

MATLAB EXPO

2021

Integrating external simulation components
with Simulink

Brad Hieb



Key Takeaways

Simulink is an integration platform for simulating your complex, heterogeneous, and multi-domain systems:

- Standard-based interfaces to integrate 3rd party simulation tools/models
- Co-simulation numeric robustness with automatic signal compensation
- Bringing in custom C/C++ code made easy
- Utilizing parallel simulation capabilities to speed up system-level simulations



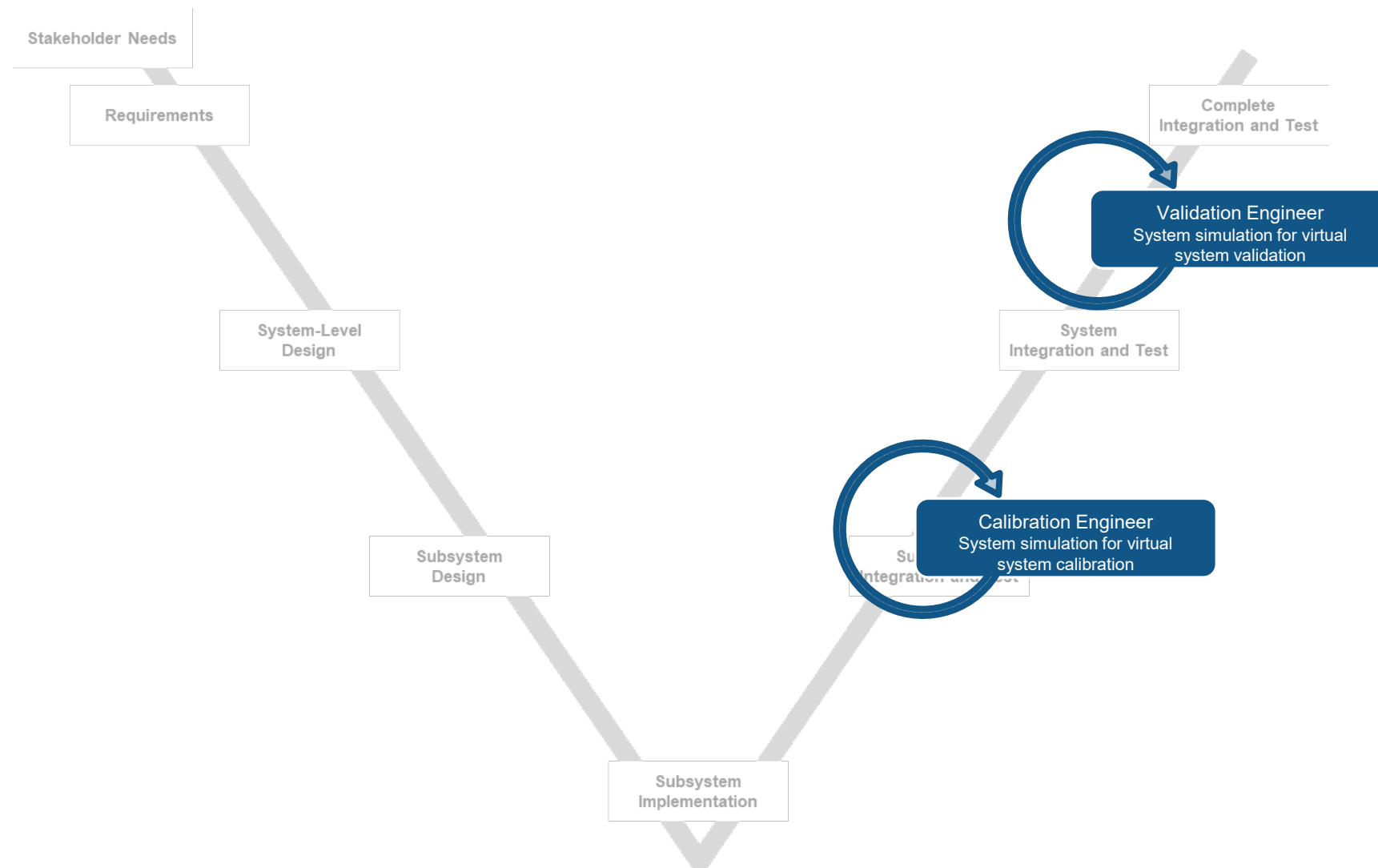
Motivation

System-level simulation is becoming pervasive in your product development cycle



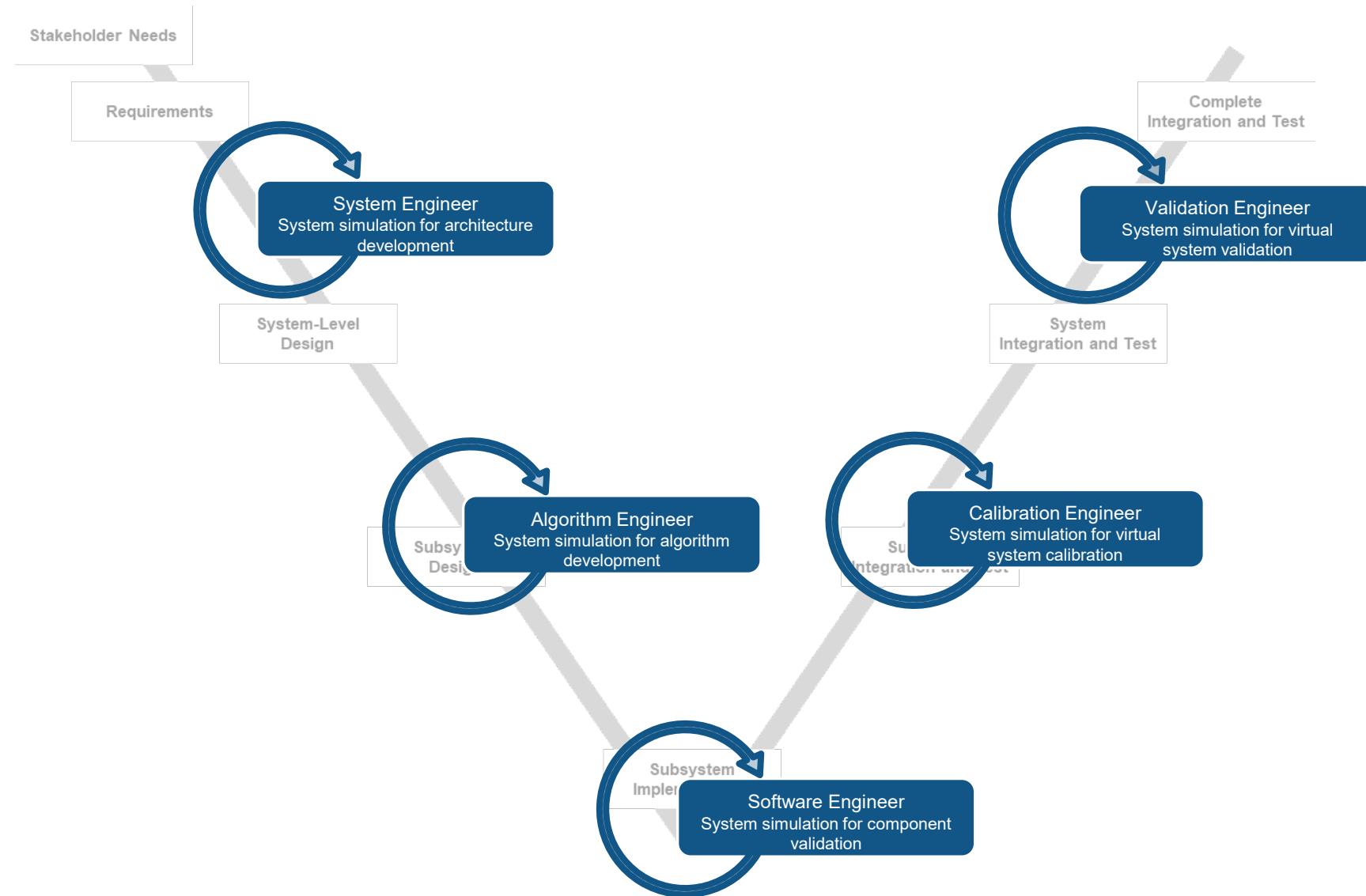
Motivation

System-level simulation is becoming pervasive in your product development cycle



Motivation

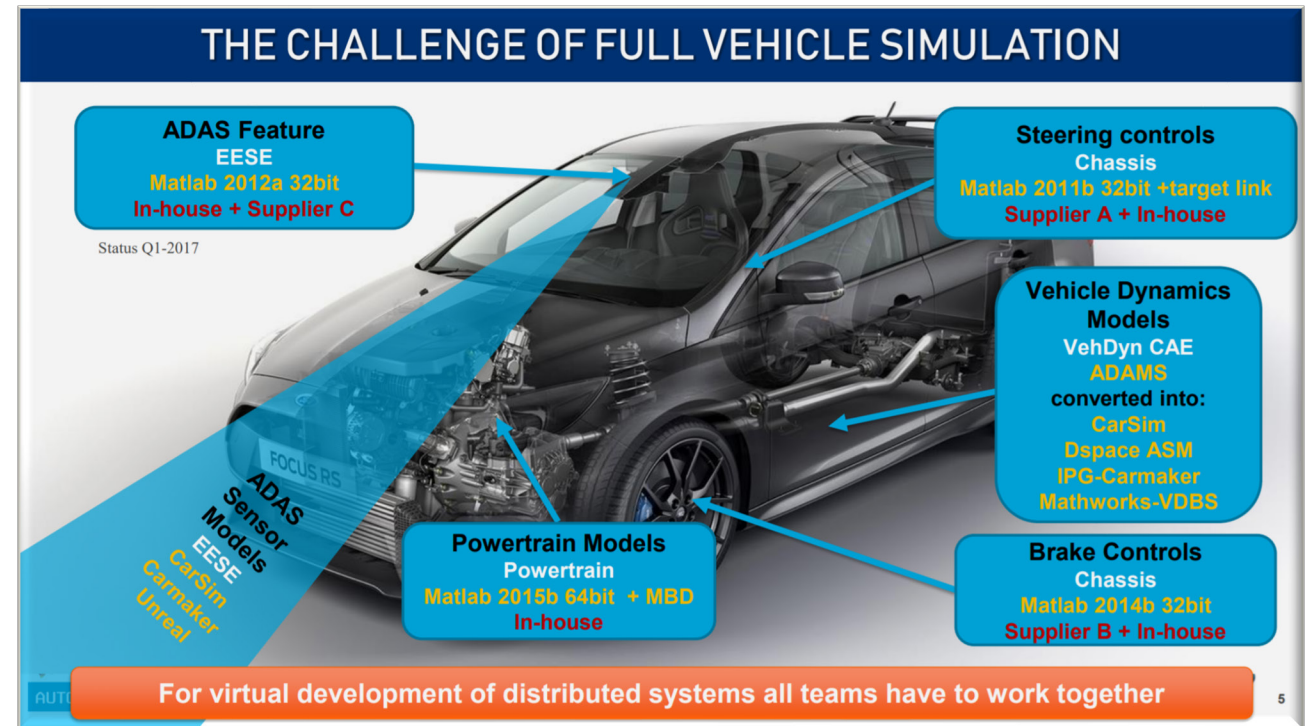
System-level simulation is becoming pervasive in your product development cycle



Motivation

Increasing challenges when simulating complex systems

