Scenario-based Modeling and Simulation

Yuming Niu, Mark Malone, Kyle Post, Amit Karjol

Ford Motor Company

Agenda

- New challenges in AV/ADAS development
  - Exponentially growing software complexity driven by advanced control features
  - Paradigm shift in vehicle Verification & Validation environment
- Some core capabilities needed to address those challenges
  - Model based system engineering is the rescue of system complexity
  - Fully digital tool chain
  - Scenario modeling
  - Large scale cloud computing
- Case study – Scenario simulation through Simulink/CARLA co-simulation
- Future work

Growing software complexity driven by safety features

"A modern car contains around 100 million lines of software code. To become fully autonomous, an estimated 10X increase is required — that's a staggering 1 billion lines of code. To put that in perspective, one of the world's most advanced fighter aircraft, the F-35, contains around 10 to 20 million lines of code." – sourced from Ansys

1950 – 2000
Safety/Convenience Features

Cruise Control
Seat Belts
Antilock Brakes

2000 – 2010
Advanced Safety Features

Stability Control
Blind Spot Detection
Forward Collision Warning
Lane Departure Warning

2010 – 2016
ADAS Features

Rearview Video Systems
Automatic Emergency Braking
Rear Cross Traffic Alert
Lane Centering Assist

2016 – 2025
Partially Automated Safety Features

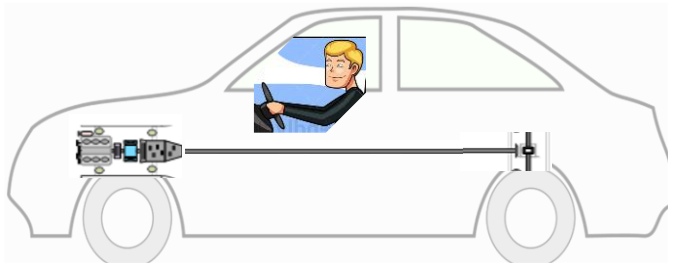
Lane Keeping Assist
Adaptive Cruise Control
Traffic Jam Assist

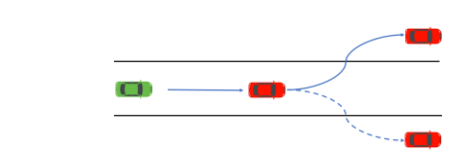
2025+
Fully Automated Safety Features

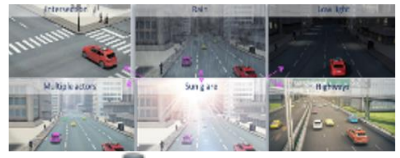
https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety#the-topic-safety-timeline

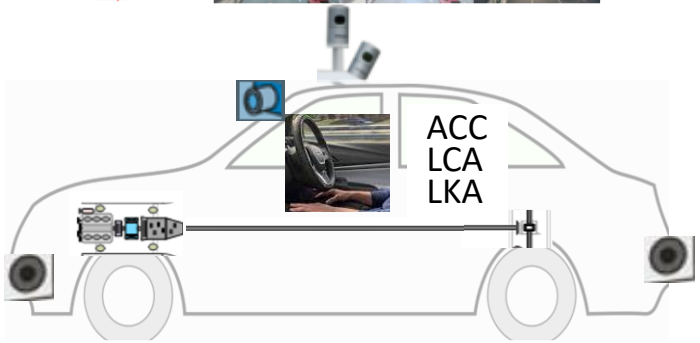
Ford Go Further

Paradigm shift from traditional vehicle V&V to AV/ADAS V&V







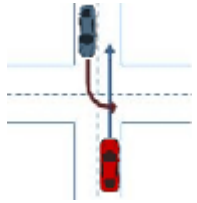


- For traditional vehicle, all driving decisions are made by human driver
- Vehicle development focuses on vehicle itself , such as powertrain, chassis, etc.
- Interaction with other vehicles are not considered

- For ADAS/AV, part of or all driving decisions are made by computer
- Interaction with other vehicles and environment are critical
- Vehicle development extends from vehicle boundary to include interaction with all elements in operational design domain

AV/ADAS Vehicle development crosses traditional vehicle body boundary, which introduces infinite variations into V&V

