

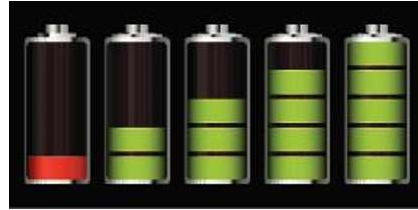
# MATLAB EXPO 2018

## **Full Vehicle Simulation for Electrification and Automated Driving Applications**

**Vijayalayan R & Prasanna Deshpande  
Control Design Application Engineering**



# Key Trends in Automotive Industry



**Vehicle Electrification**



**Autonomous Driving**

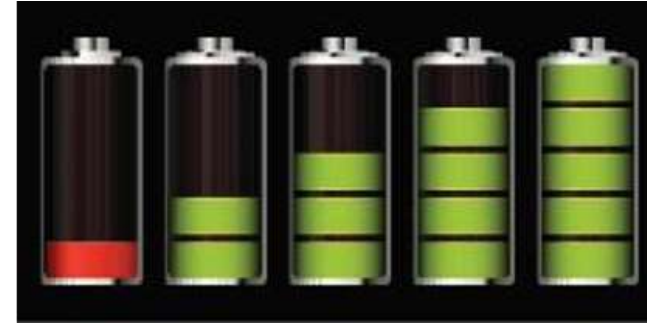


## Session Key Take Away

***Full vehicle simulation model addresses the new challenges posed by key automotive trends***

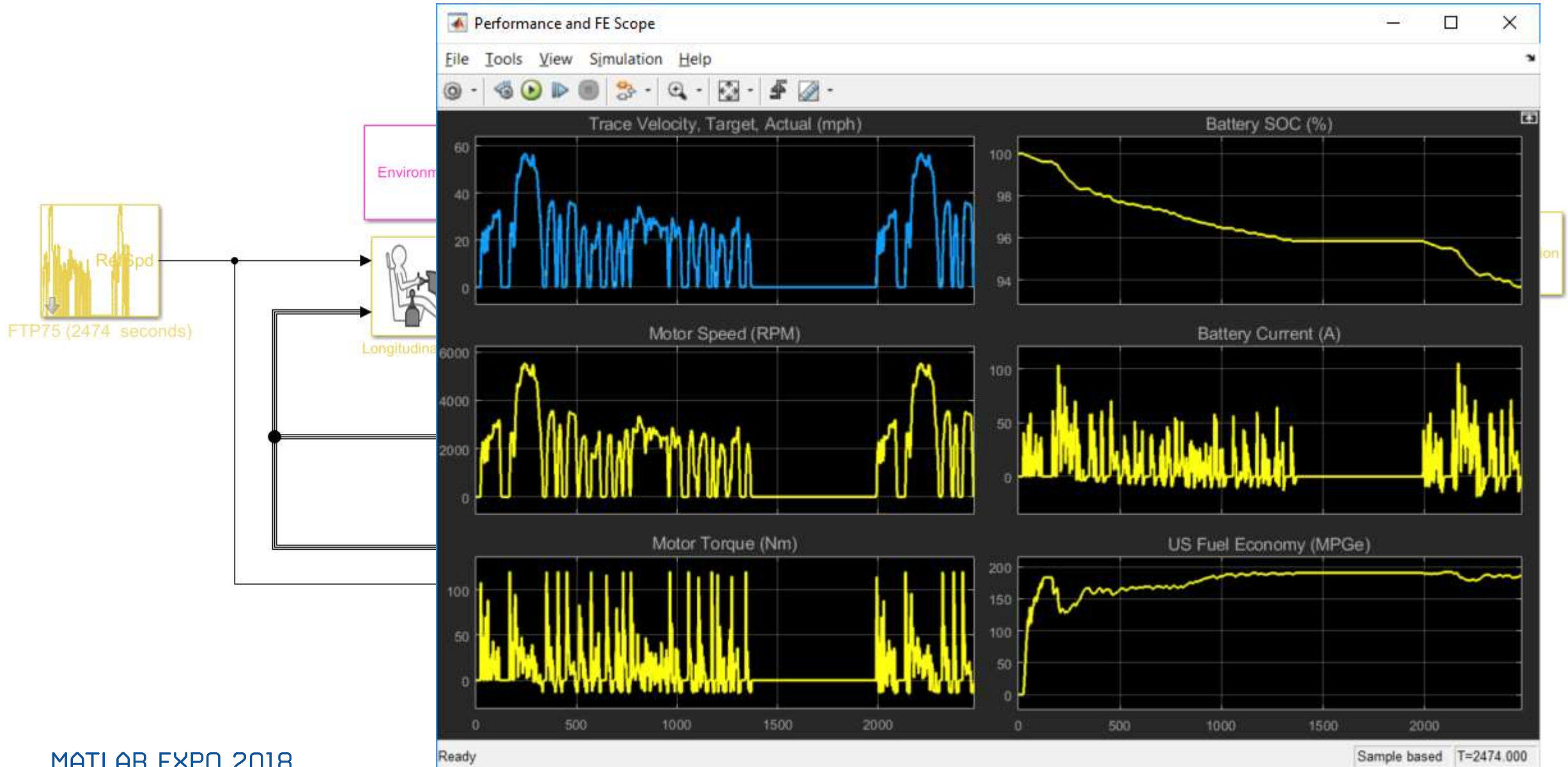
# How to build a full vehicle simulation model?

