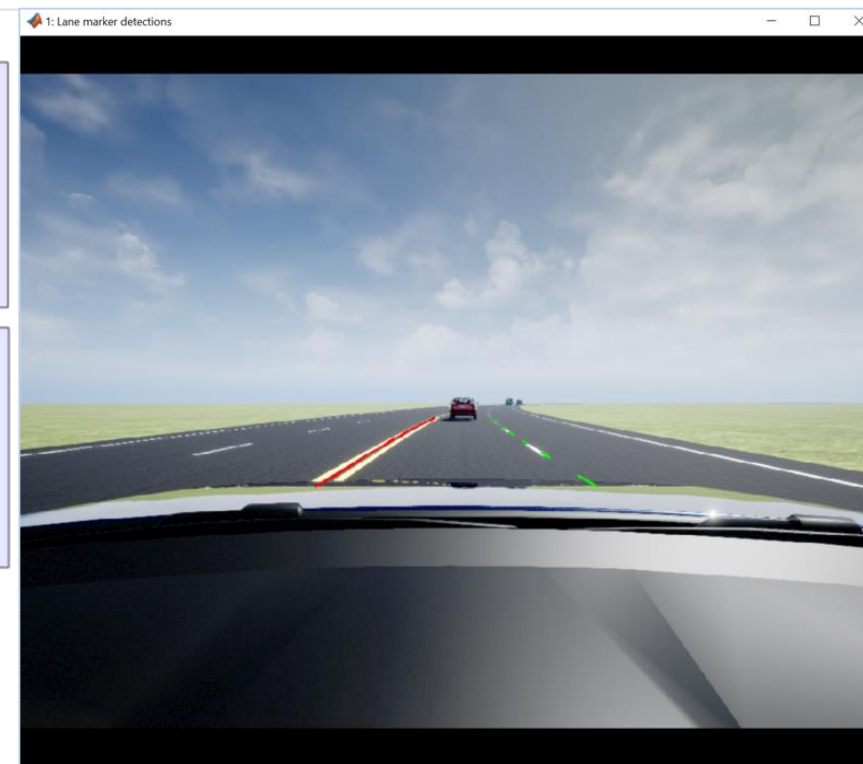
车辆本体



控制算法

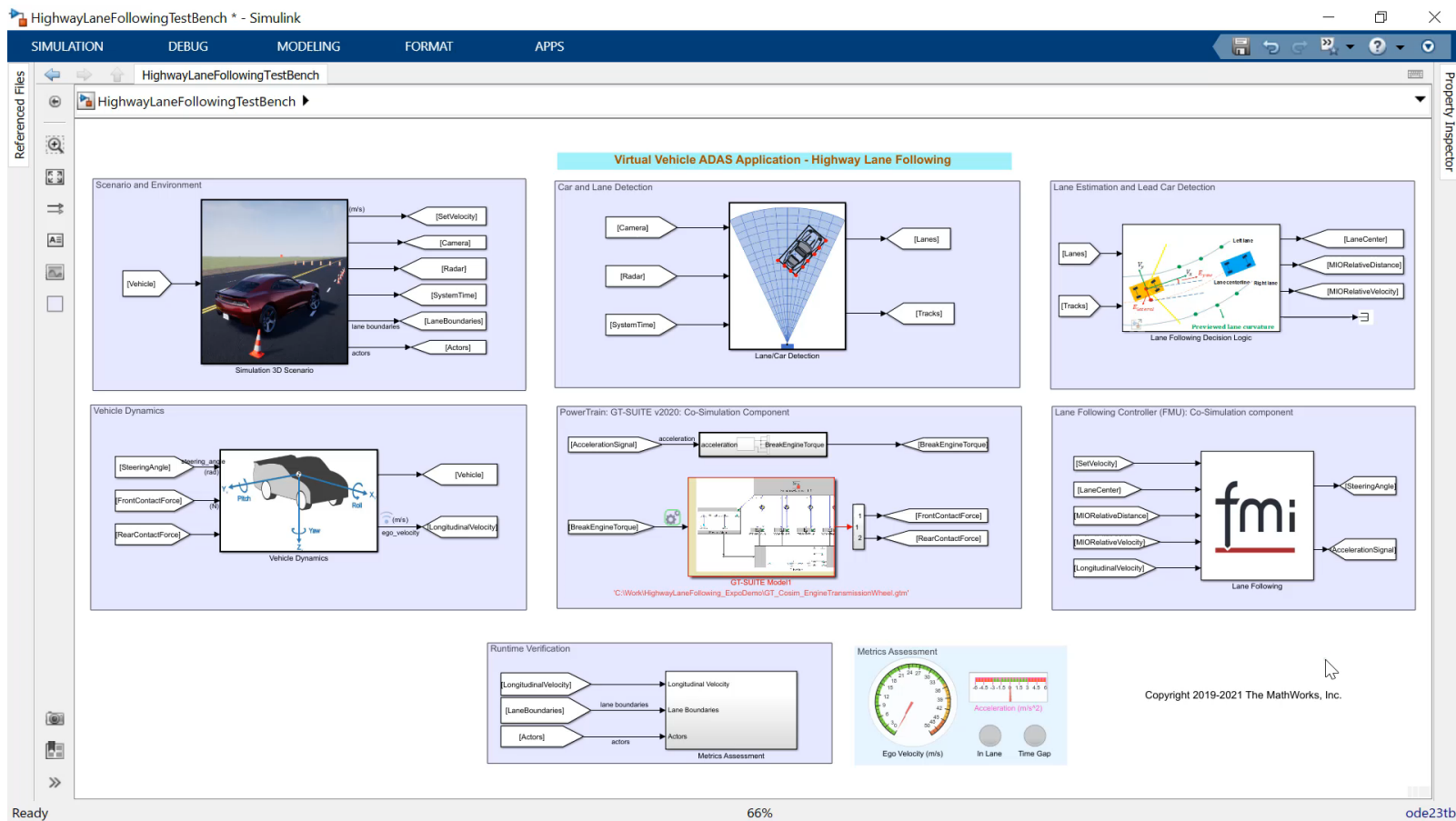
Copyright 2019-2021 The MathWorks, Inc.

感知算法



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



Copyright 2019-2021 The MathWorks, Inc.