Complexities coming from all ways



Infinite scenarios

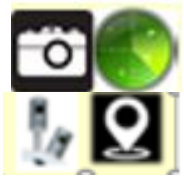
New hardware

Much more software

Requirements Management

Disruptive technologies

Operational design domain

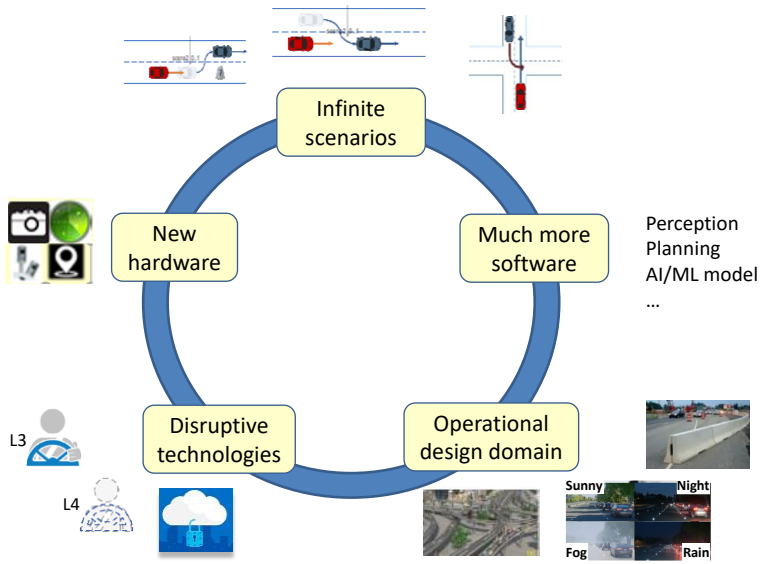


Perception
Planning
Intent prediction
AI/ML model
…

L3

L4

Challenges to vehicle development



- *"1 billion lines of code needed"* – sourced from Ansys
- *"275m miles testing are required for self-driving car"* – *source 1
- *"1.95m mile tested from Dec 2019 to Nov 2020"* -- *source 2
- Perception module and control algorithms are required to test in infinite number of scenarios
- Requirements management
- …

Requirements management complexity, scenario complexity, and number of miles required for V&V present a real challenge for product development; however, can physical testing be increased to the order of magnitude to meet the exponentially growing V&V needs?

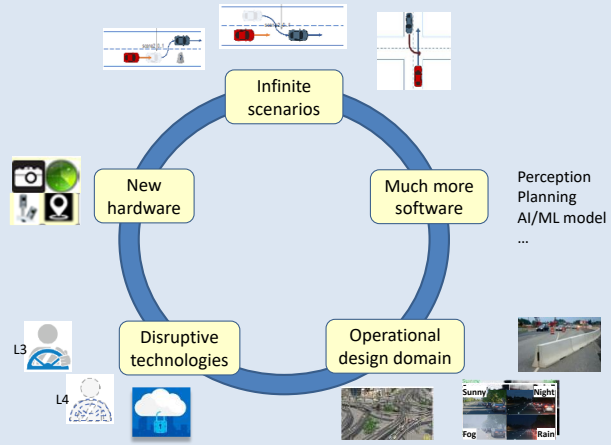
1. https://www.statista.com/chart/7009/self-driving-cars-are-on-their-way/
2. https://www.statista.com/chart/17144/test-miles-and-reportable-miles-per-disengagement/

Ford Go Further

Problem Statement

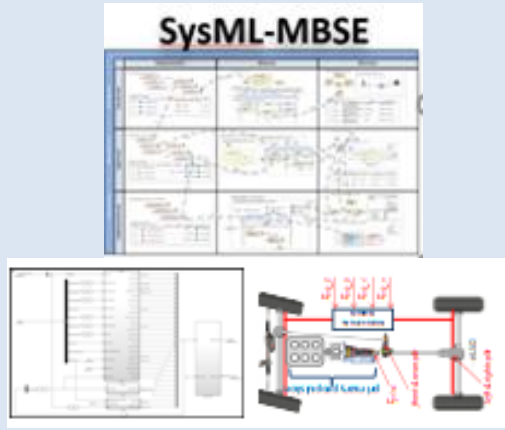
Challenges

- Rapid technology advancement and exponentially growing complexity.
- Complex interface and interactions.
- Late change is costly and causing program delay.
- Robustness verification requires extensive testing.
- Test is critical but cannot hope to increase testing by the order of magnitude needed for growing complexity.