## Scenario- 1



## Vehicle Electrification

# What if we can build . . .

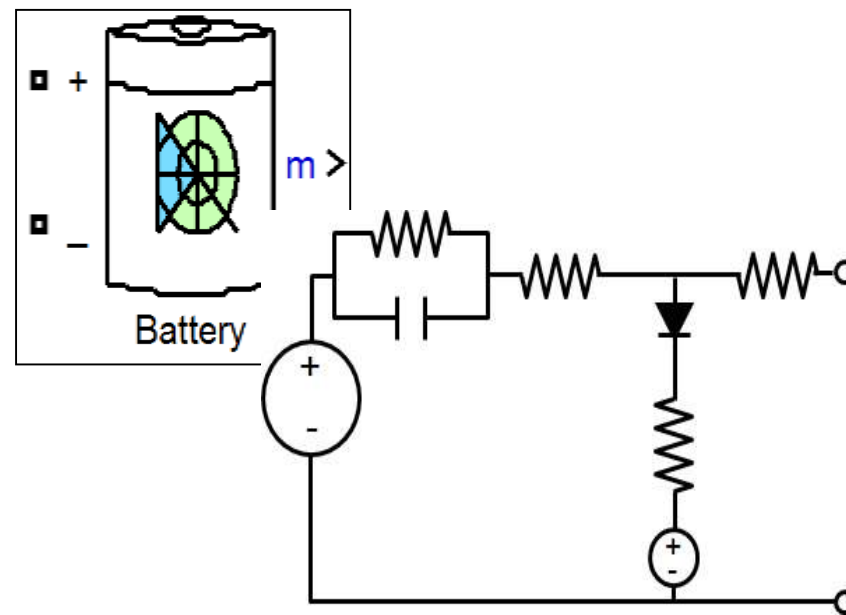
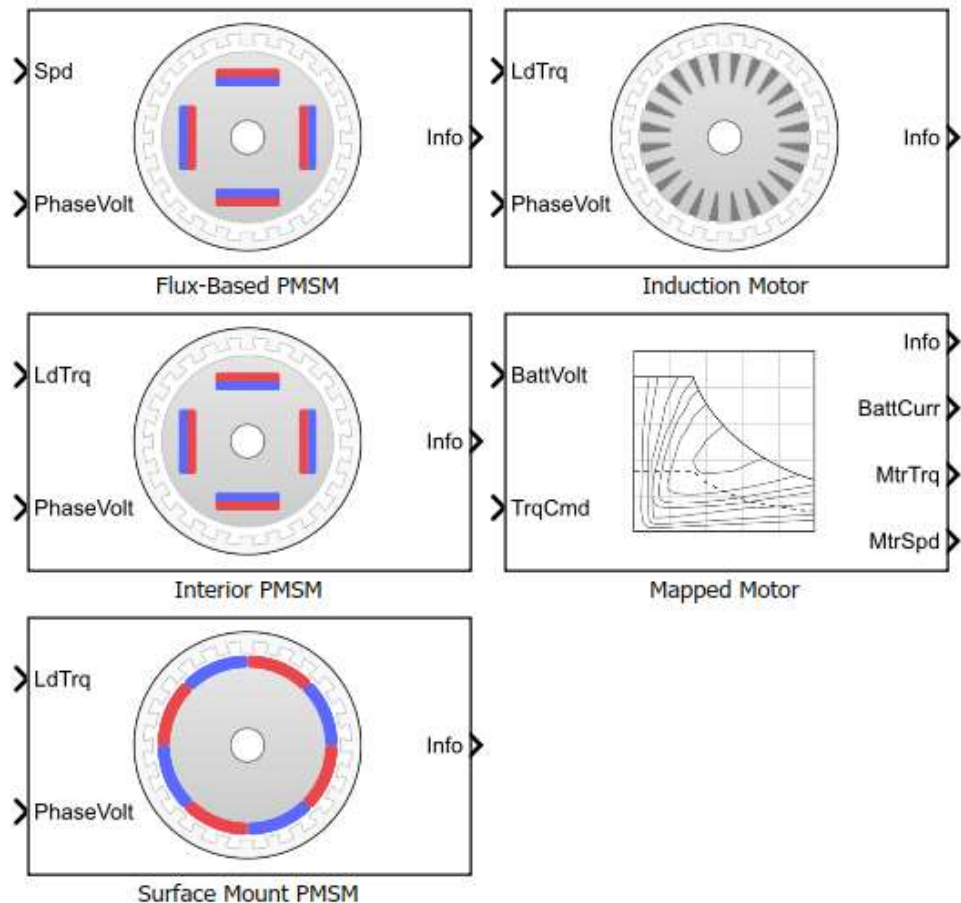


# Challenges for Powertrain Electrification

- Benchmark with existing vehicle and determine the requirements for electrification
- Component Selection
- Component Sizing
- Vehicle level performance analysis and optimization:
  - How do the selected components work together ? How does the vehicle perform?
  - With the above set of components, what best Fuel Economy / Range can I get from my vehicle?

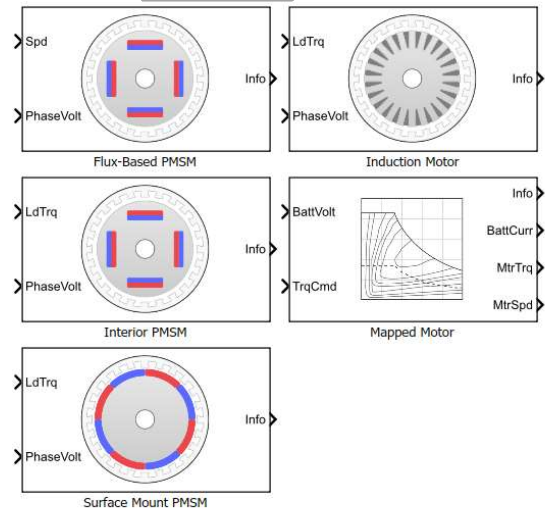
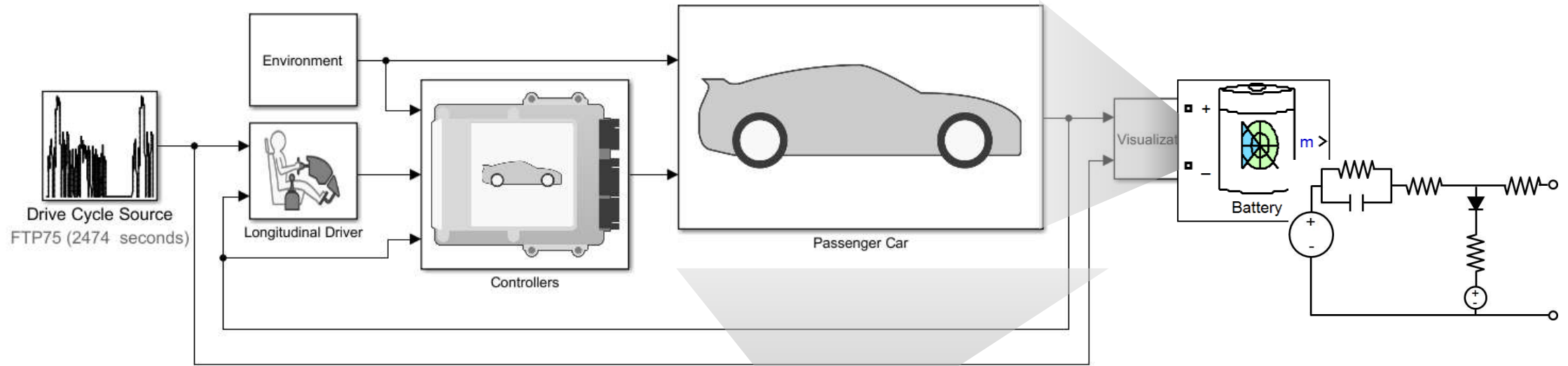