演讲提纲



与外部仿真工具的接口



联合仿真的鲁棒性



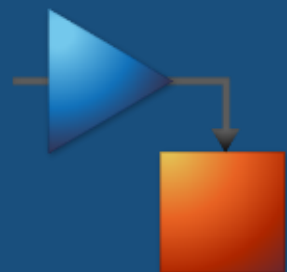
引入自定义C/C++代码



系统级仿真加速

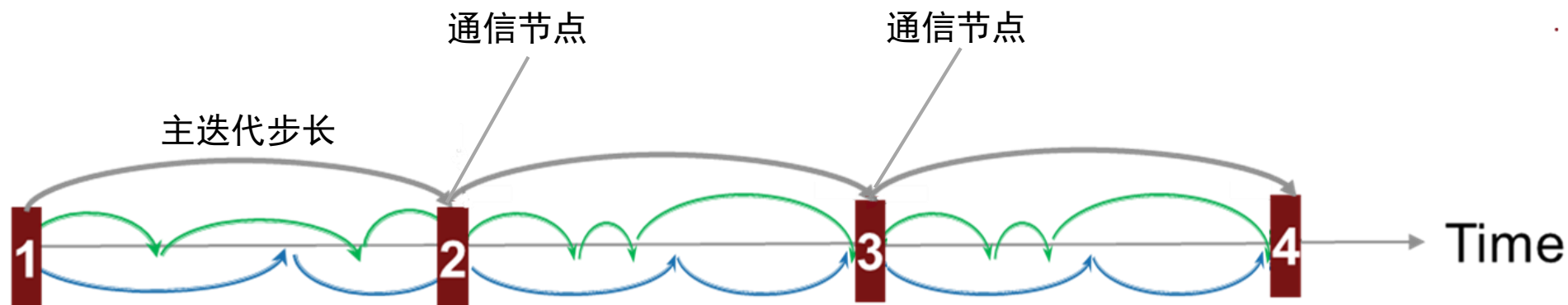
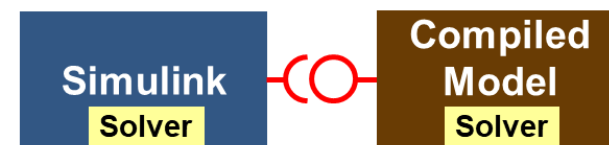