Opportunity

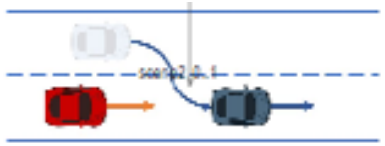
- System engineering helps to identify and solve problem at early design phase
- The use of simulation can address our inability to test everything
- Deep integration beyond hardware level by including driver/vehicle/environment interaction is the way to deliver safe and robust products



Fully digital toolchain

Scenario identification

- Data mining
- Scenario extraction
- Scenario statistics
- Scenario database



Identified Scenario

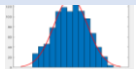
Scenario Contents

Road network

Traffic

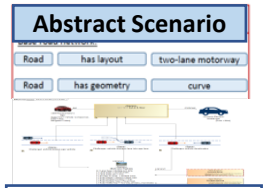
Behavior

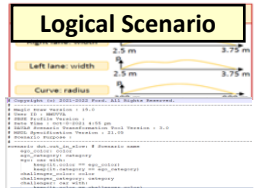
Statistics



Scenario-based SE

- Abstract/Logical scenario
- Requirements
- Open standards





Scenario Description

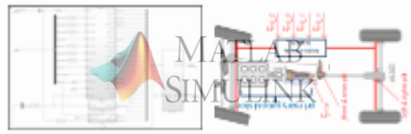
Open standards

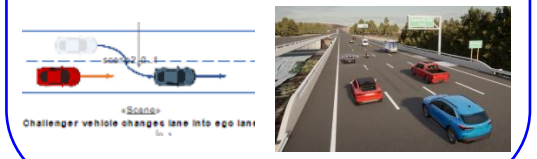
MagicDraw plugin

Concrete Scenario Model

- Low fidelity for algorithm development
- High fidelity for perception module development

Scenario Integration with Vehicle Model



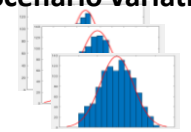


V&V

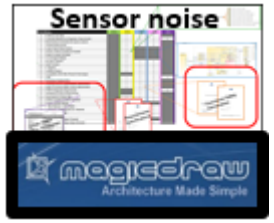
Algorithm Development

Software release

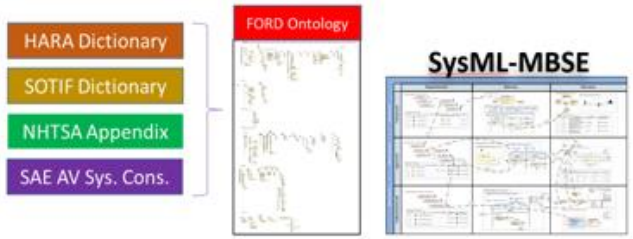
Scenario variation



Sensor noise



Feature Requirements



AV/ADAS simulation tools ecosystem

Vehicle dynamics and Scenario simulator

Open standards

- OpenLABEL
  - Object and scenario labelling format
- OpenODD
  - ODD description format
- OpenXOntology
  - Extendable domain ontology for OpenX
- OpenSCENARIO
  - Dynamic scene description
- OpenDRIVE
  - Static road network
- OpenCRG
  - Detailed Road surface description
- Open Simulation Interface
  - Generic interface between sensor models & ground truth

Source: https://wiki.eclipse.org/images/e/ee/2020-08-18_OpenADX-SC_ASAM_Simulation_Overview.pdf

Case study – Scenario simulation with Simulink/CARLA co-simulation enabled by MATLAB/Python Integration

Toolchain introduction

- CARLA is an open-source simulator for autonomous driving and ADAS research, which can simulate different environmental conditions, static/dynamic actors, and maps, *etc*.
- Many prototype controls are developed in Simulink, which need to be verified and validated with different scenarios
- There is a need to run co-simulation between Simulink and CARLA
- MATLAB provides the fundamental integration mechanism that allows co-simulation with CARLA through python API

Challenges

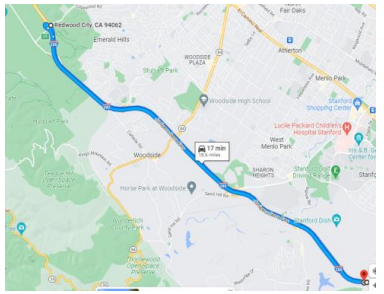
- There is no existing built-in Simulink/CARLA interface block
- Synchronization between Simulink and CARLA since each has its own simulation clock
- How to create road maps for simulation?