# Solution: Simulation Based Approach

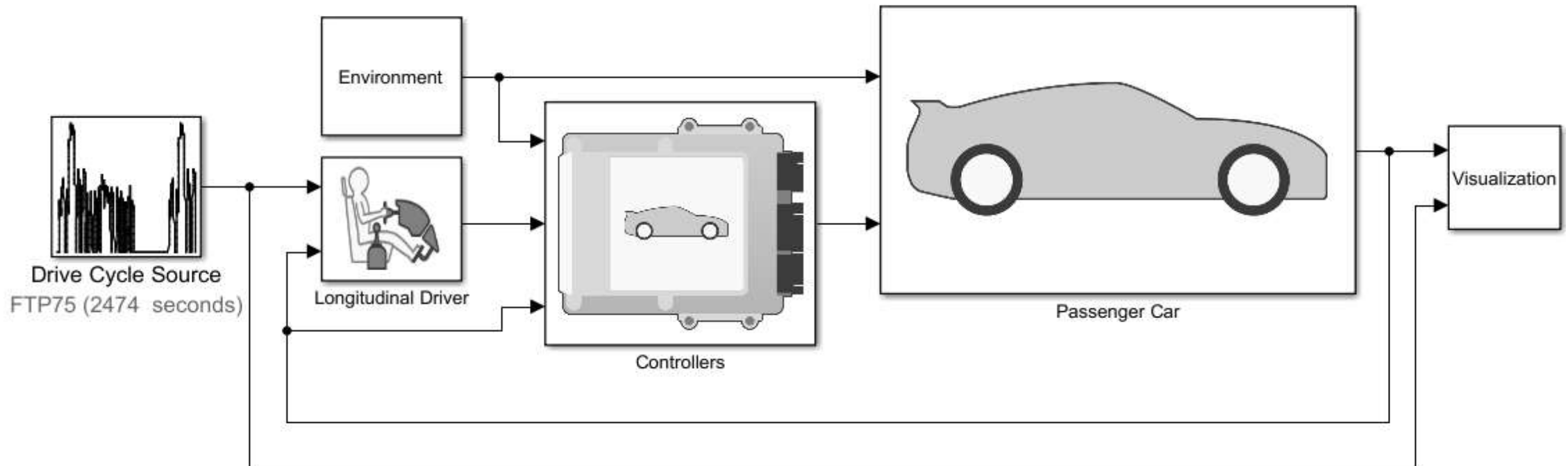




# Solution: Simulation Based Approach

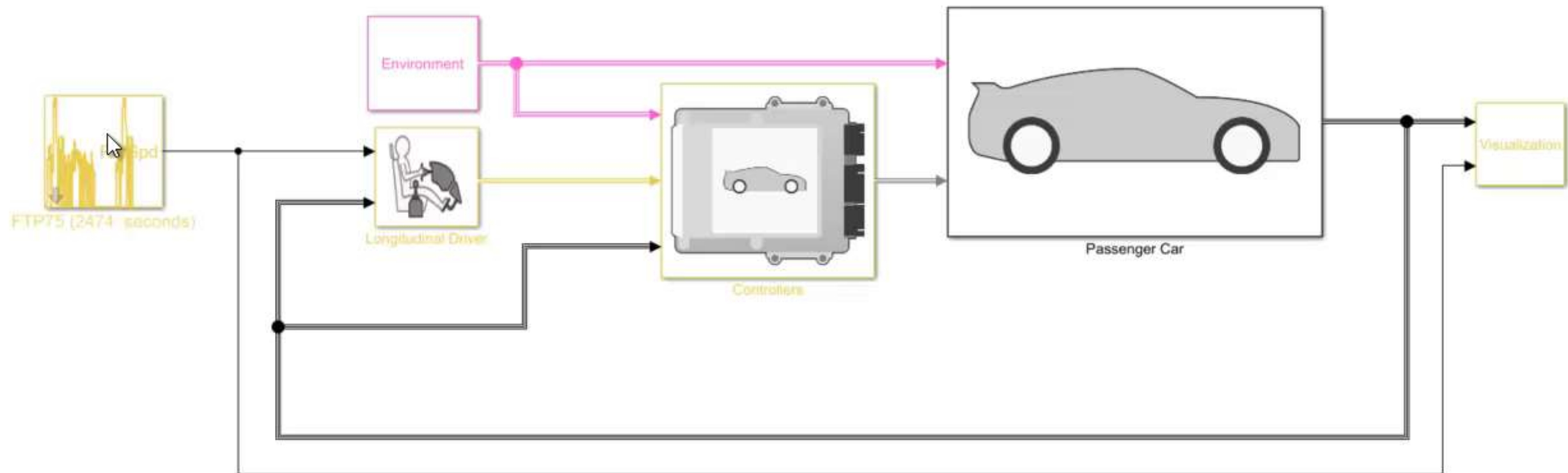


## What if we get . . .



- Good plant / controller models
- Open for customization and well documented models
- Very fast-running models that work with popular HIL systems