联合仿真的鲁棒性



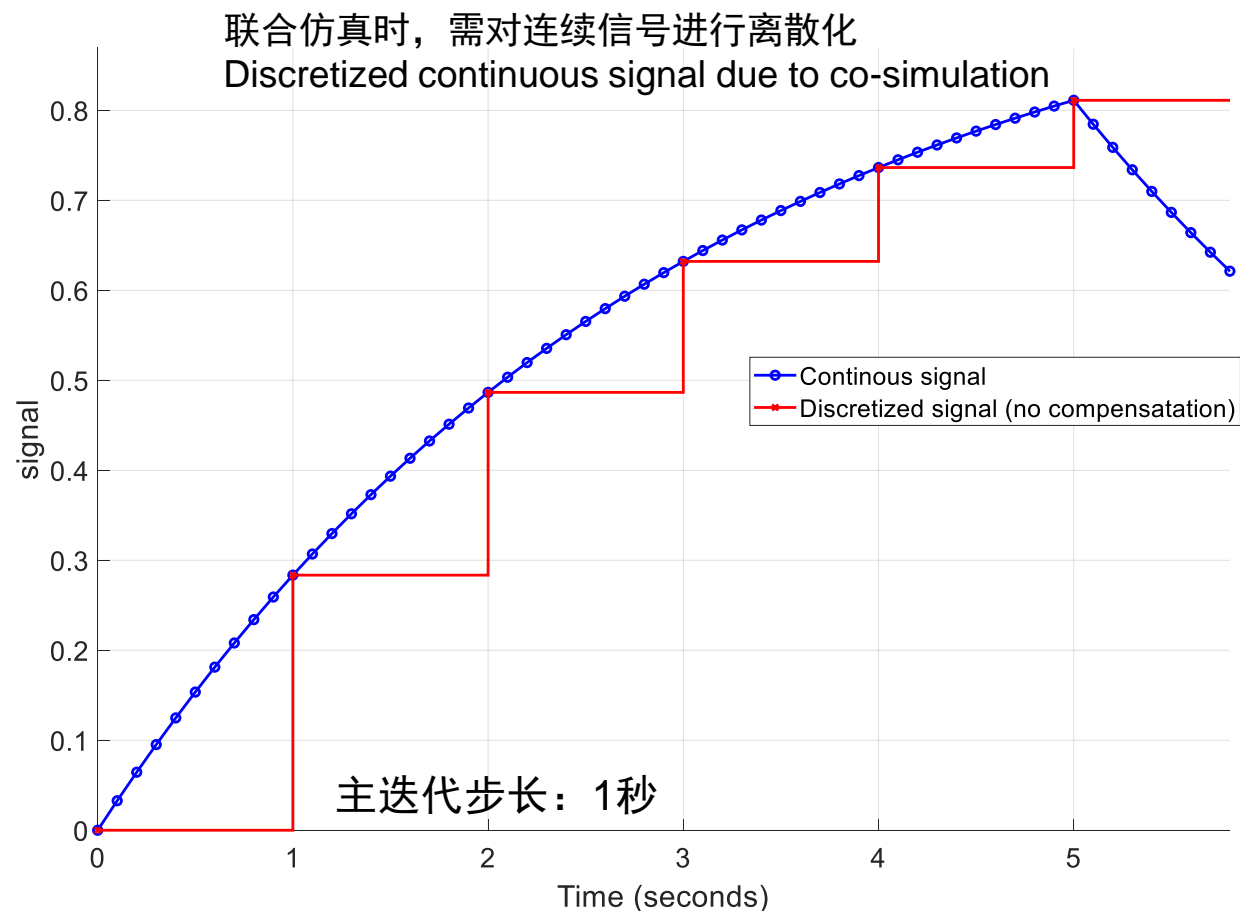
联合仿真

- 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

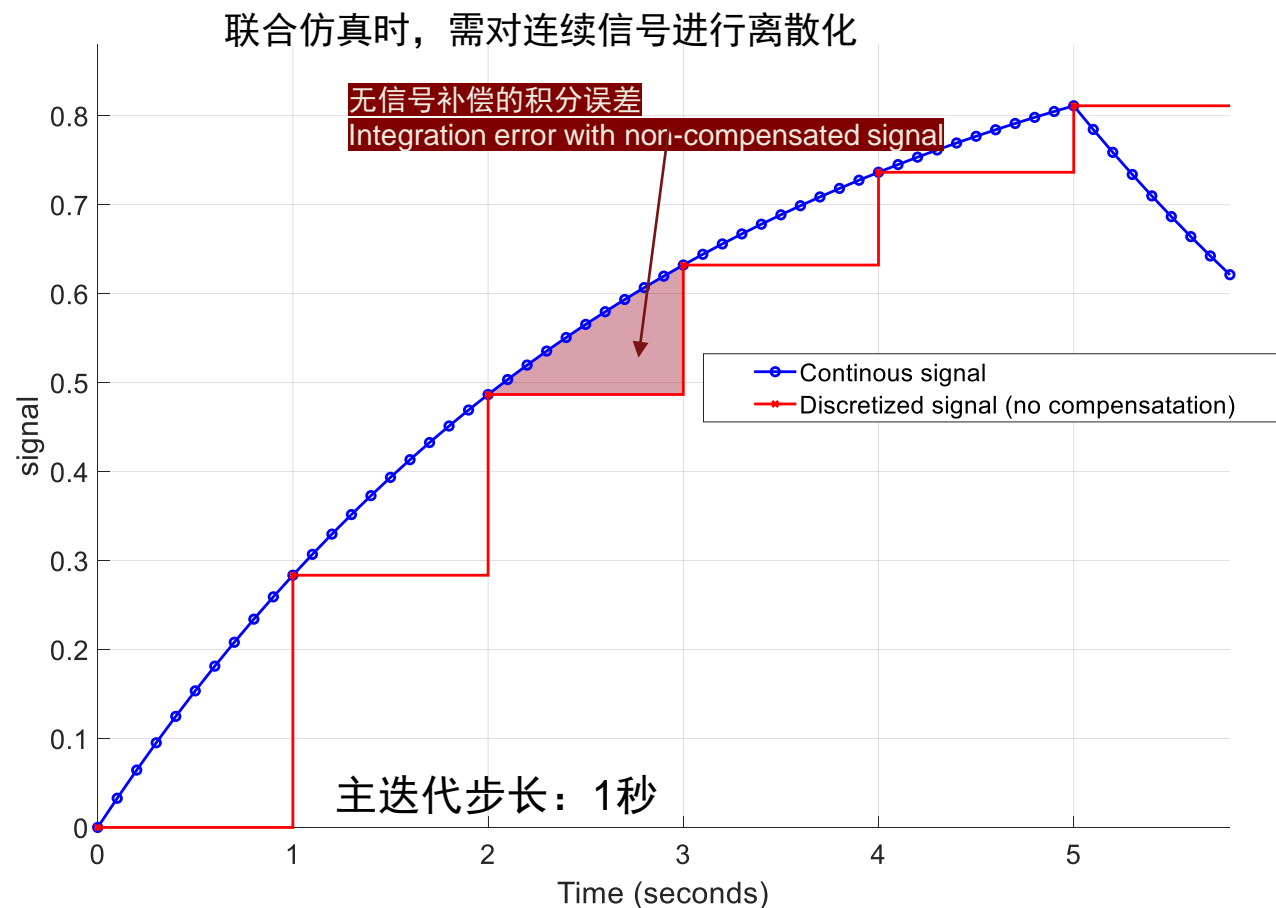


The un-compensated signal (red line) deviates from the ideal, continuous signal (blue line) due to discretization

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



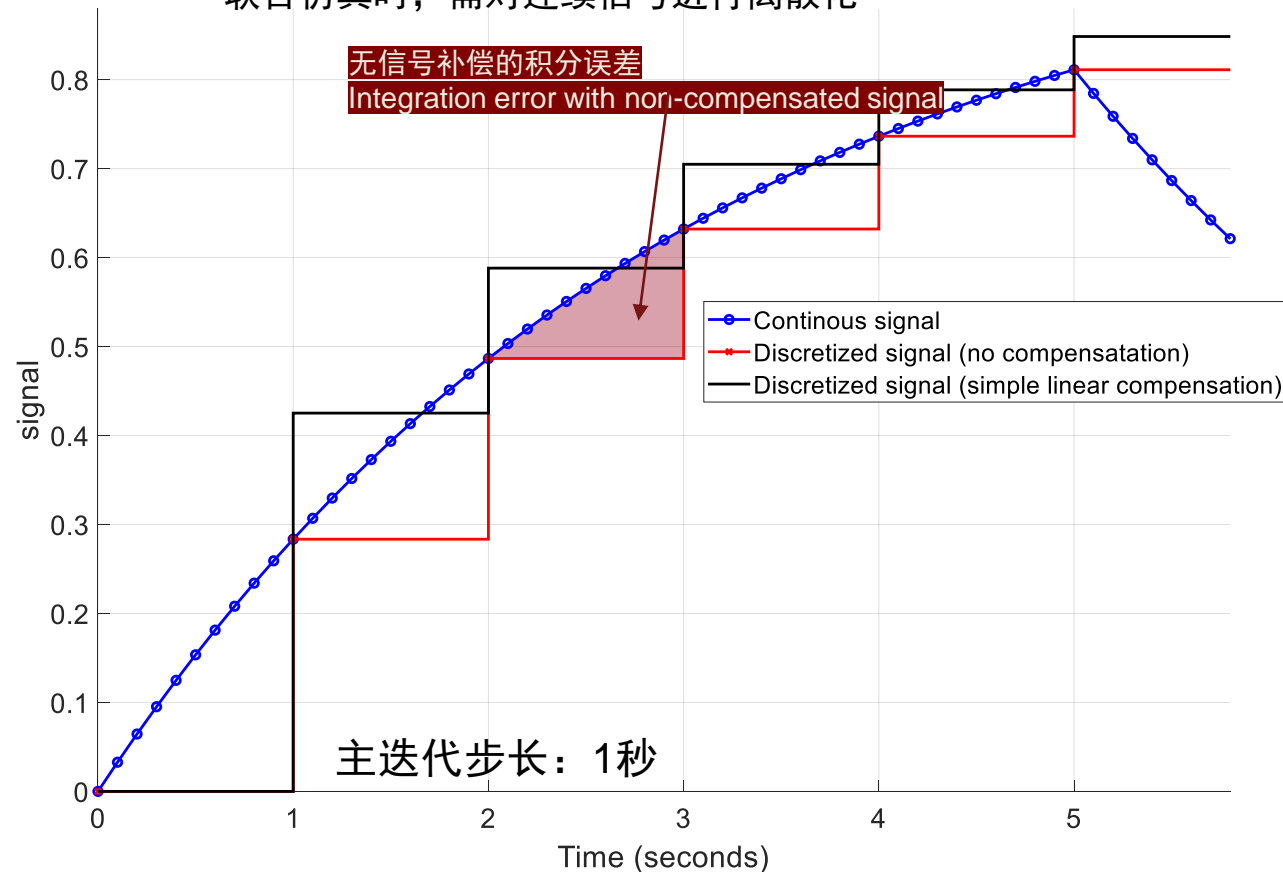
The un-compensated signal (red line) deviates from the ideal, continuous signal (blue line) due to discretization
因为离散化的原因，无补偿的信号（红线）与理想的连续信号（蓝色）发生偏差