Control CMD

How to communicate?
How to synchronize?

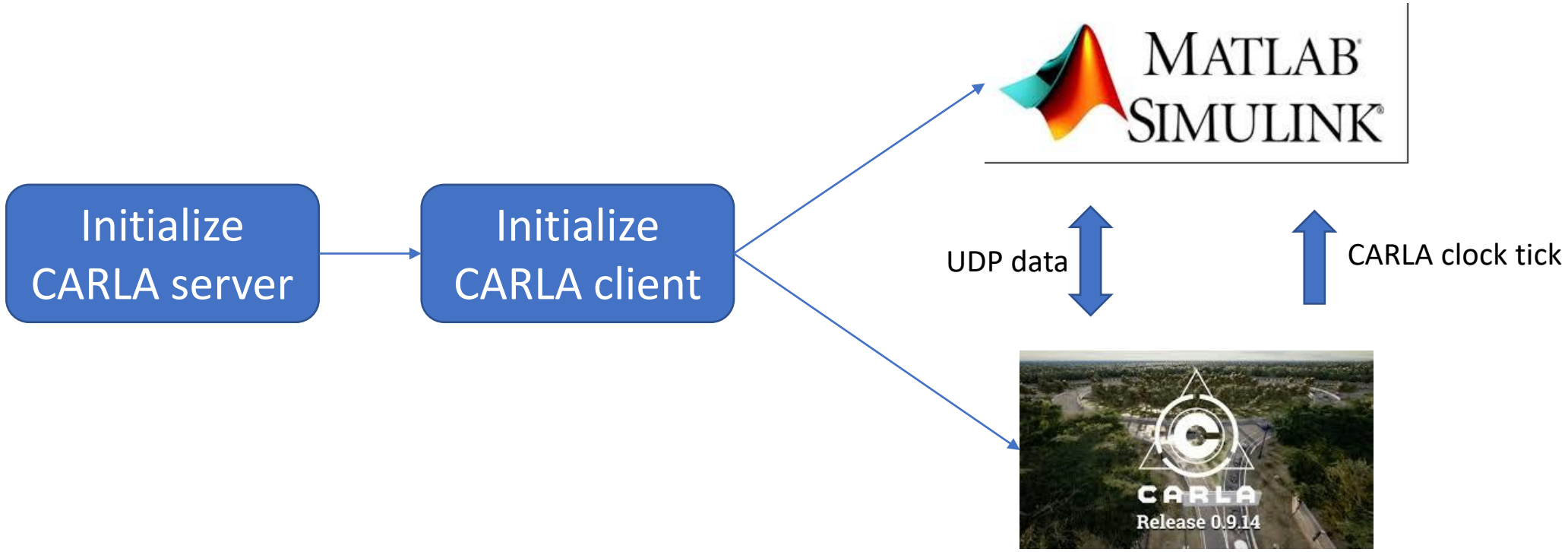
Vehicle states

How to plug in real world map?