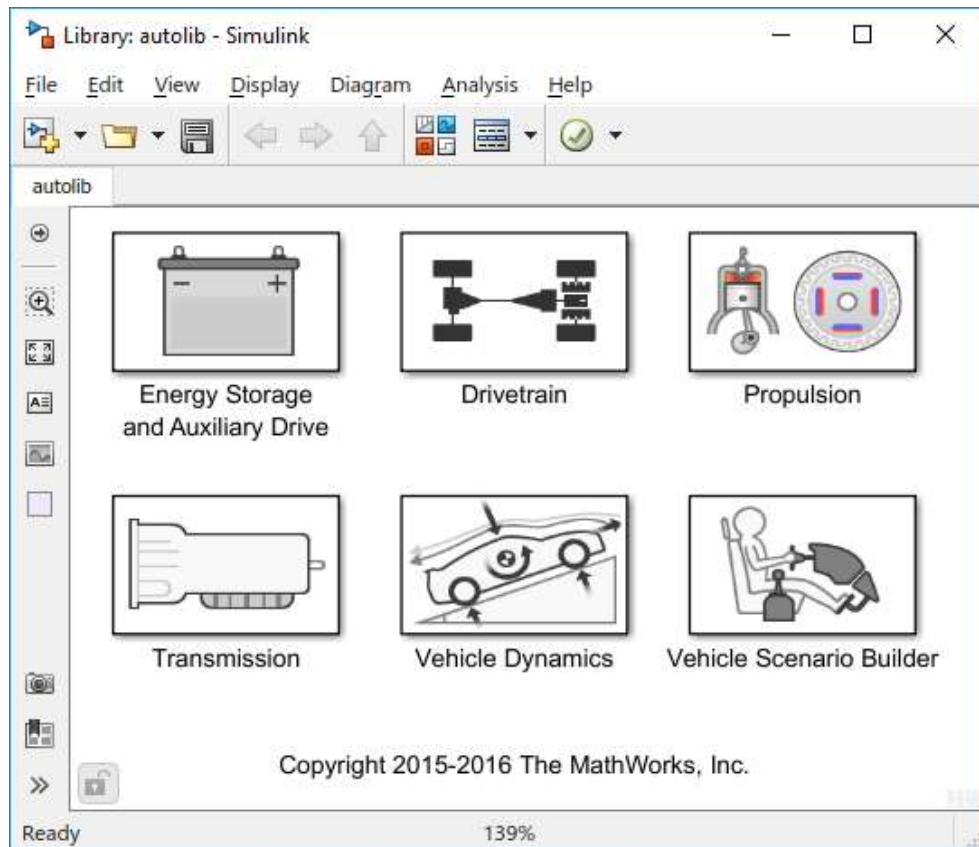
EvReferenceApplication



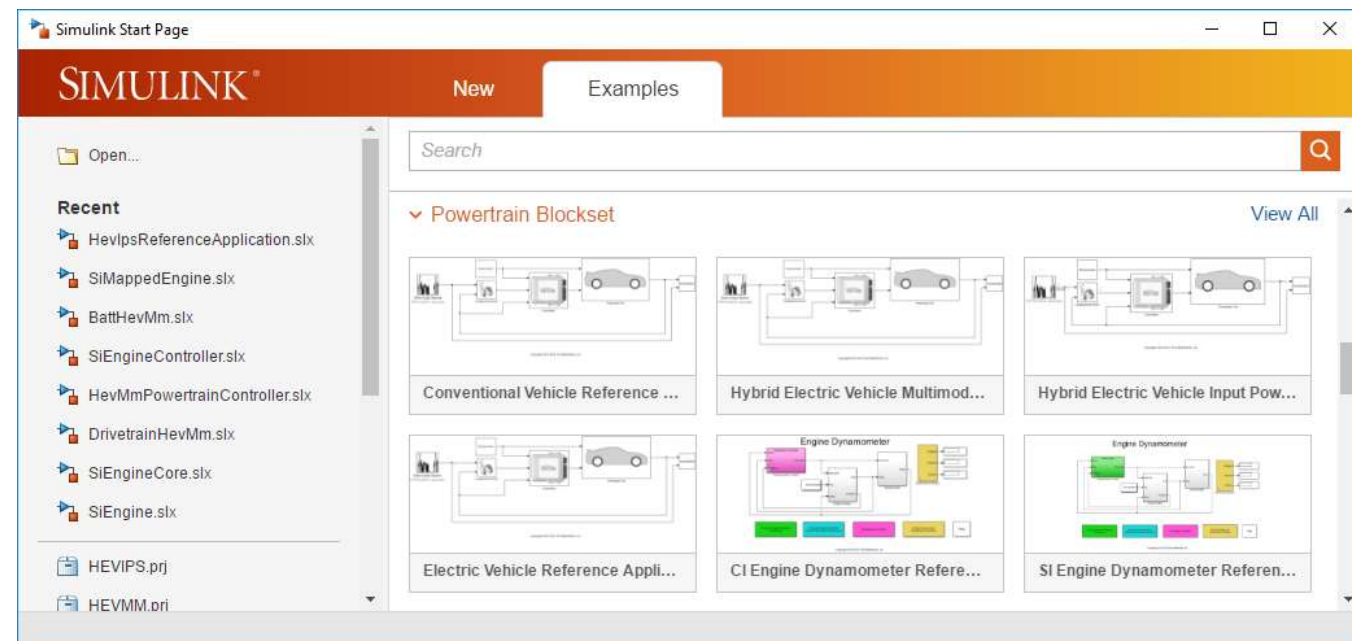
Copyright 2015-2017 The MathWorks, Inc.