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



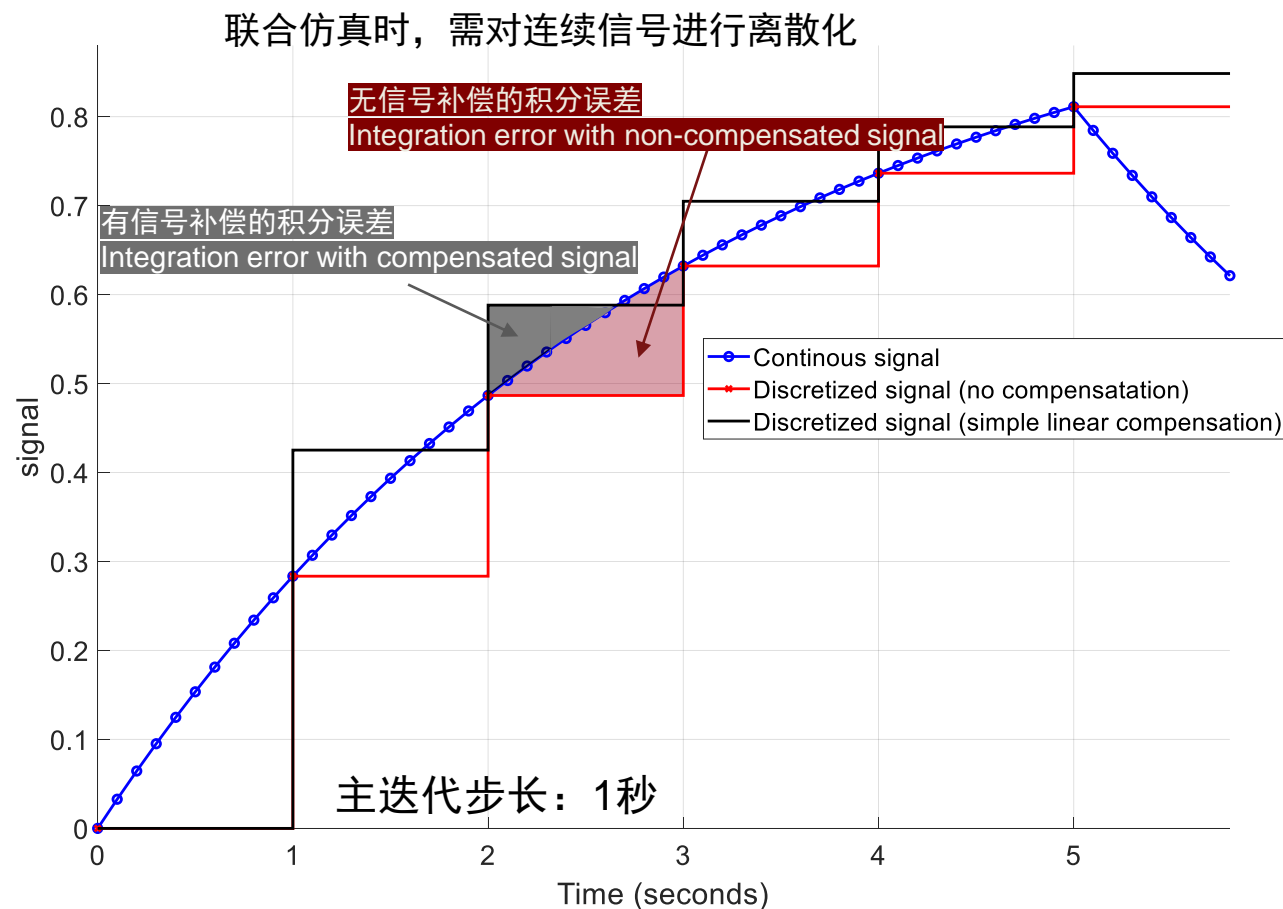
联合仿真时，需对连续信号进行离散化



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

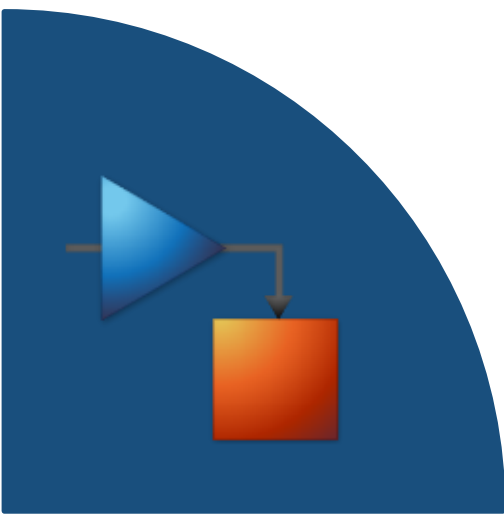
```
template<class InputString, class OutputString>
bool unhexlify(const InputString& input, OutputString& output) {
    if (input.size() % 2 != 0) {
        return false;
    }
    output.resize(input.size() / 2);
    int j = 0;
    auto unhex = [](char c) -> int {
        return c >= '0' && c <= '9' ? 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) {
        int hi = unhex(input[i]);
        int lo = unhex(input[i + 1]);
        if (hi < 0 || lo < 0) {
            return false;
        }
        output[j++] = (hi << 4) | lo;
    }
    return true;
}
```