- Multi-domain, inter-disciplinary design
- Model re-use among suppliers, clients and collaborators while hiding design details
- Performance: the need to speed up simulations for quick insights



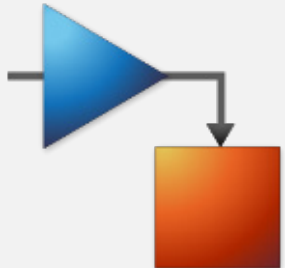
[Model-Based Agility with Ford Automated System Simulation Toolchain \(FASST\)](#)

MathWorks Automotive Conference 2020

Agenda

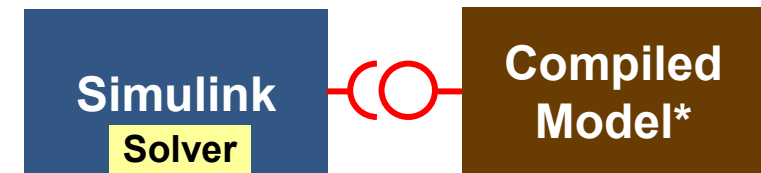
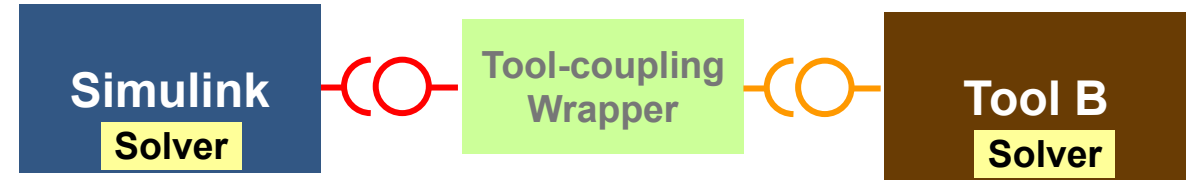
- Interfaces to external simulation tools
- Robust co-simulation
- Bringing in custom C/C++ code
- Scale up system-level simulations

Interfaces to external simulation tools



White-box and Black-box Integration

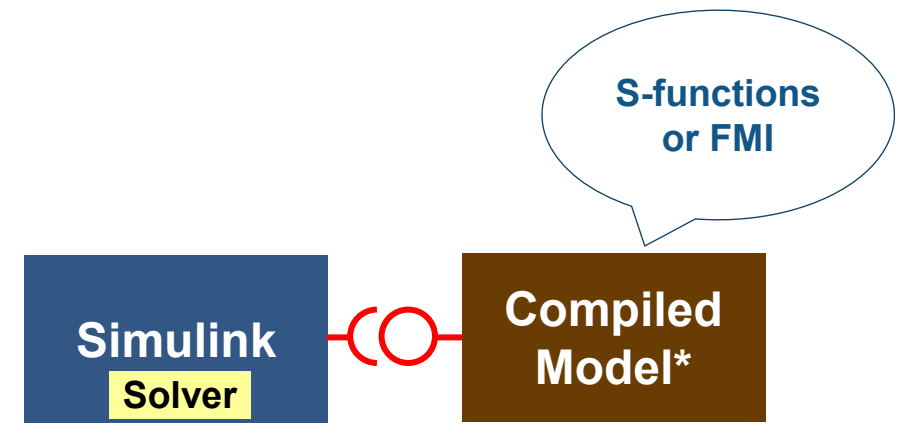
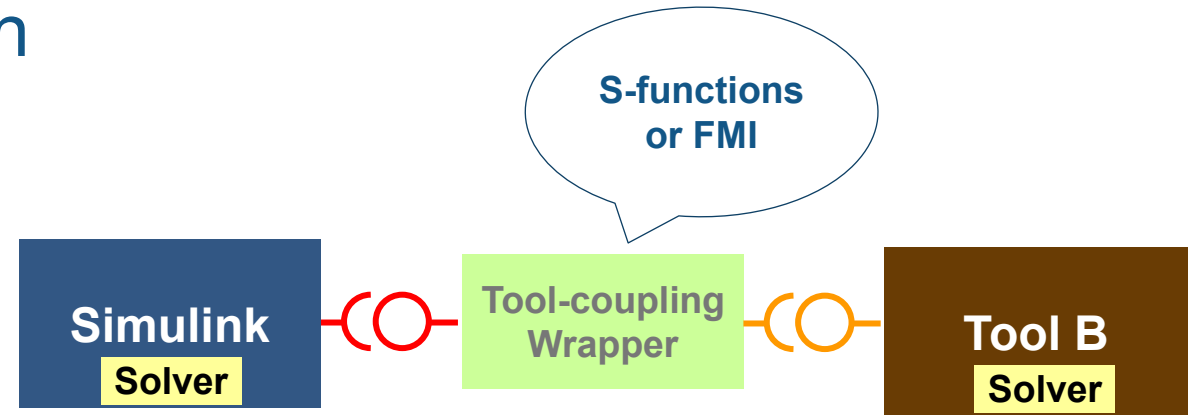
- White-box (tool-coupling)
 - Both Simulink and the external tool are running during simulation
- Black-box (compiled model)
 - Only Simulink is running during simulation
 - The 3rd party model is a component inside Simulink