# Pre-built reference applications can be used as great starting point, and library blocks help in customizing the system model

## Library of blocks



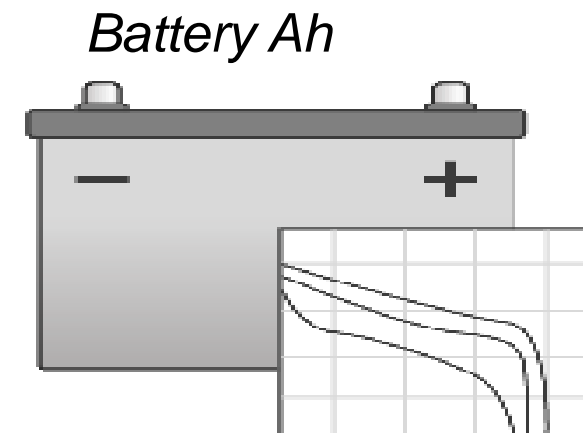
## Pre-built reference applications



# Vehicle System Level Models can be used for Design Optimization

- Maximize fuel economy of HEV / EV
- Minimize the time required for 0-60kmph
- Verify controller performance for different initial SoC points of battery
- Battery capacity or cell configuration
  - Ah rating
  - Number cells (or modules) in series / parallel
  - Affects vehicle mass

# Series, # Parallel ?



# Modeling Battery as a component and Modeling Battery as a system

**Datasheet Battery**

based off of discharge characteristics taken at battery datasheet or through experimental

**Parameters**

Rated capacity at nominal temperature, BattChargeMax [Ah]:

Open circuit voltage table data, Em [V]:

Open circuit voltage breakpoints 1, CapLUTBp []:

Internal resistance table data, RInt [Ohms]:

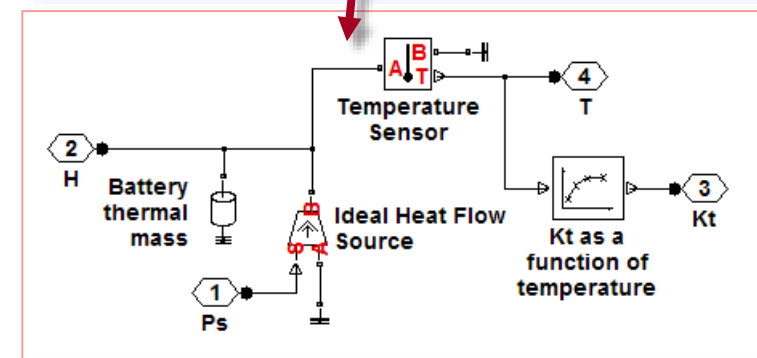
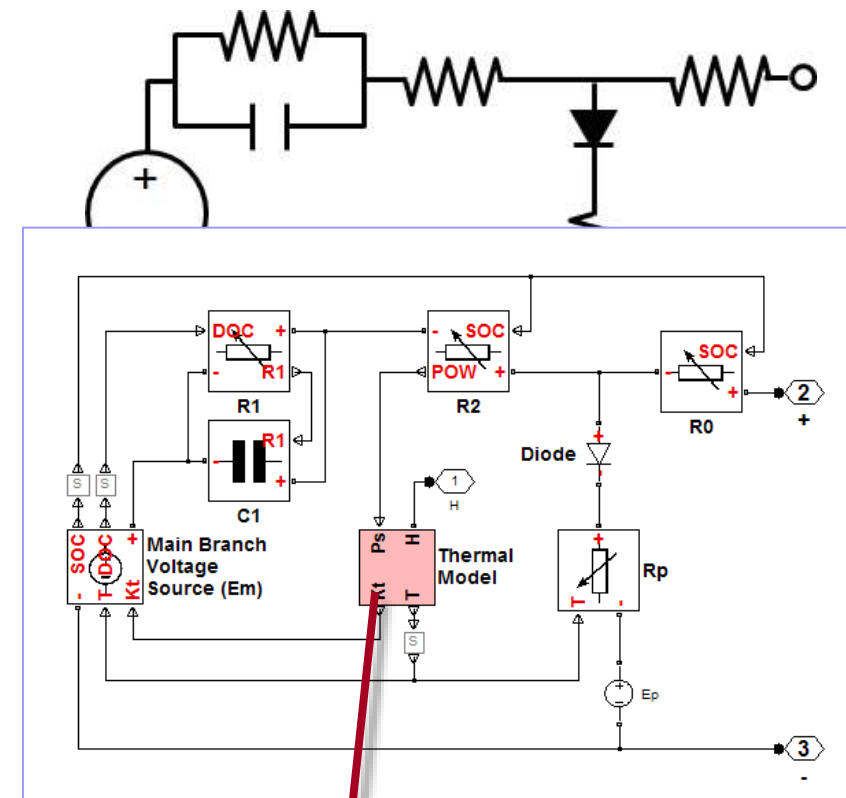
Battery temperature breakpoints 1, BattTempBp [K]:

Battery capacity breakpoints 2, CapSOCBp []:

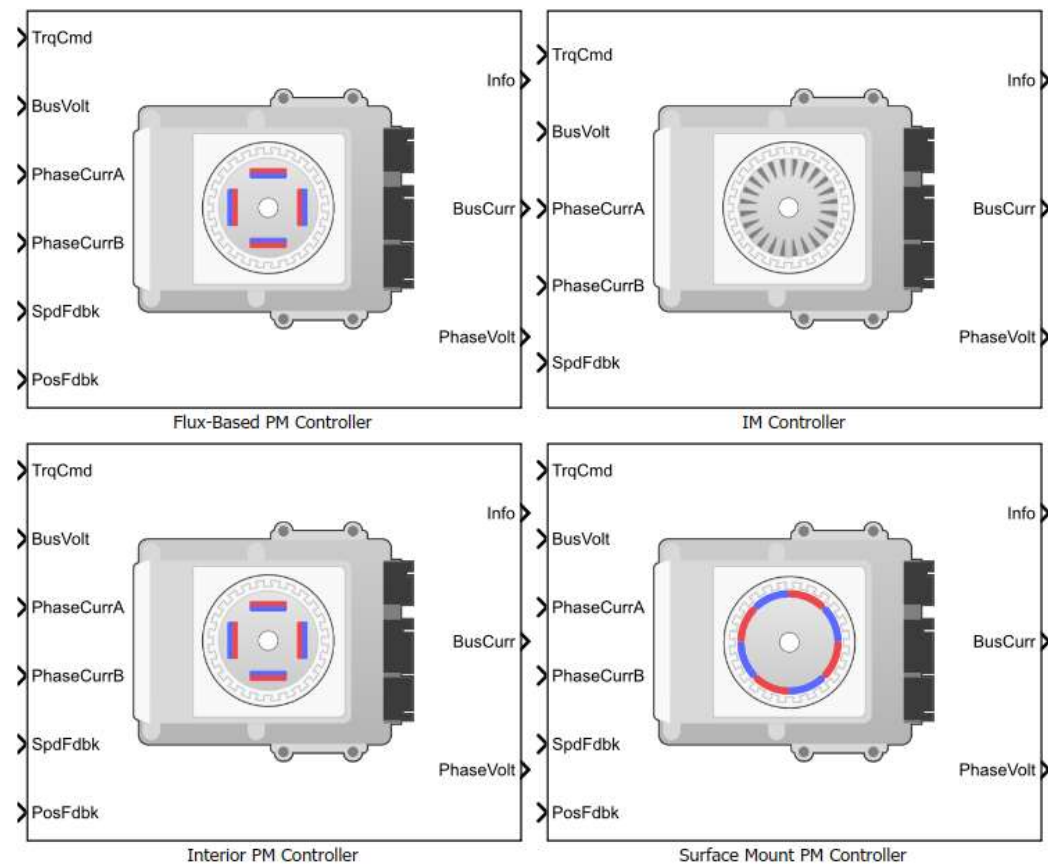
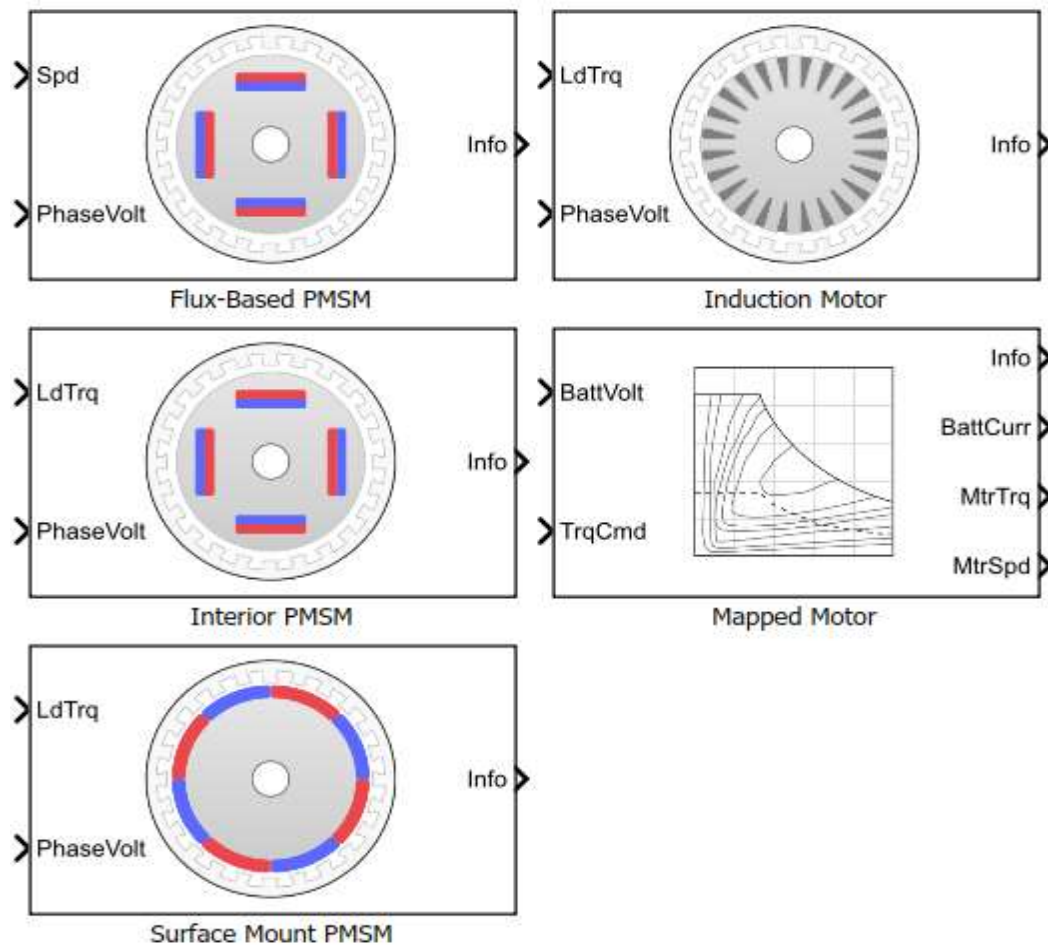
Number of cells in series, Ns []:

Number of cells in parallel, Np []:

Initial battery capacity, BattCapInit [Ah]:

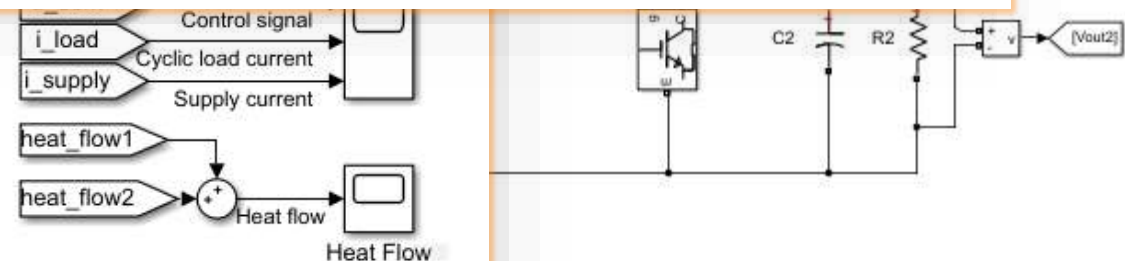
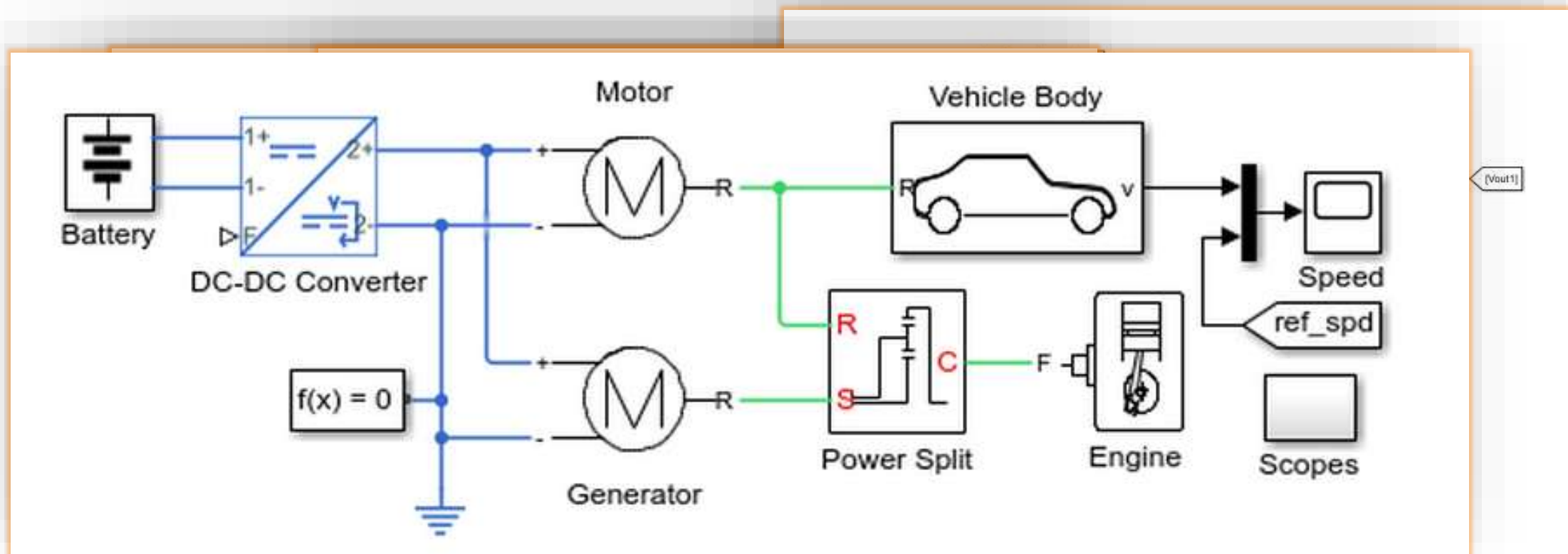