Internal Libraries

Vendor Libraries



)+ lowBi

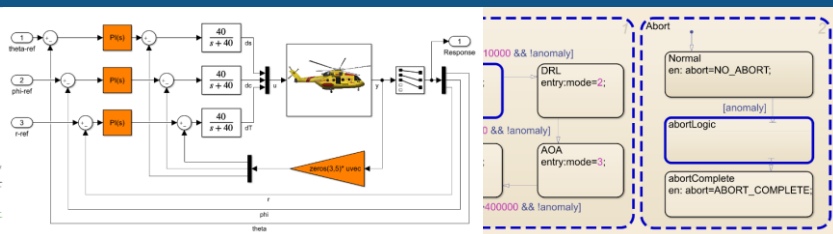


引入自定义C/C++代码

模型

```

1 % Predicted state and covariance
2 x_prd = A * x_est;
3 p_prd = A * p_est * A' + Q;
4
5 % Estimation
6 S = H * p_prd * H' + R;
7 B = H * p_prd';
8 klm_gain = (S \ B)';
9
10 % Estimated state and covariance
11 x_est = x_prd + klm_gain * (z - H *
12 p_est = p_prd - klm_gain * B * p_pr
13
14 % Compute the estimated measurement
15 y = H * x_est;
    
```



C/C++ 库

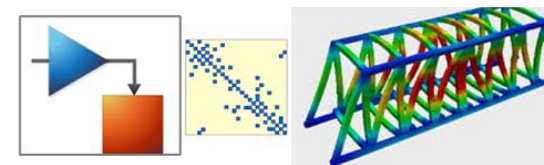
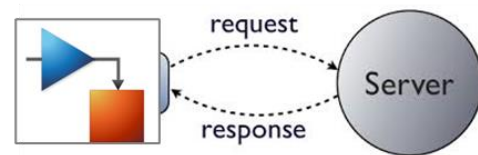
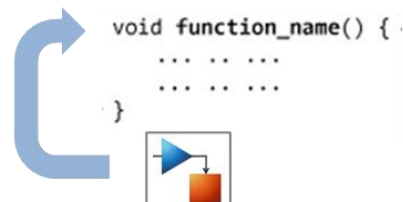
手写代码

```

templateclass InputString, class OutputString,
class anomaly(const InputString& input, OutputString& output) {
    if (input.size() & 2 != 0) {
        return false;
    }
    output.resize(input.size());
    int j = 0;
    auto umhex = [](const char* c) {
        return c == '0' || c == '1' ? c - '0' :
            c == 'A' || c == 'a' ? c - 'A' + 10 :
            c == 'X' || c == 'x' ? c - 'X' + 16 :
            '-1';
    };
    for (size_t i = 0; i < input.size(); i += 2) {
        int highbits = umhex(input[i]);
        int lowbits = umhex(input[i + 1]);
        if (highbits < 0 || lowbits < 0) {
            return false;
        }
        output[j++] = (highbits << 4) + lowbits;
    }
    return true;
}
    
```