* With or without solver

White-box and Black-box Integration

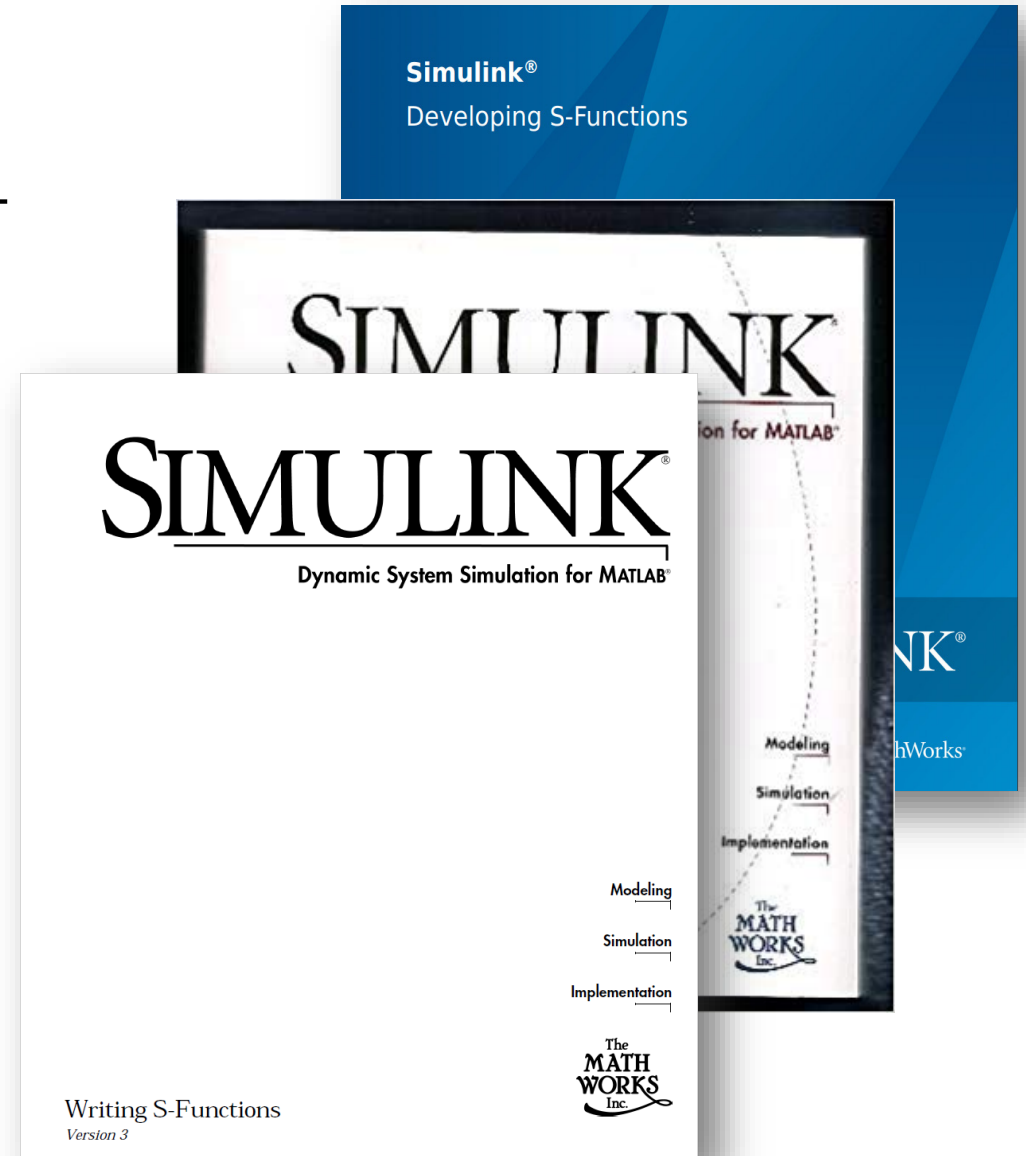
- White-box (tool-coupling)
 - Both Simulink and the external tool are running during simulation
- Black-box (compiled model)
 - Only Simulink is running during simulation
 - The 3rd party model is a component inside Simulink
- Standard-based interfaces used for both styles
 - S-functions
 - Functional Mockup Interface (FMI)



* With or without solver

S-functions Interface

- Build custom dynamic systems in MATLAB, or C/C++
- Supports all Simulink semantics
- Well validated by industry for 20+ years
- The de facto standard to couple external tools with Simulink



S-functions Interface


- Many of the MathWorks' Connection Partner simulation tools (150+) provide prebuilt co-simulation interfaces using S-functions
- S-functions based co-simulation interface is also available with some non-partner's tools

The screenshot displays the MATLAB Third-Party Products & Services interface. At the top, there is a search bar labeled "Search Third-Party" and a dropdown menu for "Third-Party Products". Below this, a "FILTER" button is visible. The main content area lists various simulation tools, each with a title, a brief description, and the company name. The tools listed include:

- Biomechanics of E...**: Biomechanics of E model consists of... Company: BoB B
- Bonsai**: Deep reinforcement... The Bonsai platform complete develop... Company: Bonsai
- Cadence Virtuoso**: Accelerate proces... Designed to help t the advanced desi... Company: Caden
- CANoe**: Tool for design an... CANoe is a distrib application behavi... Company: Vector
- CarMaker for use**: Open integration a... CarMaker for Sim and Simulink® m... Company: IPG At
- CarSim, TruckSim, BikeSim**: Simulation of the vehicle dynamics for SIL, HIL, and Driving Simulators
- FTire and FTire/li**: Physical tire mode... FTire (Flexible Rin for vehicle comfort... Company: cosin s
- Gas Dynamics ar**: Dynamic simulatio... ACUSYS simulate network, induced t... Company: SATE :
- GENESYS**: Software for mode... GENESYS deliver collaboratively buil... Company: Vitech
- GL Studio**: Transition high-en... GL Studio © is a p their software prod... Company: The Di
- gPROMS Block C**: Process modeling, gPROMS is an ad complexity involv... Company: Proces
- GT-SUITE**: Engine, powertrain, and vehicle engineering simulation software
- Siemens Simcenter TIRE**: Providing a versatile and cost-efficient approach to tire modeling... The tire is a highly non-linear vehicle component that has a significant effect on the behavior of a vehicle. As the numb...
- rFpro**: Photo-realistic 3D... rFpro allows custo photo-realistic, hig... Company: rFpro L
- Riviera-PRO**: High-performance... Riviera is a high-p batch processing... Company: Aldec,
- Saber**: Design and analys... Saber... Company: Synop
- Sensors and Elec**: SEWES is a few-o... SEWES is a few-o number of sensors... Company: Coun
- SIDLAB**: Simulation of sour... SIDLAB is a comb design and optimiz... Company: SIDLA
- Siemens Simcenter TIRE**: Providing a versatile and cost-efficient approach to tire modeling... The tire is a highly non-linear vehicle component that has a significant effect on the behavior of a vehicle. As the numb...
- Simcenter Amesim**: Mechatronic system simulation software... Simcenter Amesim™ software is an integrated, scalable mechatronic system simulation platform that allows design engineers to virtually assess and optimize systems' performance... Company: Siemens Industry Software
- Simcenter MAGNET**: Low-frequency electromagnetics simulation software... Simcenter MAGNET 2D/3D is a powerful simulation software which engineers and scientists worldwide use for the design of motors, sensors, transformers, actuators, solenoids or any component with permanent... Company: Mentor, a Siemens Business
- SIMPACK**: Complete multibody simulation in combination with MATLAB... SIMPACK is a general-purpose, nonlinear, 3D multibody simulation tool. It is designed to simulate mechanical systems, analyze vibrational behavior, calculate forces and accelerations, and describe and... Company: SIMPACK AG
- SIMTEST Toolbox**: Vibration analysis and control for multi-axis simulation testing... The SIMTEST Toolbox addresses the needs of noise, vibration, and structural durability test laboratories for multichannel dynamic analysis and control. It provides advanced vibration analysis and control... Company: Simulation Techniques, Inc.
- SimulationX**: High-end modeling tool for simulating nonlinear, dynamic effects... SimulationX is ITI's multiphysics software tool for system simulation. It is used for modeling, analyzing, and optimizing complex, dynamic, nonlinear systems. Simulation models are defined interactively... Company: ITI GmbH
- SimWise 4D**: Simulation and validation of functional performance for mechanical parts and assemblies

To learn more to use S-functions to communicate with an external application

- Example template to use S-functions as the tool-coupling interface
 - [Available on “Guy on Simulink”](#)



Guy on Simulink
Simulink & Model-Based Design

you 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](#) ▾

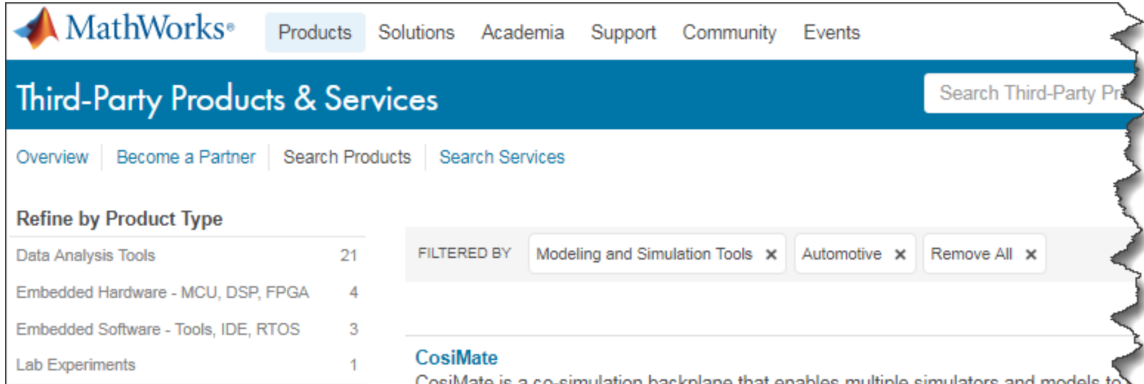
< MATLAB Online, MATLAB Mobile, MATLAB... Simulation Data Inspector in R2018a >

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 'Refine by Product Type' section with a table of categories and counts, and a 'FILTERED BY' section with active filters for 'Modeling and Simulation Tools' and 'Automotive', along with a 'Remove All' button. The first result listed is 'CosiMate'.