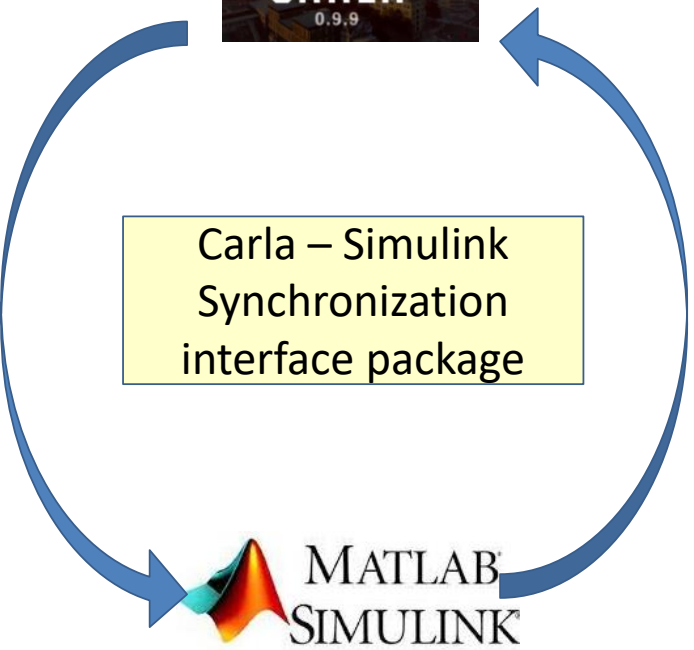


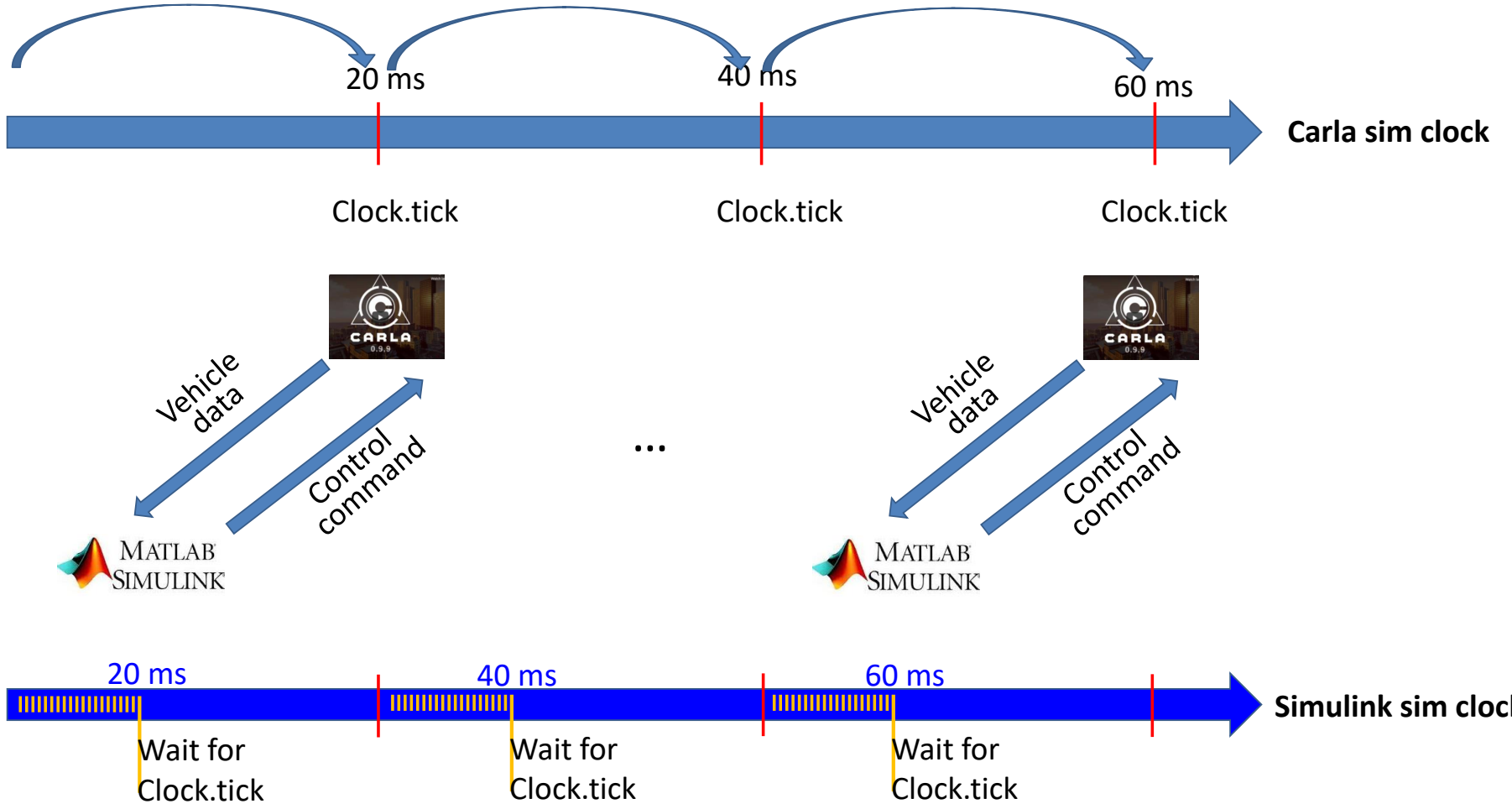
MATLAB/CARLA co-simulation interface package

- A co-simulation interface package is developed with MATLAB System block in Simulink
- UDP is used for data communication