引入自定义C/C++代码



Basic

Advanced

R2018b

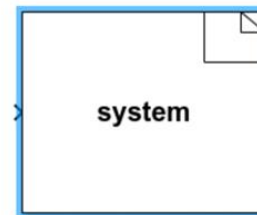


C Caller

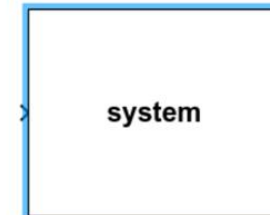
R2020a



C Function



S-Function Builder

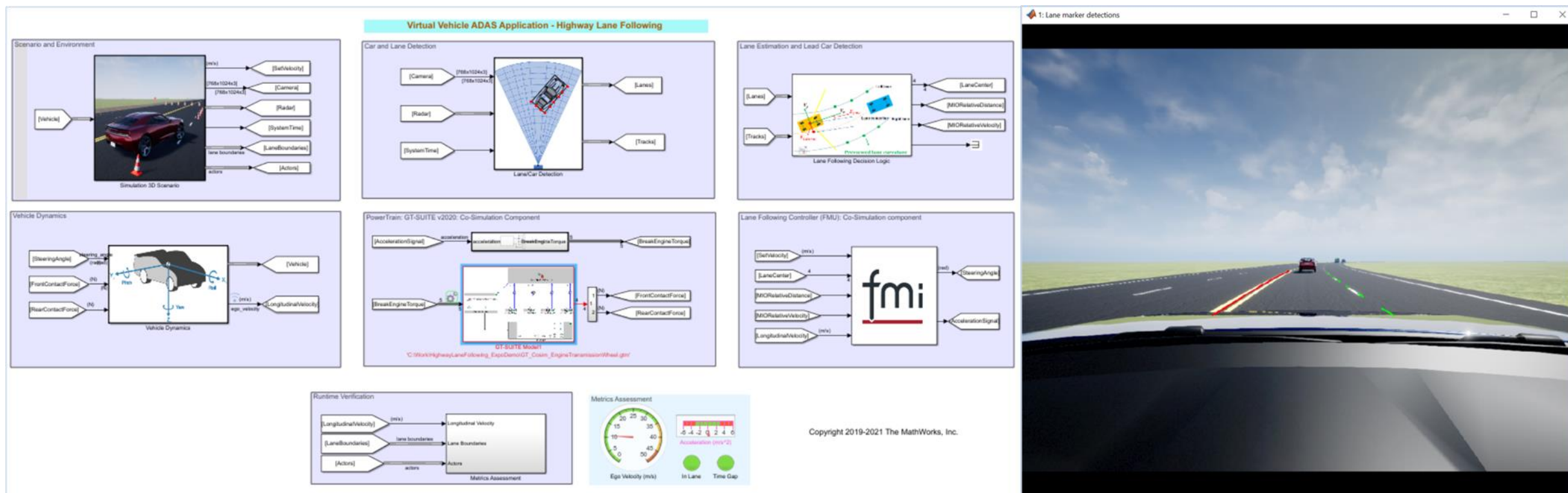


S-Function

- 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

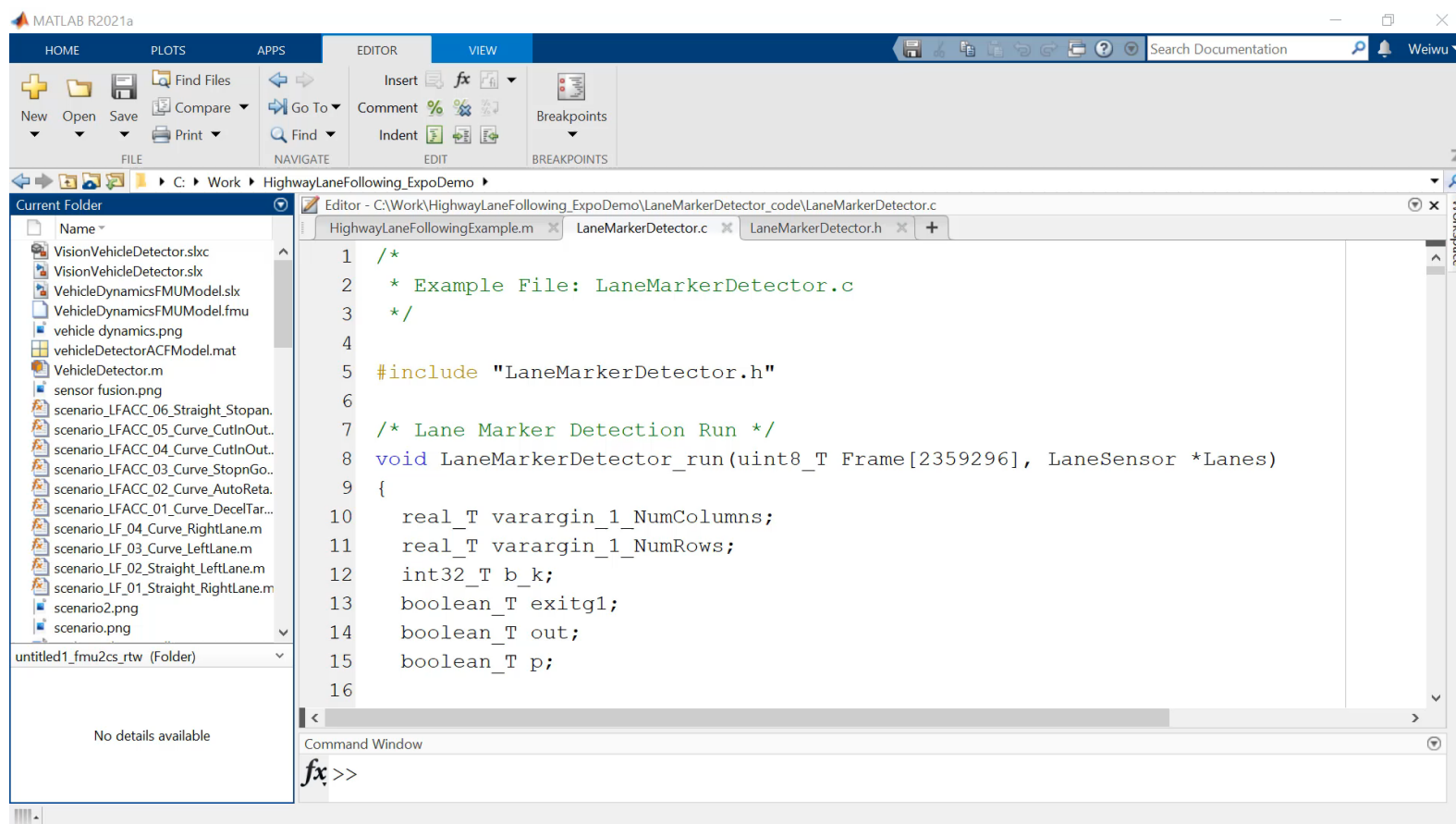
演示示例 – 虚拟车辆的ADAS应用

- 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



The screenshot displays the MATLAB R2021a environment. The left sidebar shows a file explorer with a project structure including folders like 'untitled1_fm2cs_rtw' and various files such as 'VisionVehicleDetector.slxc', 'VehicleDynamicsFMUModel.slx', and several scenario files. The main editor window is open to a C file named 'LaneMarkerDetector.c'. The code in the editor is as follows:

```
1 /*
2  * Example File: LaneMarkerDetector.c
3  */
4
5 #include "LaneMarkerDetector.h"
6
7 /* Lane Marker Detection Run */
8 void LaneMarkerDetector_run(uint8_T Frame[2359296], LaneSensor *Lanes)
9 {
10     real_T varargin_1_NumColumns;
11     real_T varargin_1_NumRows;
12     int32_T b_k;
13     boolean_T exitg1;
14     boolean_T out;
15     boolean_T p;
16 }
```

At the bottom of the window, the Command Window shows the MATLAB prompt `fx>>`.