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

Functional Mock-up Interface (FMI)



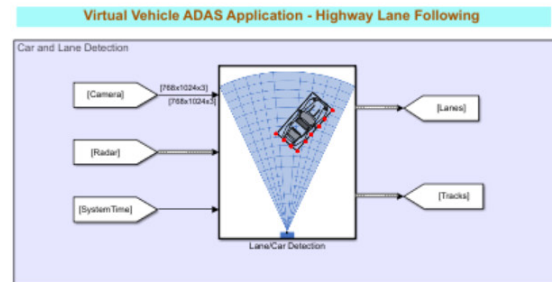
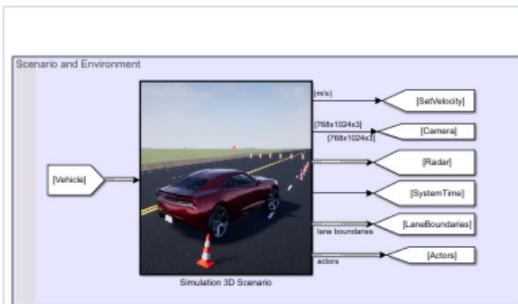
- FMI is a tool independent specification to support dynamic system simulation
 - A FMU is a ZIP file packaging a model exported in FMI format
- A growing list of tools of supporting FMU export or / and FMU import
- Simulink can import both co-simulation and model-exchange FMUs for both FMI 1.0 and FMI 2.0

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

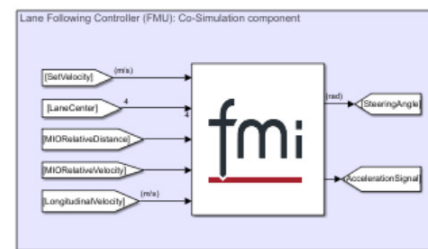
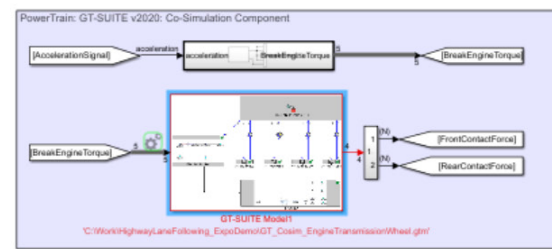
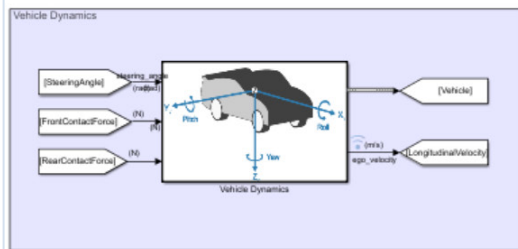
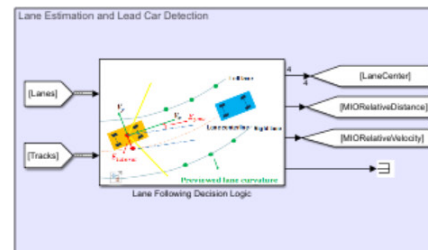
Demo - Virtual Vehicle ADAS Applications

- Integrating external components using S-functions and FMU

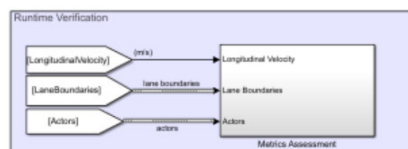
Environment



Perception



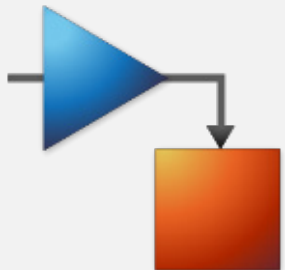
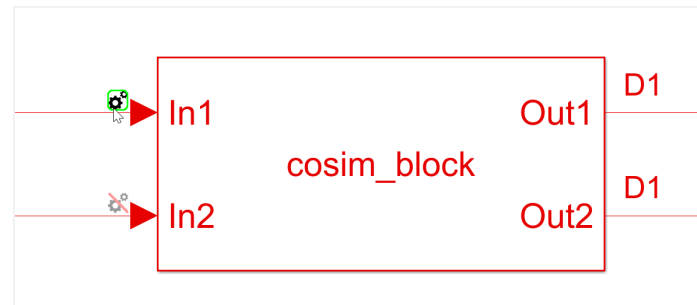
Vehicle



Control

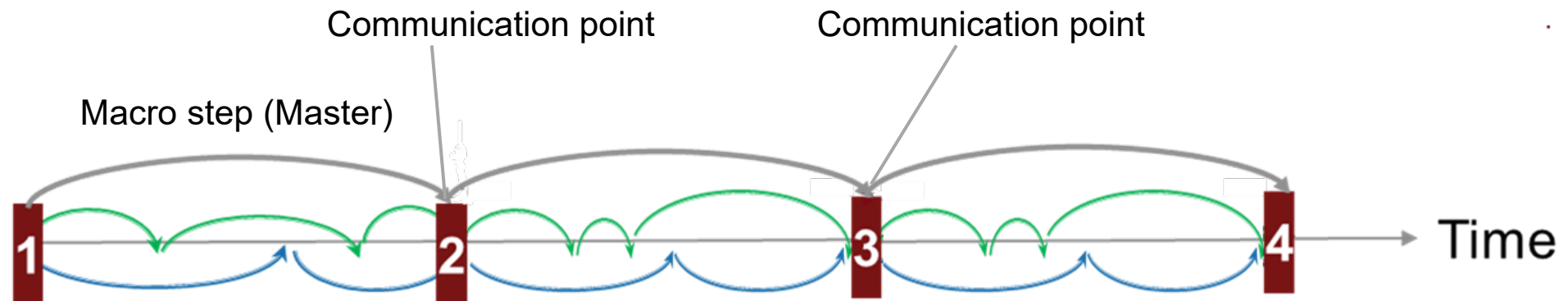
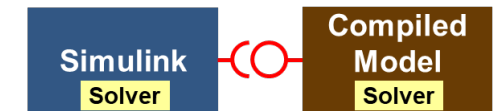
Copyright 2019-2021 The MathWorks, Inc.