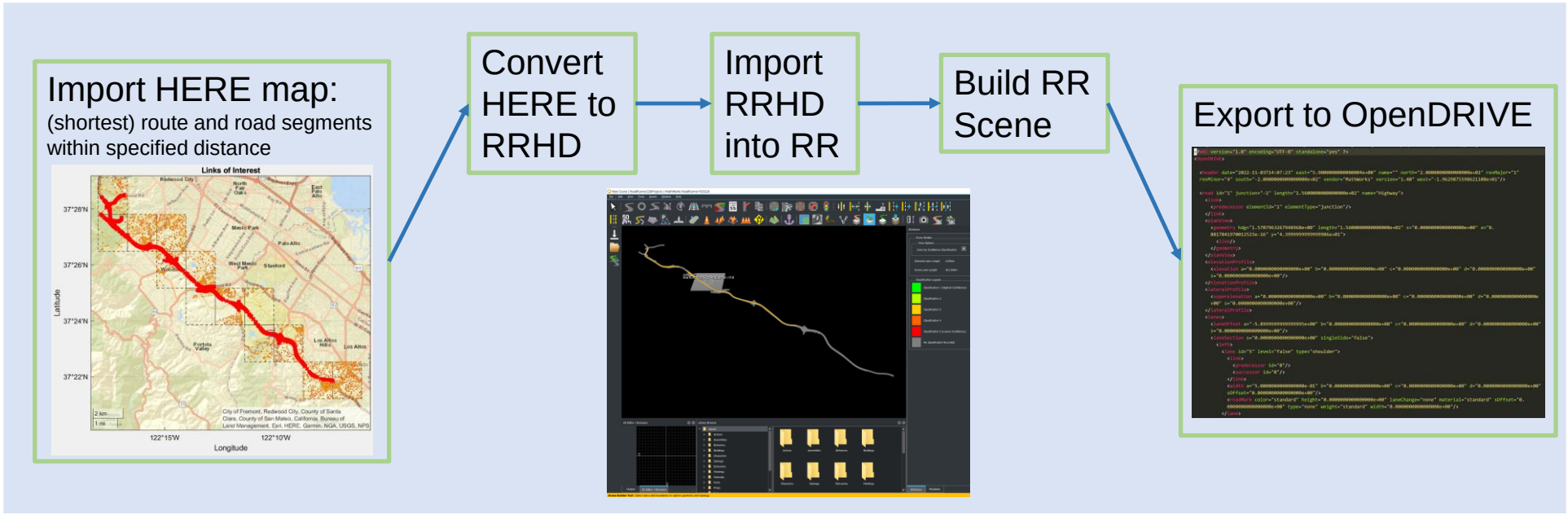
SIMULINK/CARLA Synchronization

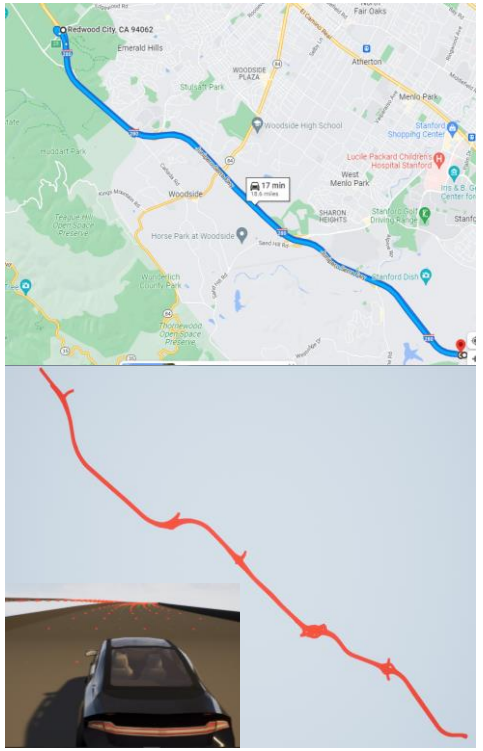




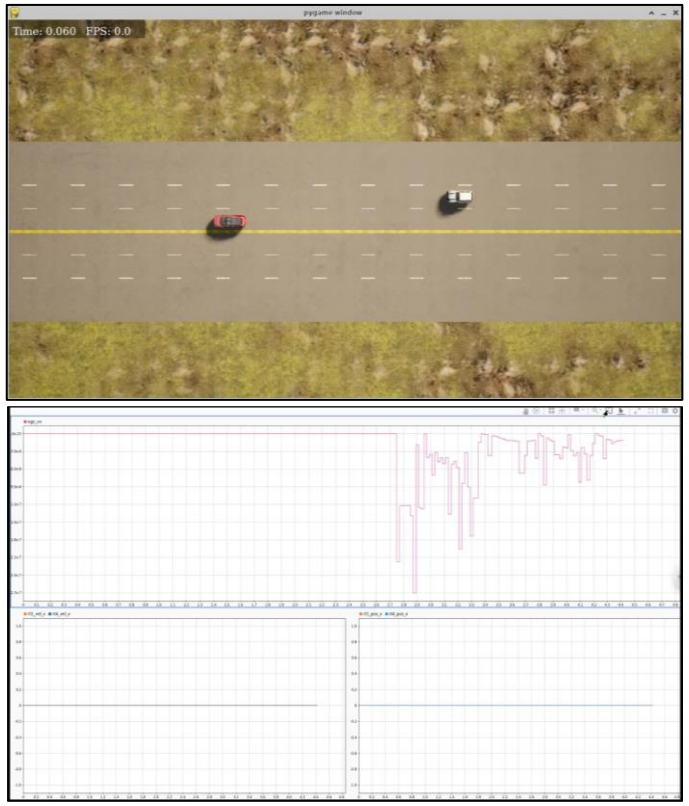
Map creation with Road Runner

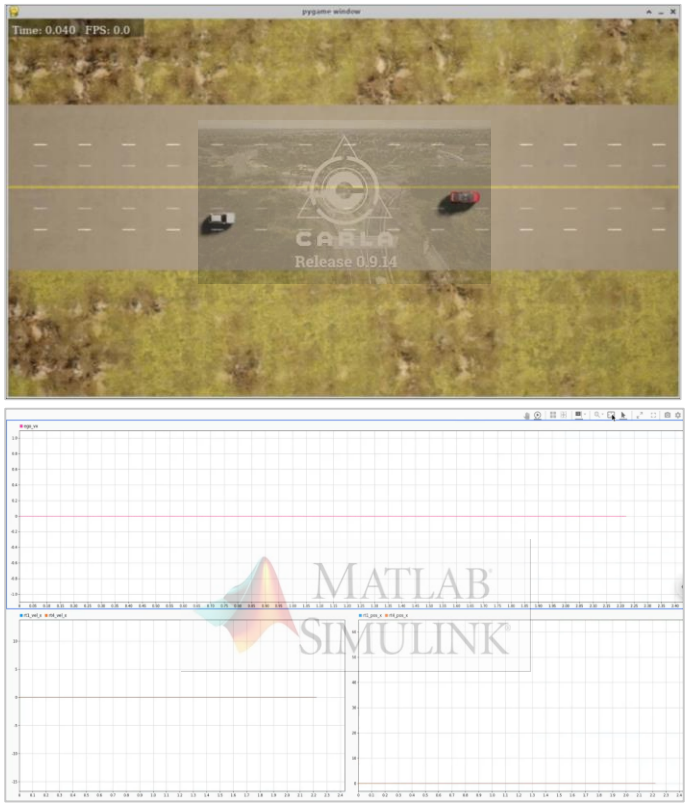
- It is required to represent the real-world map accurately in simulation
- Both manual and automated process have been developed to import HERE map into simulation
- Significantly reduced map size for simulation through auto trimming