演讲提纲



与外部仿真工具的接口



联合仿真的鲁棒性

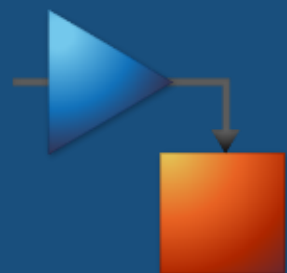
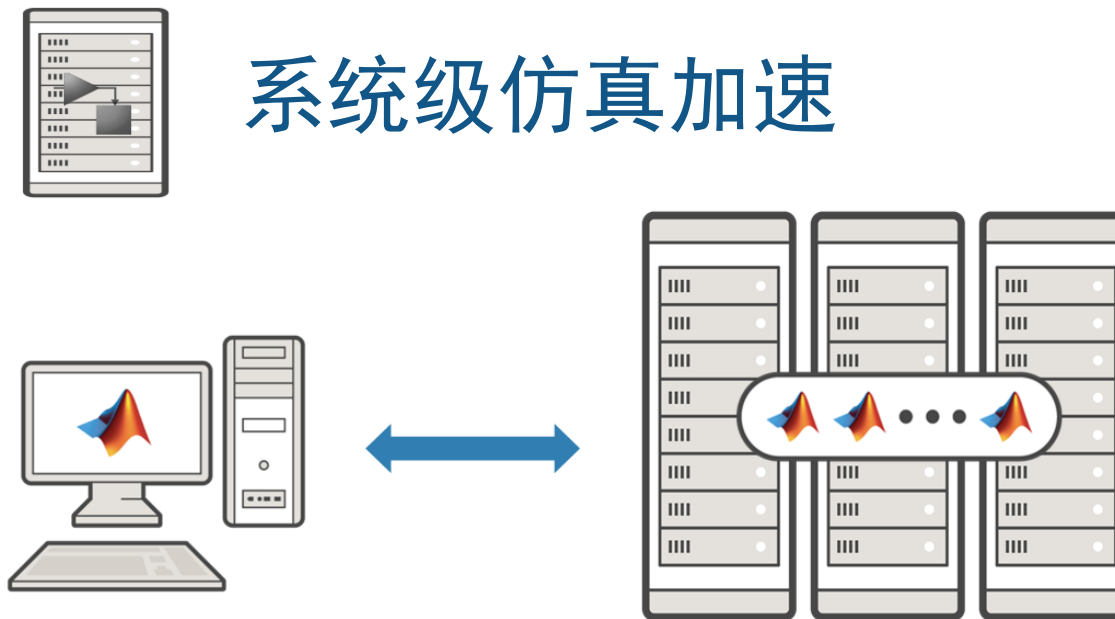