Robust Co-simulation



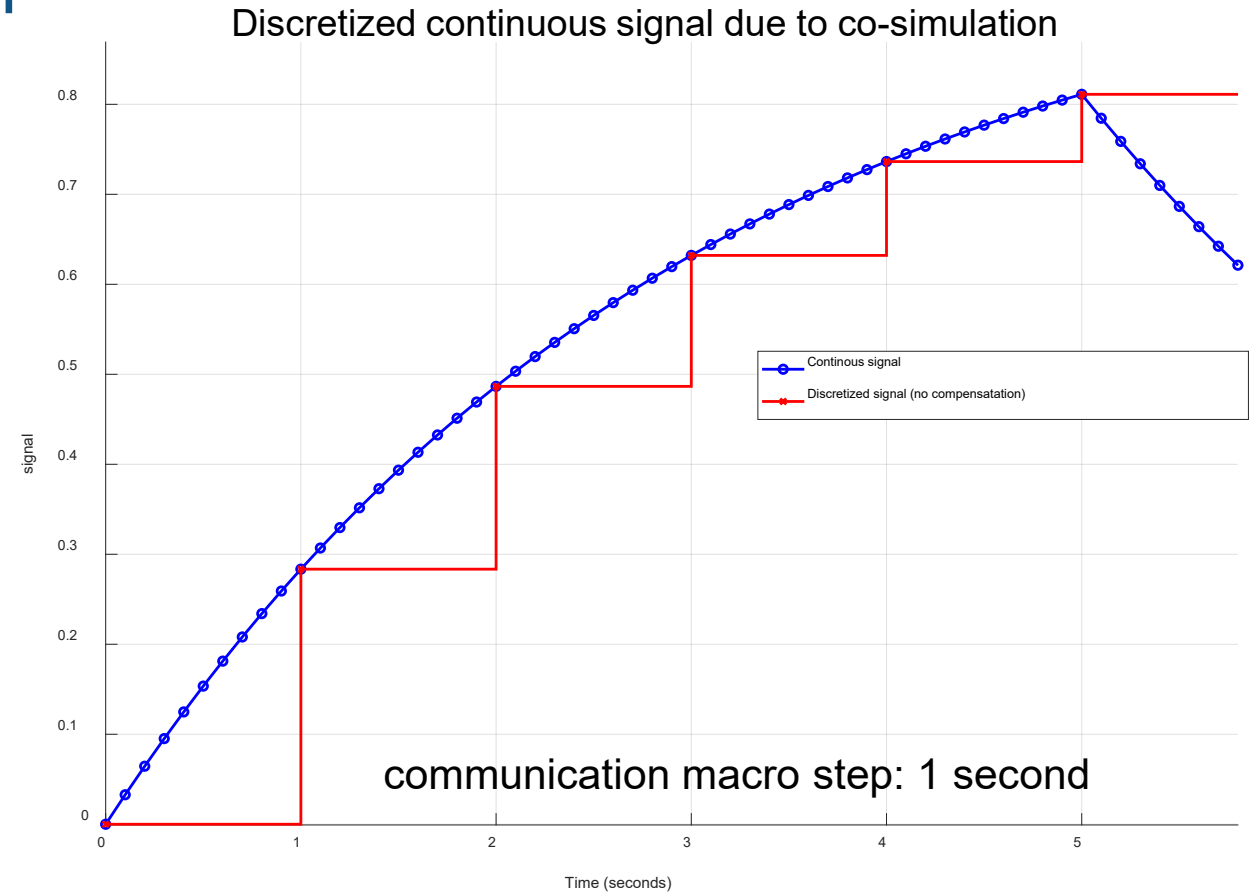
Co-simulation

- A frequently used method to bring models of external tools into Simulink
 - Each co-simulation component has its own solver
 - Can be implemented either white-box or black-box style
- Co-simulation components can run in parallel freely between communication macro steps



Co-simulation numeric behavior

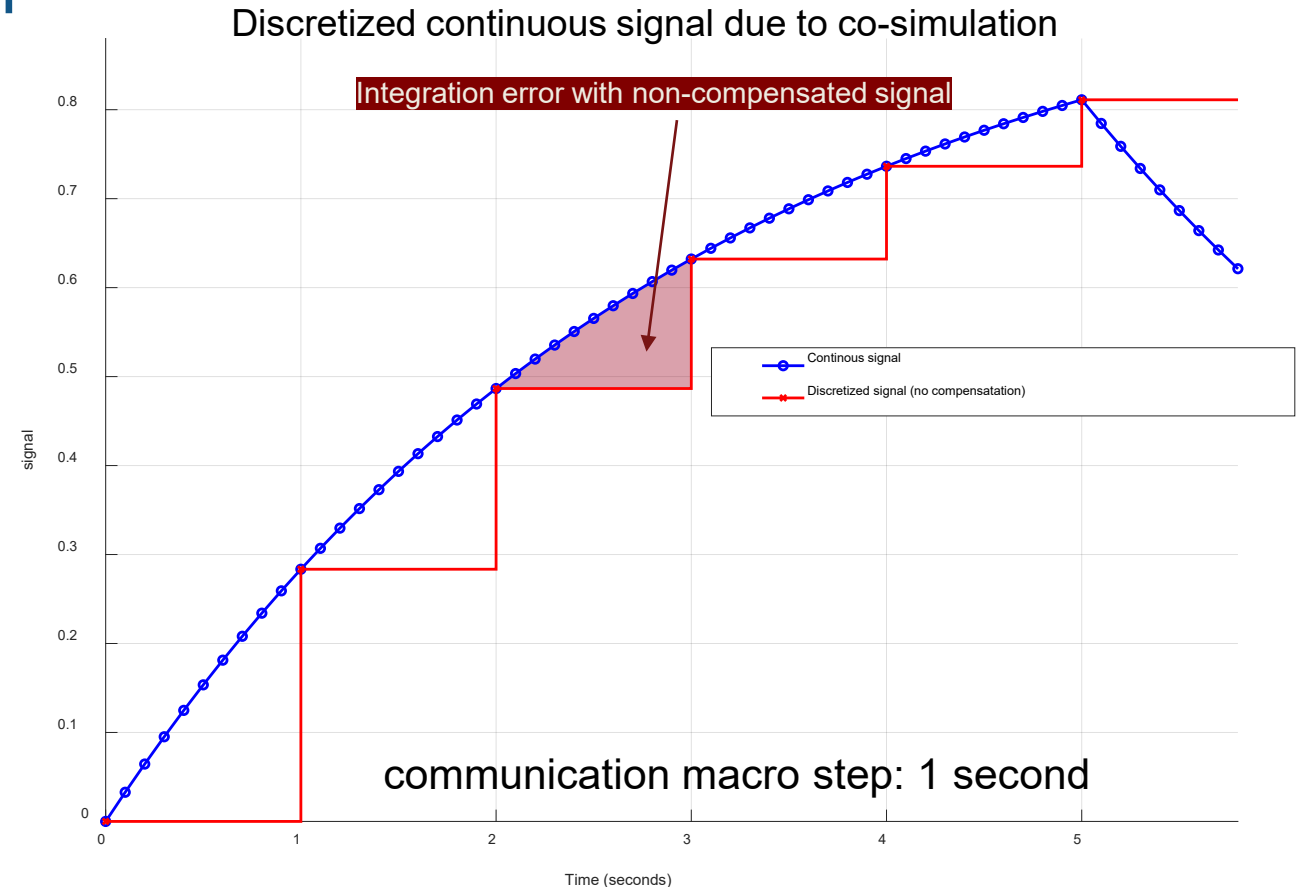
- Model integration is more than coupling the signals
- Potential error when coupling continuous signals
 - Discretized and delayed crossing co-simulation boundary
 - Non-compensated signals could lead to accuracy loss or even system instability



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

Co-simulation numeric behavior

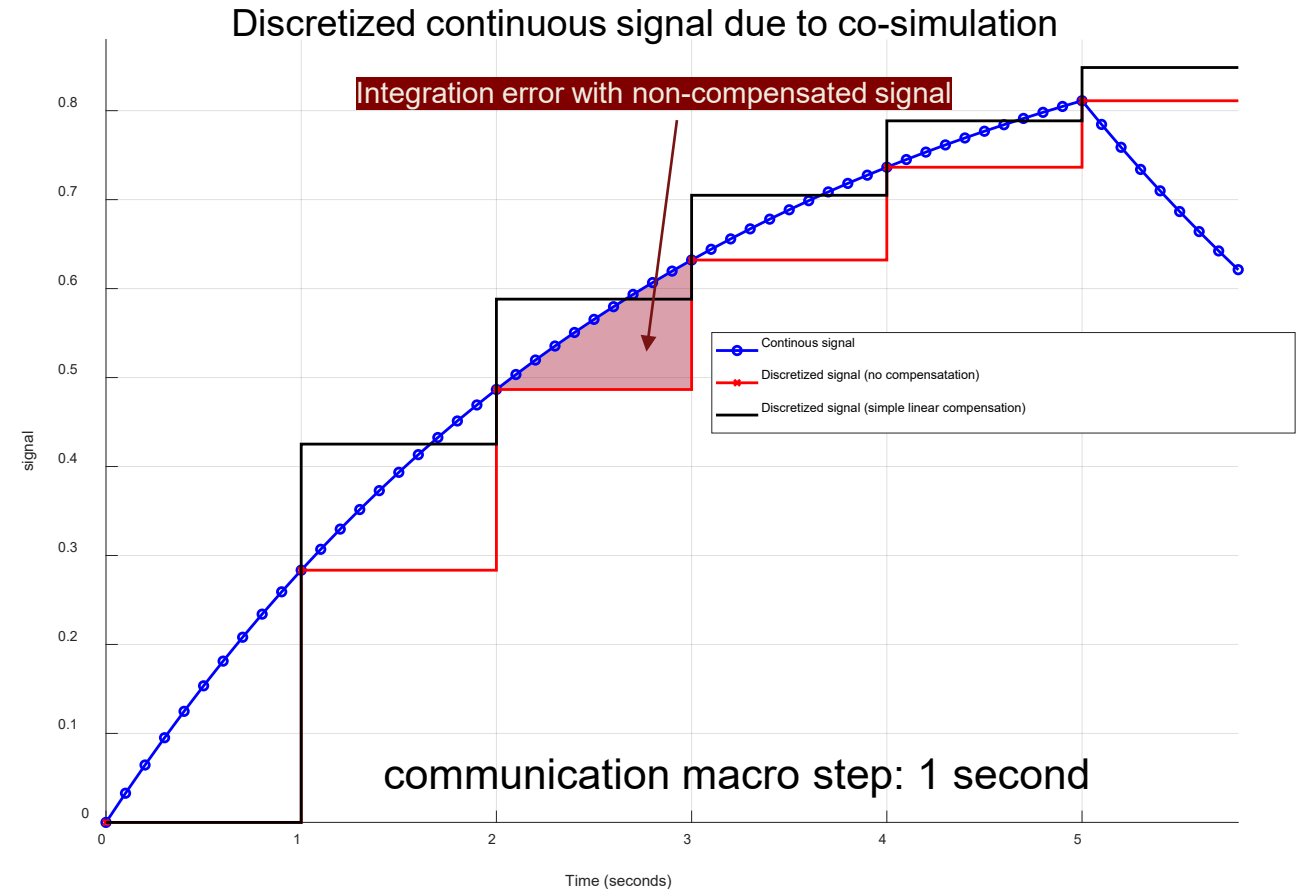
- Model integration is more than coupling the signals
- Potential error when coupling continuous signals
 - Discretized and delayed crossing co-simulation boundary
 - Non-compensated signals could lead to accuracy loss or even system instability



