Simulink for Virtual Vehicle Development

Chris Fillyaw
Application Engineering Manager
cfillyaw@mathworks.com

Mike Sasena, PhD
Automotive Product Manager
msasena@mathworks.com
Key takeaways

MathWorks provides a **powerful platform** for building your **Virtual Vehicle**

Our platform is very **flexible**, and we can help you **customize** it for your needs

Out-of-the-box capability

Custom virtual vehicle solution
Virtual vehicle: functional simulation of full vehicle behaviors

Reduced years of effort and expensive prototypes

Tesla: vehicle design tradeoff
Ricardo: simulating passenger comfort
Ford: software validation

Reduce physical testing needed before design validation
Embedded software is essential for many virtual vehicle applications

**Virtual vehicle applications** such as attribute development, software validation, calibration require simulation of embedded software.

- Application software behavior fully represented
- Interfaces consistent with software component definitions
- Basic software components included as need for the application
Example: Validating lane following software functional safety requirement (FSR)

FSR: The lane following system lateral error shall be less than 1 meter

Questions to consider:
- System performance under normal conditions?
- Impact of environment conditions?
- Impact of a component failure?
- Required processor throughput?
System level interactions need to be considered

FSR: The lane following system lateral error shall be less than 1 meter
FSR: The lane following system lateral error shall be less than 1 meter.

Discovering problems during system-level integration is expensive.
Validate software against function safety requirements early

FSR: The lane following system lateral error shall be less than 1 meter

Use simulation to do system-level integration testing early

Virtual vehicle

- Sensors
- Controllers
- Powertrain
- Environment
- Driver
- Vehicle

Define Requirements

System-Level Specification

Subsystem Design

Subsystem Integration & Test

Subsystem Implementation

Complete Integration & Test
Agenda

- Common challenges
- MathWorks solutions
- Case study
Agenda

- Common challenges
- MathWorks solutions
- Case study
Using a virtual vehicle for systems integration testing early in development can **save time / money**.

What are the **challenges** to building one?
Challenges to early system-level testing

- Availability of appropriate vehicle level model
- Access to plant and sensor models with “right” level of fidelity
- Model calibration
Challenges to early system-level testing

- Standardizing interfaces and data management
- Access to software components across different teams
- Assembly of software components from multiple sources
Challenges to early system-level testing

- Creation of virtual 3D environment
- Definition of scenarios to test
- Linking test cases to requirements
Challenges to early system-level testing

- Post-processing and visualizing results
- Automatically generating reports
- Running large numbers of simulations efficiently
Challenges to early system-level testing

- Sharing models across the organization
- Deploying models to users who aren’t tool experts
- Deploying models for SIL, HIL, etc.
Agenda

- Common challenges
- **MathWorks solutions**
- Case study
MathWorks Virtual Vehicle: reference applications

- Start with in-house vehicle models
  - We can help you customize it and apply best practices for Model-Based Design

- Start with our reference applications
  - Detailed system and vehicle level models for powertrain, vehicle dynamics, ADAS and other applications

Learn more:
- Powertrain Blockset
- Vehicle Dynamics Blockset
- Automated Driving Toolbox
Add detail where needed using:
- In-house Simulink models
- Simulink and Simscape libraries
- 3rd party tools (S-function, FMU, …)

Learn more:
- Simscape
- Multi-core cosim
- Integrate with existing sims
Integrate controller algorithms:

- Native Simulink models
- 3rd party tools (S-function, FMU, …)
- C / C++ code

Learn more:
- C / C++ code integration
- C Caller block
MathWorks Virtual Vehicle: complex project management

Use MathWorks platform to:
- Collaborate across teams
- Reference related project files
- Manage version control

Learn more: MATLAB Projects
Use Driving Scenario Designer to:
- Create roads and lane markings
- Add actors and trajectories
- Specify actor size and radar cross-section (RCS)
- Explore pre-built scenarios
- Import OpenDRIVE and HERE HD Live Map roads
- Export MATLAB code
- Export Simulink model

Learn more:
Automated Driving Toolbox
MathWorks Virtual Vehicle: automotive scene creation

Use RoadRunner to:

- Design 3D scenes for AD simulation
- Customize with region-specific road signs and markings
- Configure traffic signal timing
- Import from OpenDRIVE
- Export to OpenDRIVE, FBX, …
- Use scenes in Unreal, Unity, CARLA, …

Learn more: RoadRunner
MathWorks Virtual Vehicle: requirements definition

Use V&V tools to:
- Define sequence of simulations to run
- Define requirements for these tests
- Define custom report template

Learn more:
Verification & Validation
MathWorks Virtual Vehicle: results analysis

Use post-processing tools to:

- Review results with flexible MATLAB platform and visualization tools
- Interact with user-friendly Live Scripts
- Automate report generation

Learn more:
MATLAB Live Editor
Simulink Report Generator
MathWorks Virtual Vehicle: scalability

Use MATLAB and Simulink to:
- Distribute simulations to local multicore, GPU, clusters, or the cloud
- Scale up computation power as needed without needing to rewrite code

Learn more:
Parallel Computing Toolbox
MATLAB Parallel Server

Perform large-scale computations using multicore desktops, GPUs, clusters, grids, and clouds
MathWorks Virtual Vehicle: model deployment

Use MATLAB and Simulink to take applications farther:

- Create custom UI’s
- Create installers for distribution
- Deploy models as executables, FMU’s or web apps
- Generate code for SIL, HIL testing

Learn more:
MATLAB Web App Server
MATLAB App Designer
Simulink Compiler
Embedded Systems
MathWorks Consulting Services can support you

- Model Architecture
  - Model assessment
  - Simulation performance
  - Interface standardization
  - …

- Construction
  - Build process automation
  - Database/Repo interface
  - Model-Building know-how
  - …

- User Experience
  - GUI driven workflow
  - Tool compatibility support
  - Artifact creation
  - …

- Provide expert-level guidance
- Automate workflows
- Develop custom UI’s
Agenda

- Common challenges
- MathWorks solutions
- Case study
Validate software against function safety requirements early

FSR: The lane following system lateral error shall be less than 1 meter

Use simulation to do system-level integration testing early

Learn more:
Highway Lane Following
Automate Testing for Highway Lane Following
Case study: highway lane following algorithm

- Create Unreal Engine scene
- Specify target trajectories
- Model camera and radar sensors
- Model ego vehicle dynamics
- Specify system metrics

Learn more: 
Highway Lane Following
Case study: highway lane following algorithm

- Author and associate requirements and scenarios

Learn more:
Automate Testing for Highway Lane Following
Case study: highway lane following algorithm

- Visualize system behavior with Unreal Engine
- Visualize lane detections
- Visualize vehicle detections
- Visualize control signals
- Log simulation data

Learn more: Highway Lane Following
Automate test execution and reporting
Execute simulations in parallel

Learn more:
Automate Testing for Highway Lane Following
Case study: highway lane following algorithm

- Assess system metrics
- Assess lane detection metrics

Learn more:
Automate Testing for Highway Lane Following
Case study: highway lane following algorithm

- Generate algorithm code
- Test with Software-in-the-Loop (SIL) simulation
- Workflow could be extended to test hand coded algorithms

Learn more:
Automate Testing for Highway Lane Following
Summary

1. Started with reference application, then customized
2. Integrated software
3. Defined scenarios to test
4. Simulated model and analyzed results
5. Deployed model
Key takeaways

MathWorks provides a powerful platform for building your Virtual Vehicle.

Our platform is very flexible, and we can help you customize it for your needs.

Out-of-the-box capability

Custom virtual vehicle solution
Please contact us with questions

Chris Fillyaw  
Application Engineering Manager  
cfillyaw@mathworks.com

Mike Sasena, PhD  
Automotive Product Manager  
msasena@mathworks.com

- On a scale of 1 - 4, how challenging is it for your department to:
  - Create the vehicle model
  - Integrate software
  - Author scenarios
  - Simulate and analyze results
  - Deploy simulations

- Are you interested in a follow-up conversation with MathWorks?

- Additional comments
Thank You

Chris Fillyaw
Application Engineering Manager
cfillyaw@mathworks.com

Mike Sasena, PhD
Automotive Product Manager
msasena@mathworks.com