引入自定义C/C++代码



系统级仿真加速

系统级仿真加速



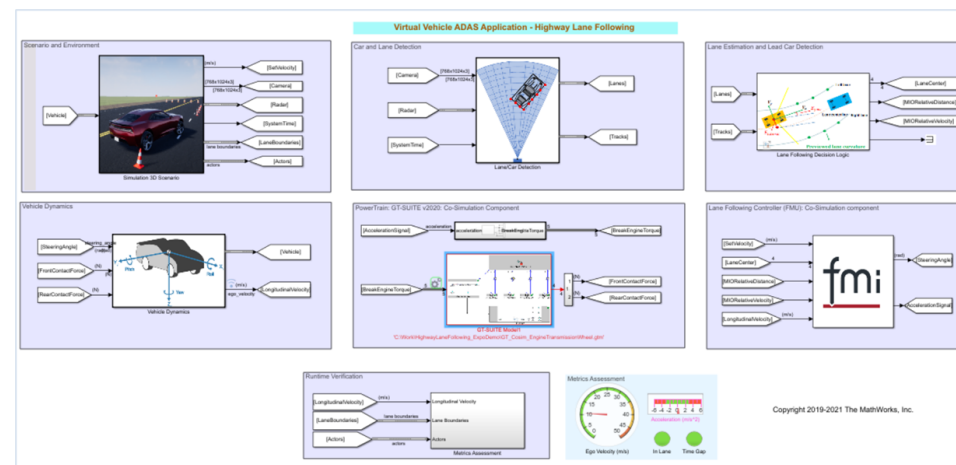
大规模系统仿真

- 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

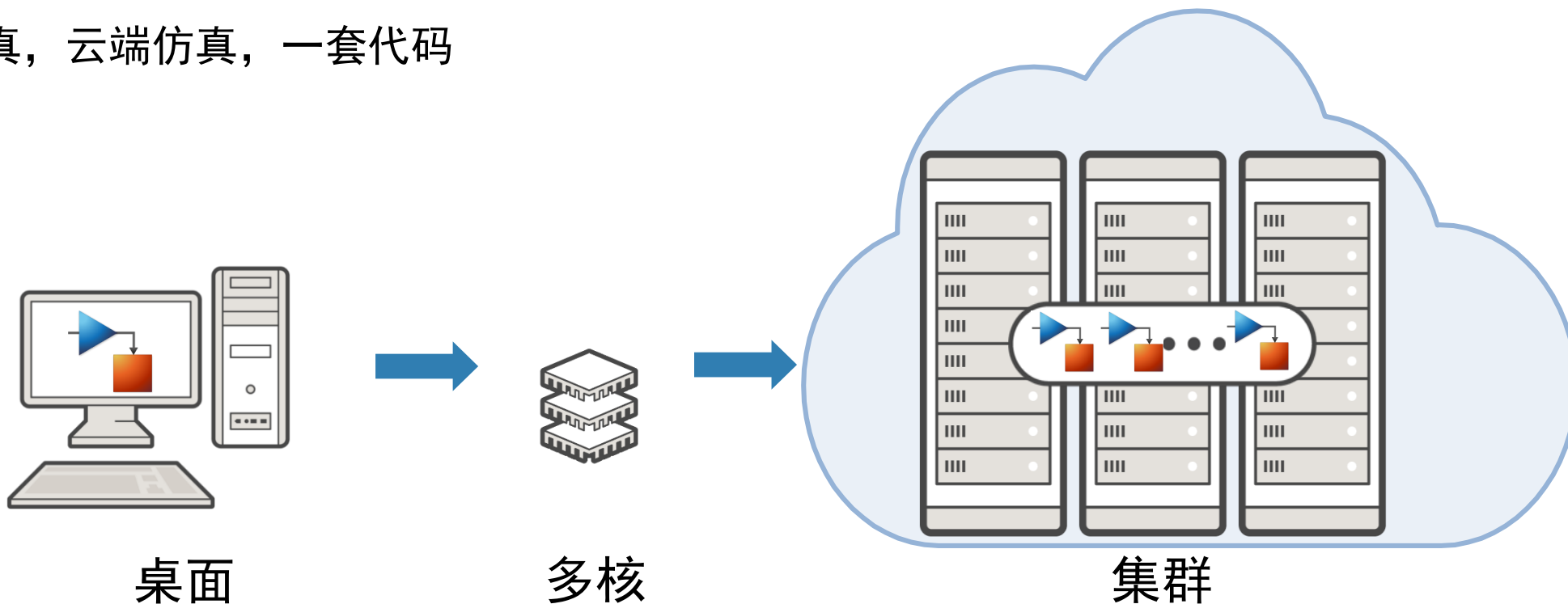
10个驾驶循环
10种天气条件
10种车辆载荷
10种传动比
10个轮胎尺寸

-> 100,000 仿真



大规模系统仿真

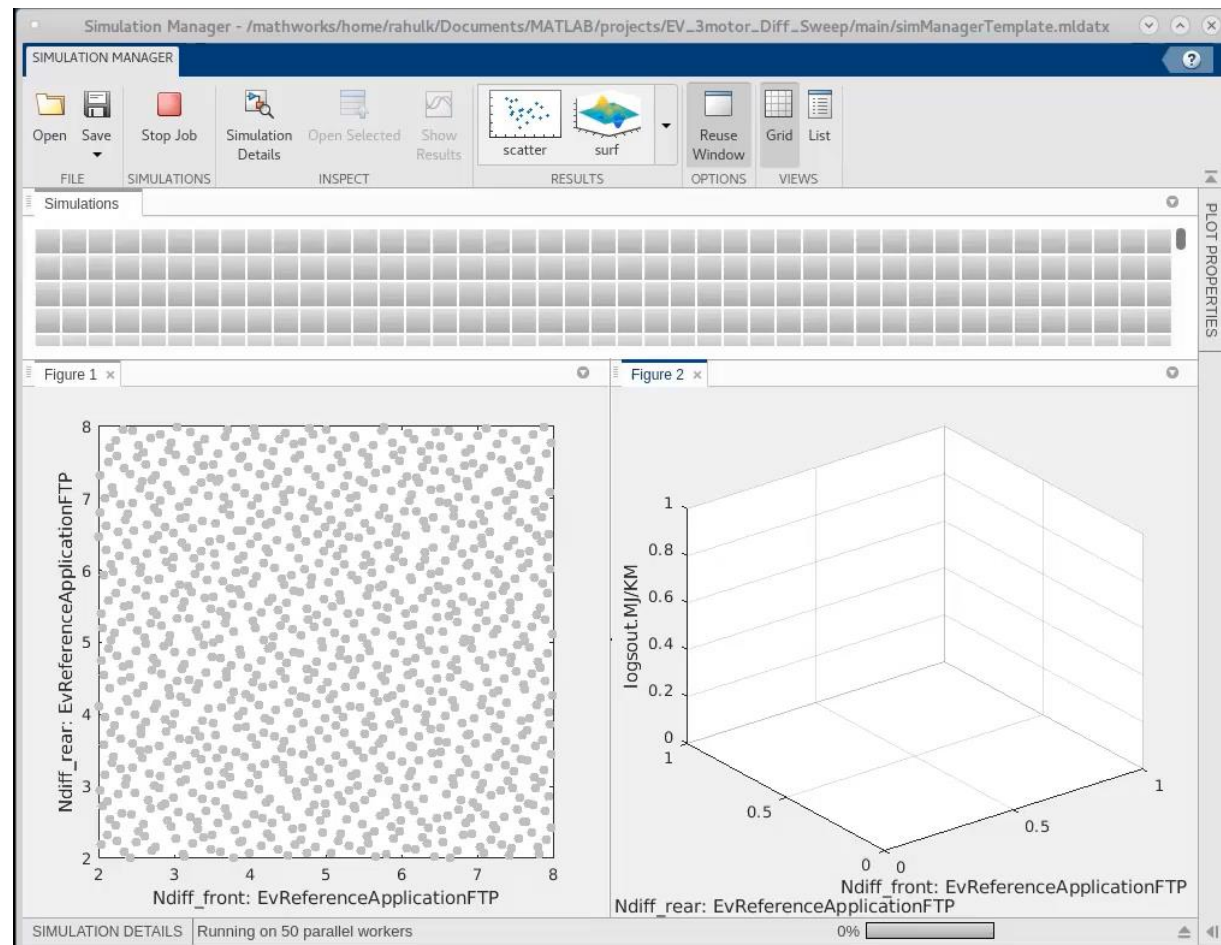
- 桌面仿真，云端仿真，一套代码



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

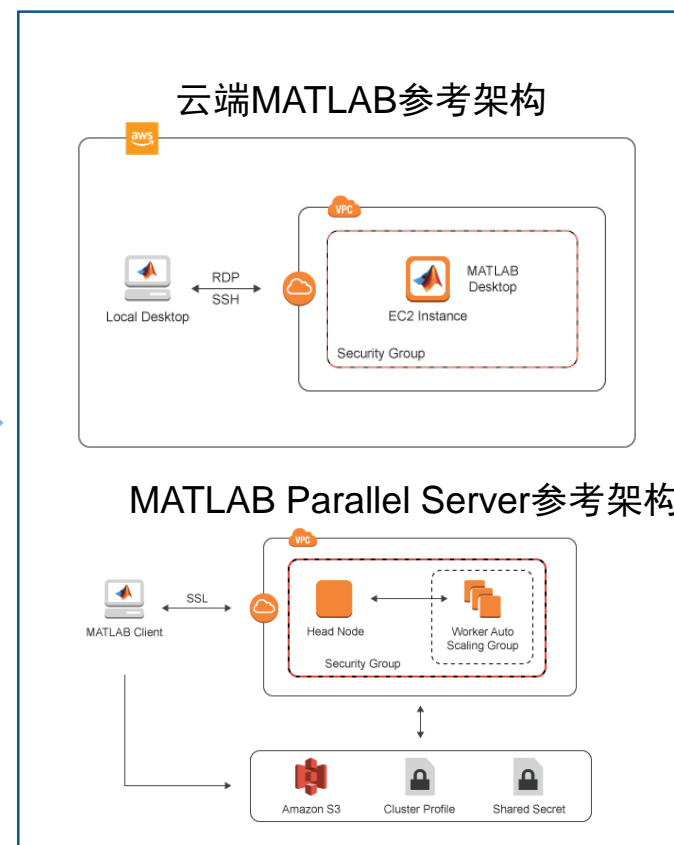
大规模系统仿真

- 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



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





MATLAB EXPO 2021

感谢您的聆听