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

Robust co-simulation

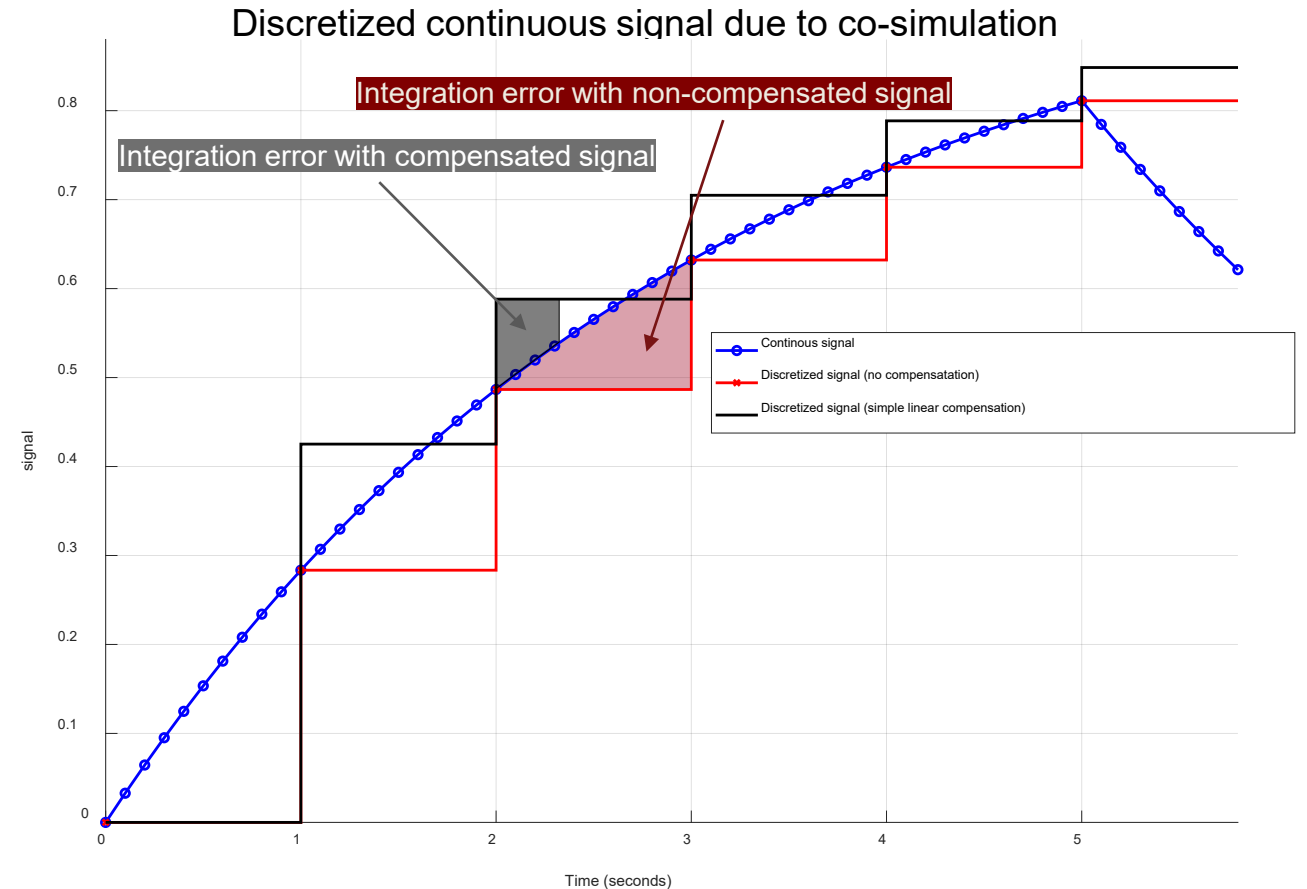
- Automatic and manual mechanism to compensate the discretized continuous signals
 - Choice of linear or high order extrapolation compensation methods
- More robust co-simulation results compared to un-compensated co-simulation



The compensated signal (black line) with simple linear extrapolation is closer to the ideal, continuous signal (blue line) than the uncompensated, discretized signal (red line)

Robust co-simulation

- Automatic and manual mechanism to compensate the discretized continuous signals
 - Choice of linear or high order extrapolation compensation methods
- More robust co-simulation results compared to un-compensated co-simulation



The compensated signal (black line) with simple linear extrapolation is closer to the ideal, continuous signal (blue line) than the uncompensated, discretized signal (red line)

Custom Code Integration

```

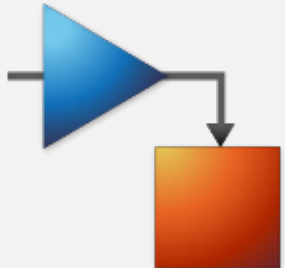
template<class InputString, class OutputString>
bool unhexlify(const InputString& input, OutputString& output) {
    if (input.size() % 2 != 0) {
        return false;
    }
    output.resize(input.size() / 2);
    int j = 0;
    auto unhex = [](char c) -> int {
        return c >= '0' && c <= '9' ?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

```

Internal Libraries

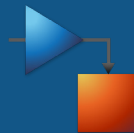
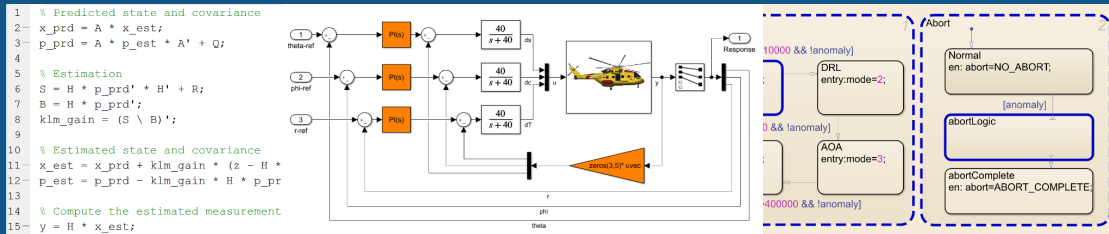


Vendor Libraries



Custom Code Integration

Model-Based Design

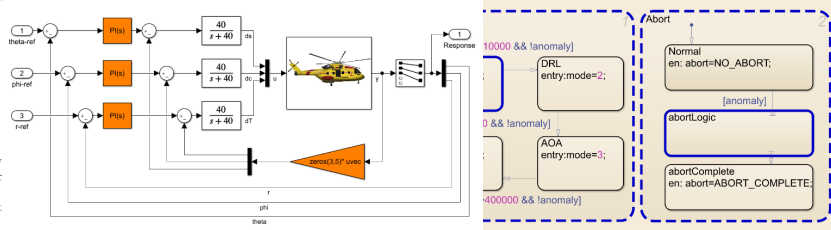


Custom Code Integration

Model-Based Design

```

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 * H * p_pr
13
14 % Compute the estimated measurement
15 y = H * x_est;
    
```



C/C++ Libraries

Hand Code

```

template<class InputString, class OutputString>
bool unhexlify(const InputString& input, OutputString& output) {
    if (input.size() % 2 != 0) {
        return false;
    }
    for (int i = 0; i < input.size(); i += 2) {
        char c1 = input[i], c2 = input[i+1];
        return c1 == '0' || c1 == '1' || c1 == '2' || c1 == '3' ||
            c1 == '4' || c1 == '5' || c1 == '6' || c1 == '7' || c1 == '8' ||
            c1 == '9' || c1 == 'a' || c1 == 'f' || c1 == 'F' || c1 == 'b' ||
            c1 == 'B' || c1 == 'c' || c1 == 'C' || c1 == 'd' || c1 == 'D' ||
            c1 == 'e' || c1 == 'E' || c1 == 'f' || c1 == 'F';
    }
    output += (input[i] << 4) + lowbits;
    return true;
}
    
```

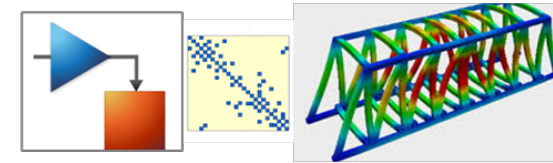
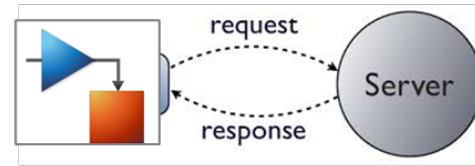
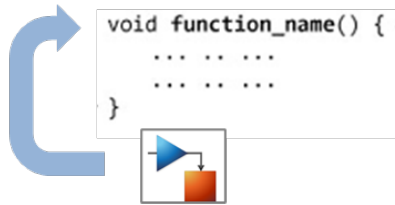
Internal Libraries