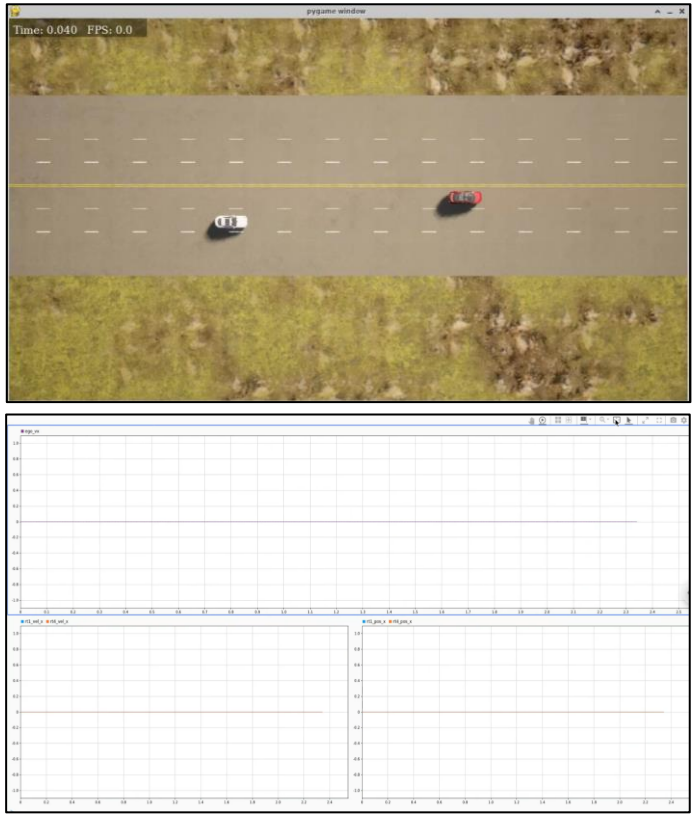




Scenario examples





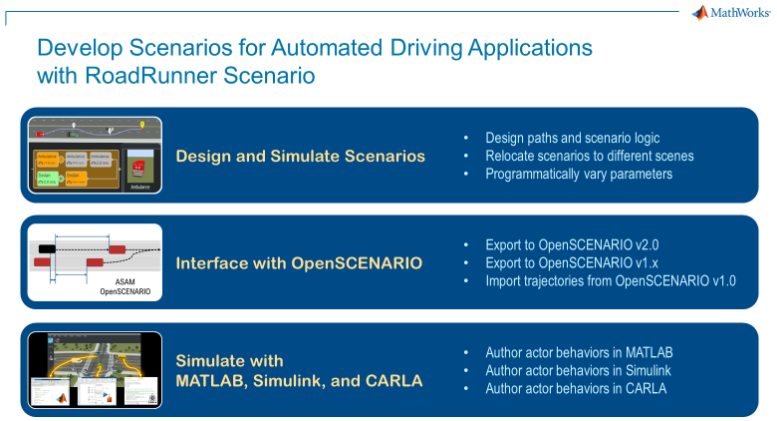


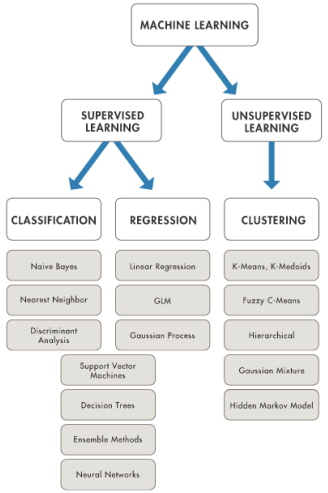
Scenario Parameters Example:

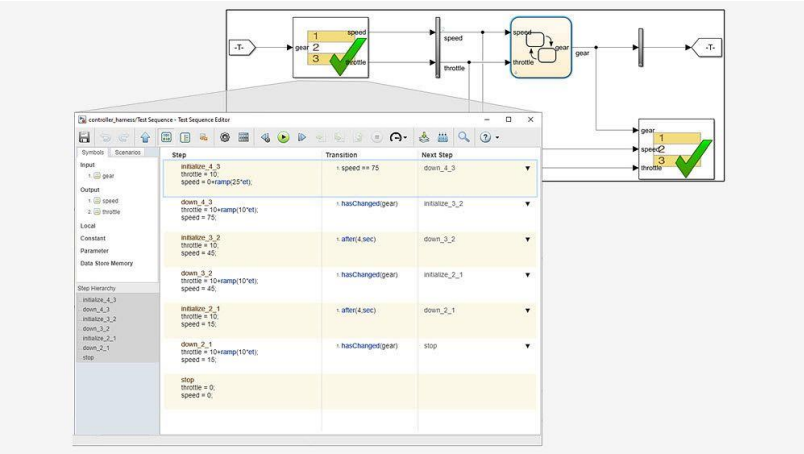
- Direction of travel
- Ego Vehicle Speed
- Cut-in Vehicle Speed
- Cut in Distance
- Vehicle Type

Future Development

- Scenario Design and Co-simulation with RoadRunner Scenario
- AI algorithm integration with Simulink
- Integration with Simulink Test and System Composer







Summary

- Advances in ADAS and AV technologies have driven the need to develop scenario-based methods versus the traditional methods.
- Fundamental core capabilities, including requirements management, scenario creation, vehicle dynamics modeling, and continuous integration and validation, etc., are needed to fill in the gap to address hardware-based development inefficiency
- Entire toolchain needs to be fully digitized
- Cloud-based computing platform is the future to go

Q/A