Vendor Libraries

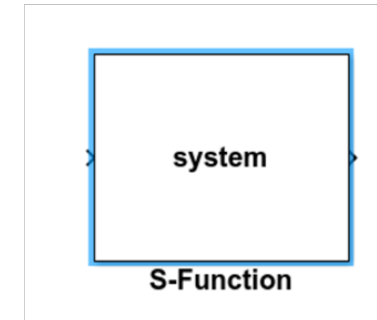
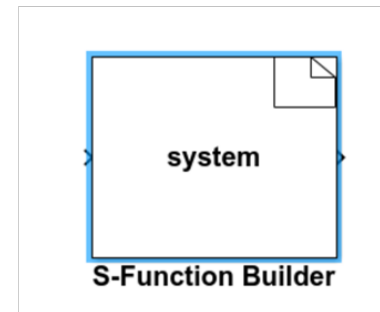
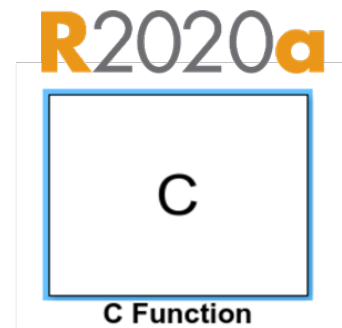


Custom Code Integration



Basic

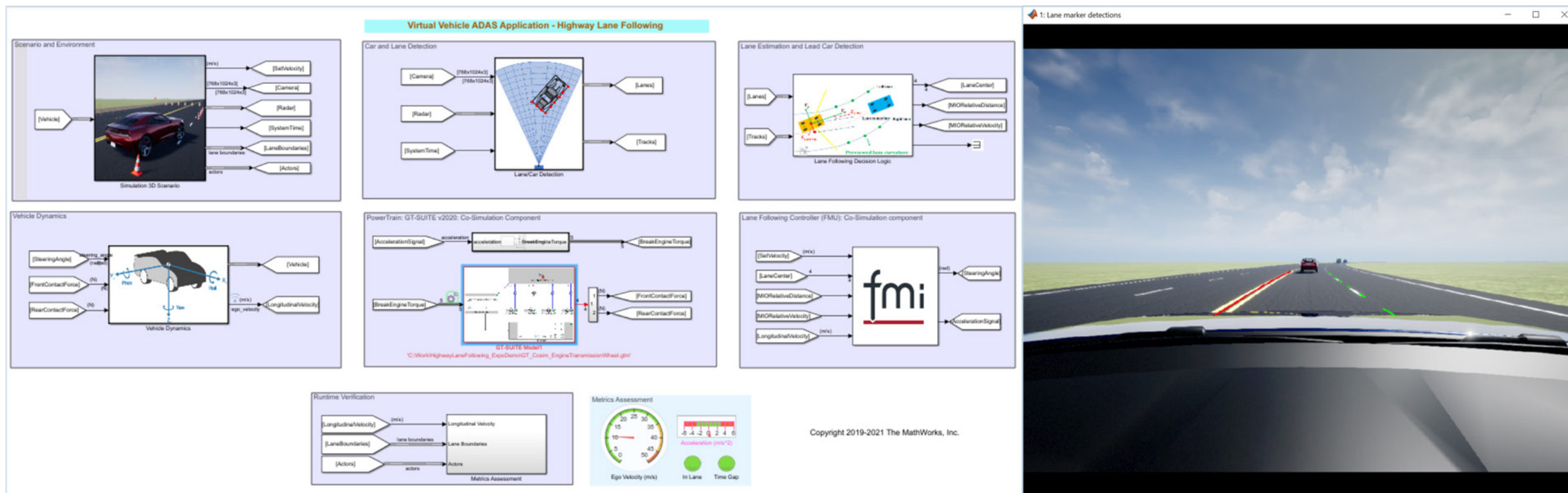
Advanced



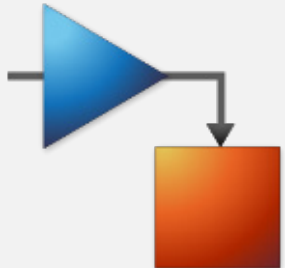
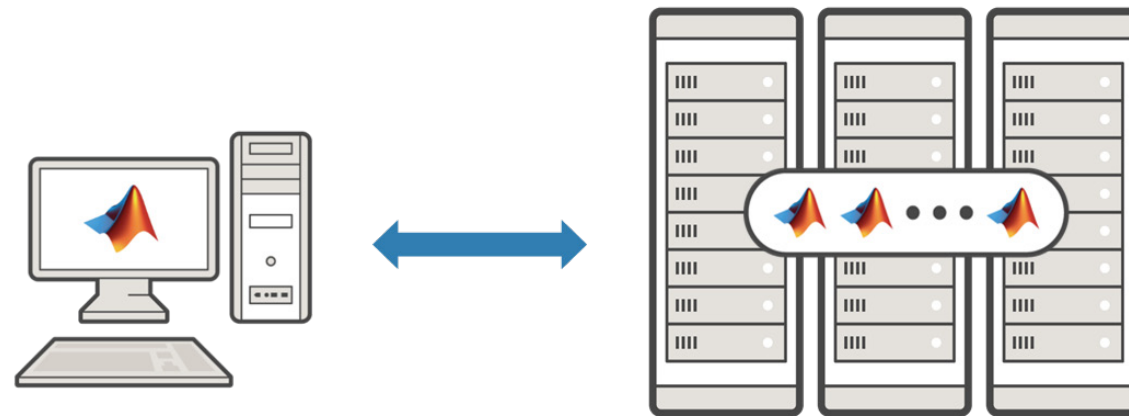
- Versatile ways to integrate your custom code
 - Simply calling your function
 - Reuse your code as a Simulink library
 - Scripting algorithm with discrete states
 - Dynamic system creation

Demo - Virtual Vehicle ADAS Applications

- Integrating custom C code for lane marker detection



Simulation Scalability



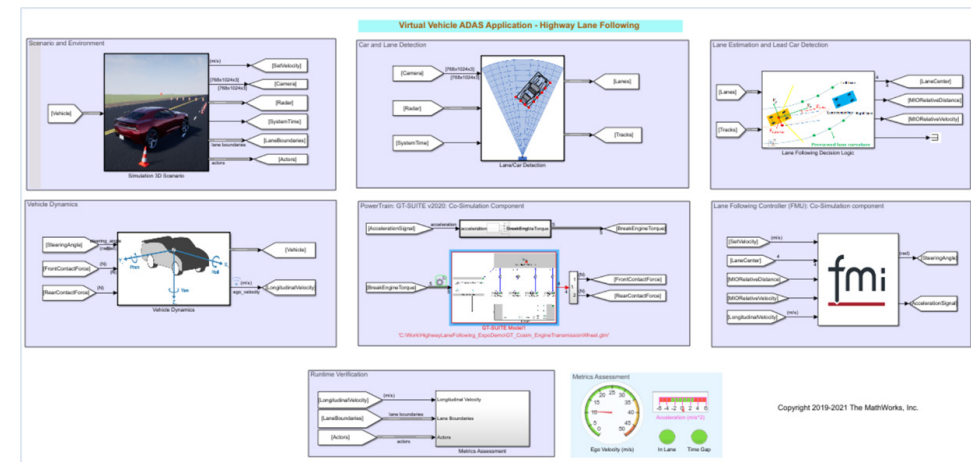
Scale Up System Simulation

- System-level simulation problems may involve a large number of simulation iterations due to the complexity of design combinations
- Complex system simulation takes time to execute
- The capability to scale up is a must-have of an integration platform to deliver quick simulation insights

Full vehicle simulation

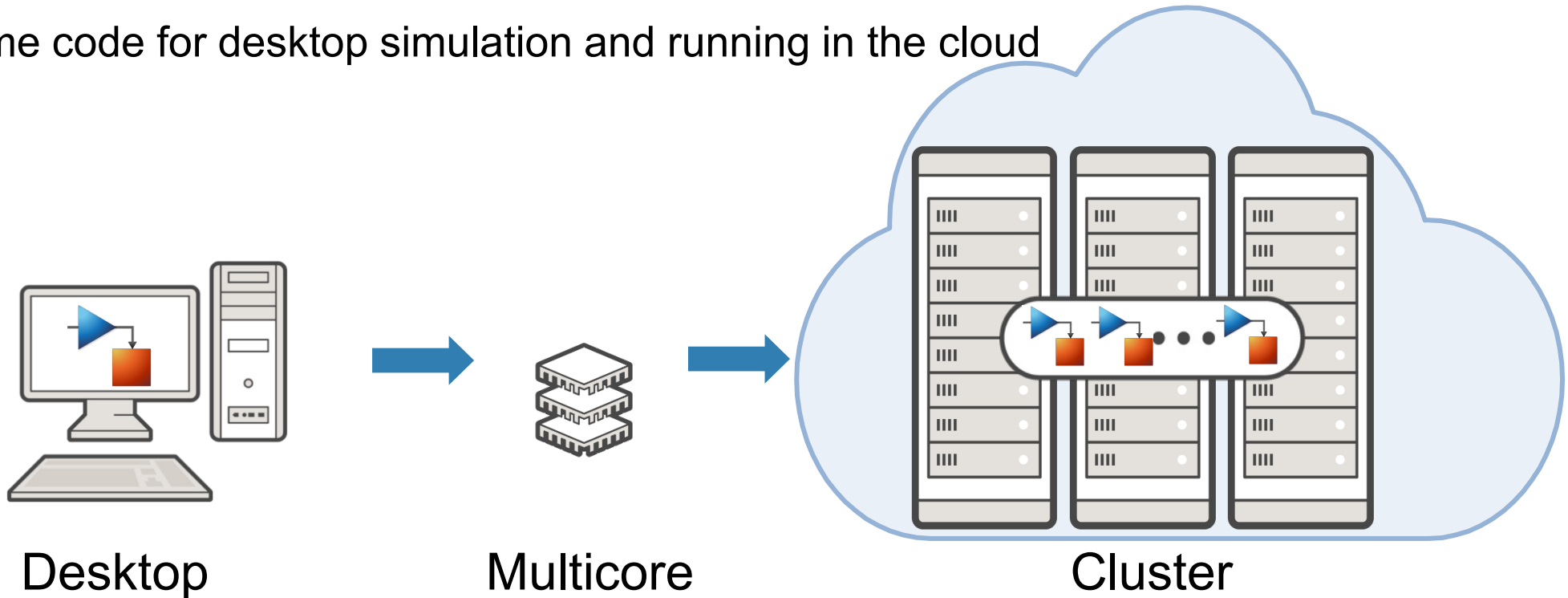
10 drive cycles
10 weather conditions
10 vehicle loadings
10 gear ratios
10 tire sizes

-> 100,000 simulations



Scale Up System Simulations

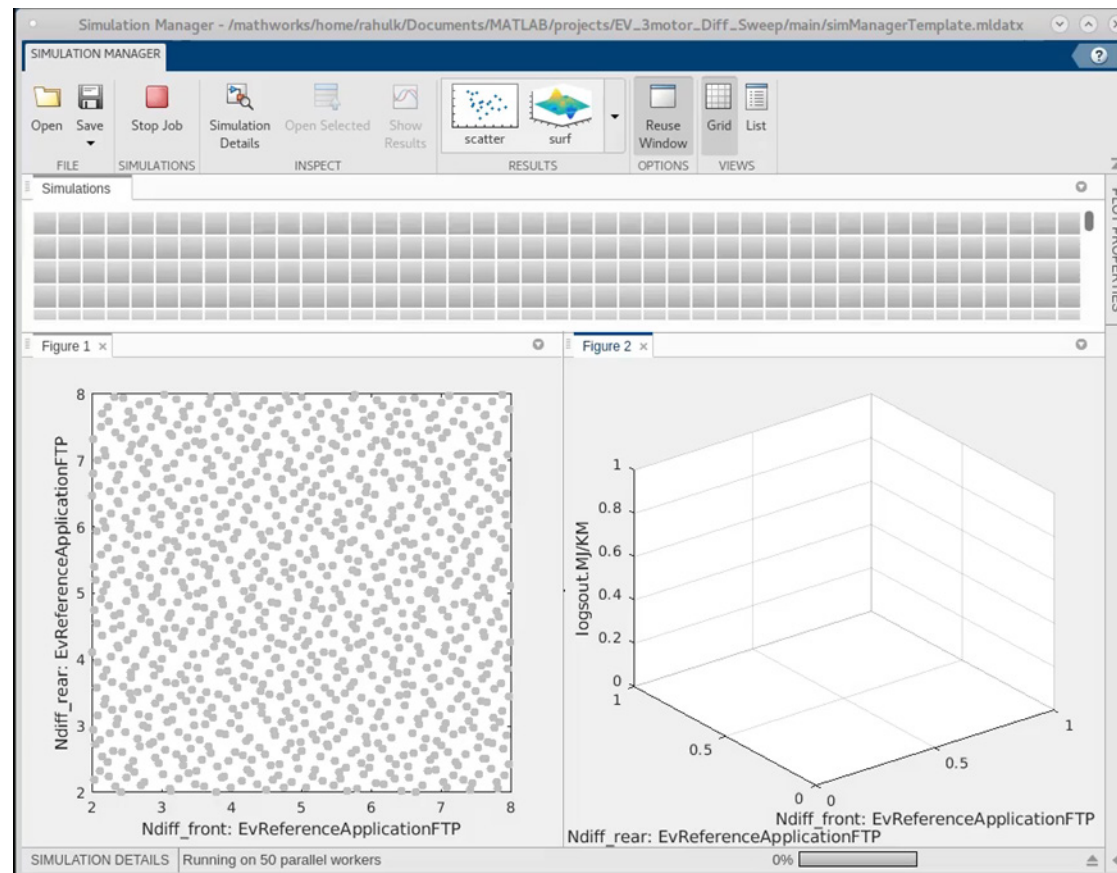
- The same code for desktop simulation and running in the cloud



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

Scale Up System Simulations

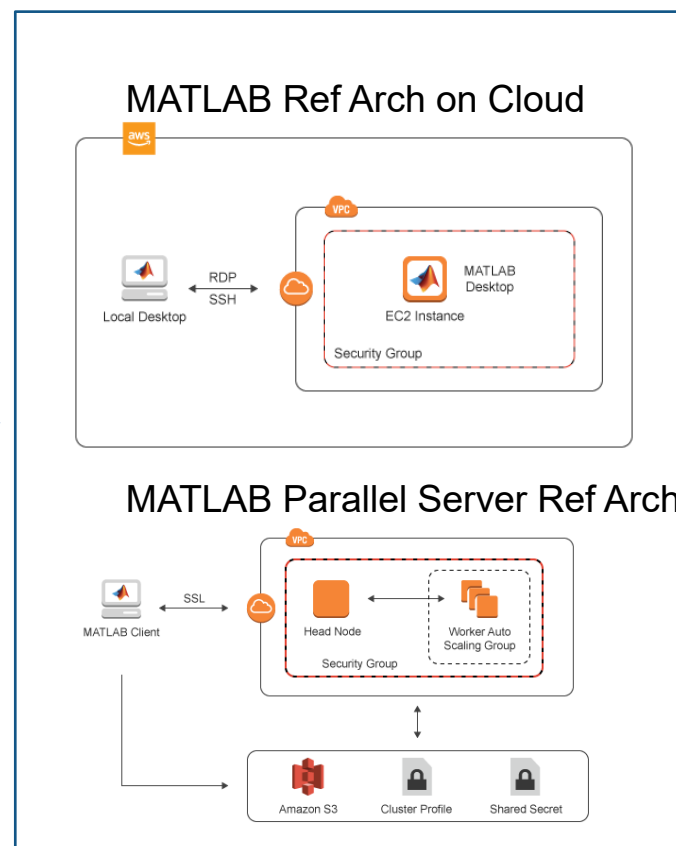
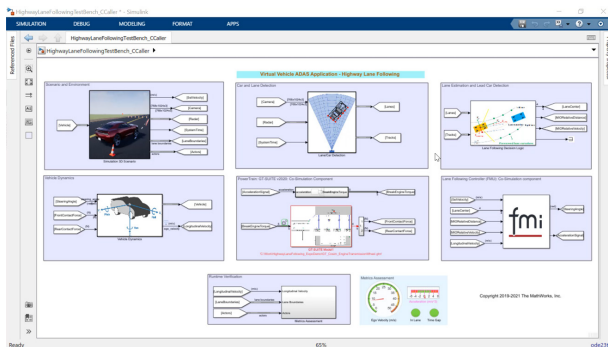
- Manage and visualize the simulations as the simulations are progressing



Scale Up System Simulations

- Move the simulation to the Cloud by leveraging a Prebuilt Cloud Configuration via Reference Architecture

System Simulation



[Use MATLAB/Simulink in the Cloud](#)

Summary

As an integration platform Simulink provides key capabilities to scale up your complex, system-level simulations:

- Standard-based interfaces to integrate 3rd party simulation models
- Co-simulation numeric robustness with automatic signal compensation
- Bringing in custom C/C++ code made easy
- Utilizing parallel simulation capabilities to speed up system level simulations



MATLAB EXPO 2021

Thank you



© 2021 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.