# System level models of motor and controller are available with Powertrain Blockset



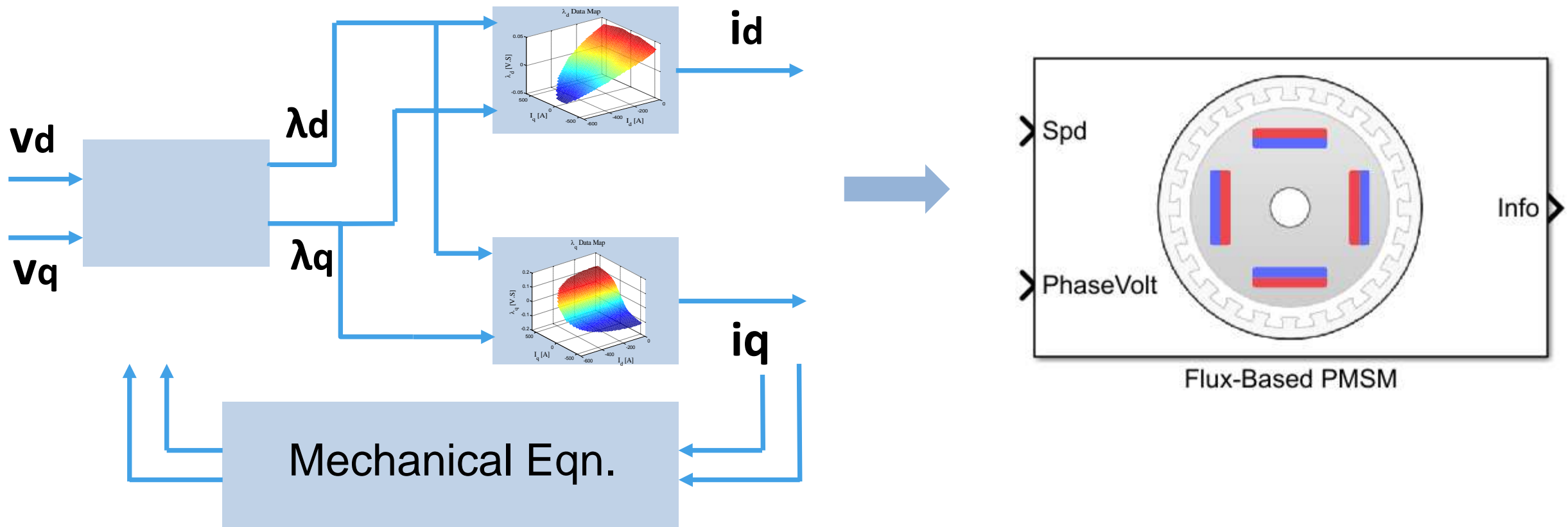


# Model Power Electronics, Thermal Dynamics, Vehicle Electrical network using Simscape



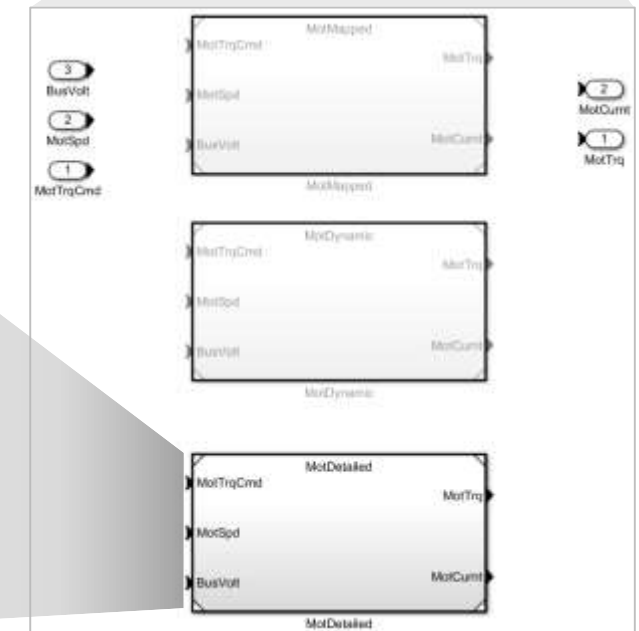
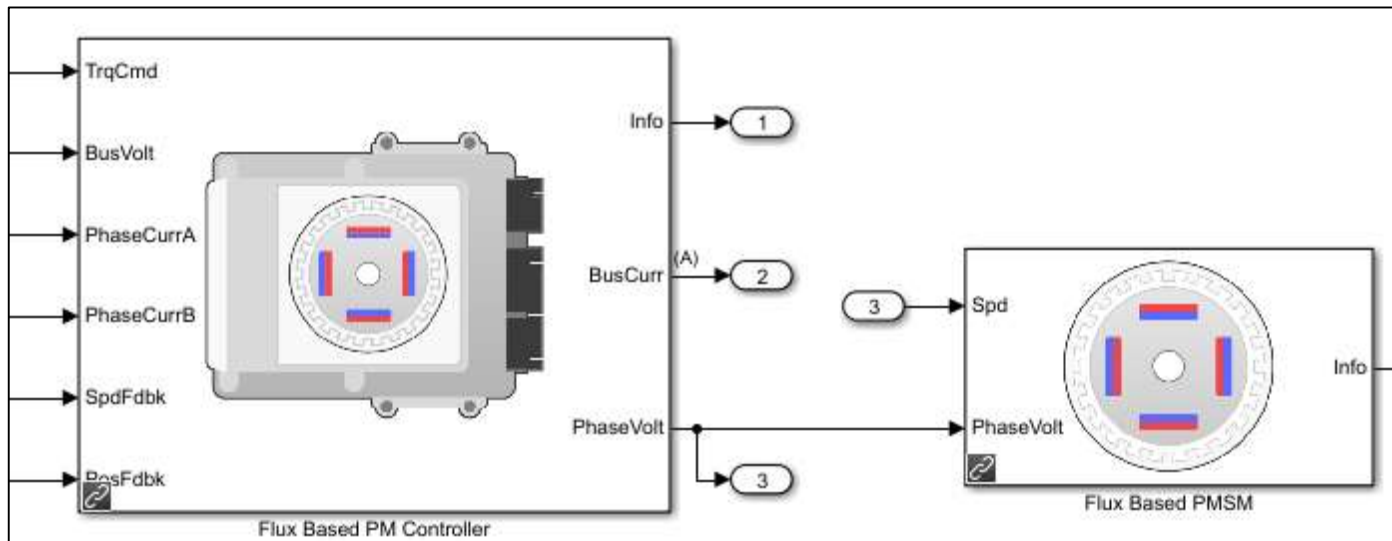
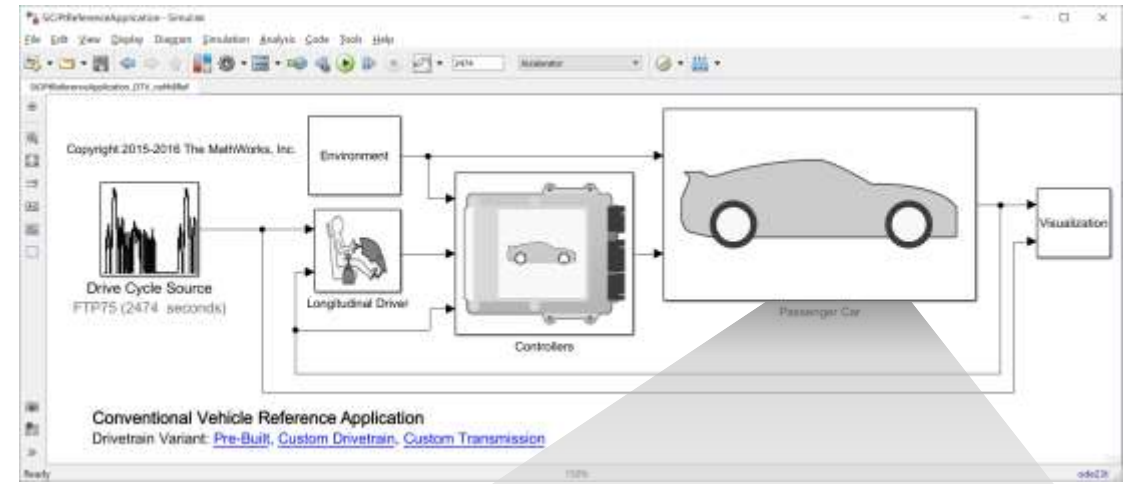
# High Fidelity Detailed Motor Model in Simscape

- FEA simulations or dynamometer data used to obtain non-linear flux table
- Flux-based PMSM model created to capture this effect



# Including Detailed Subsystem Variants

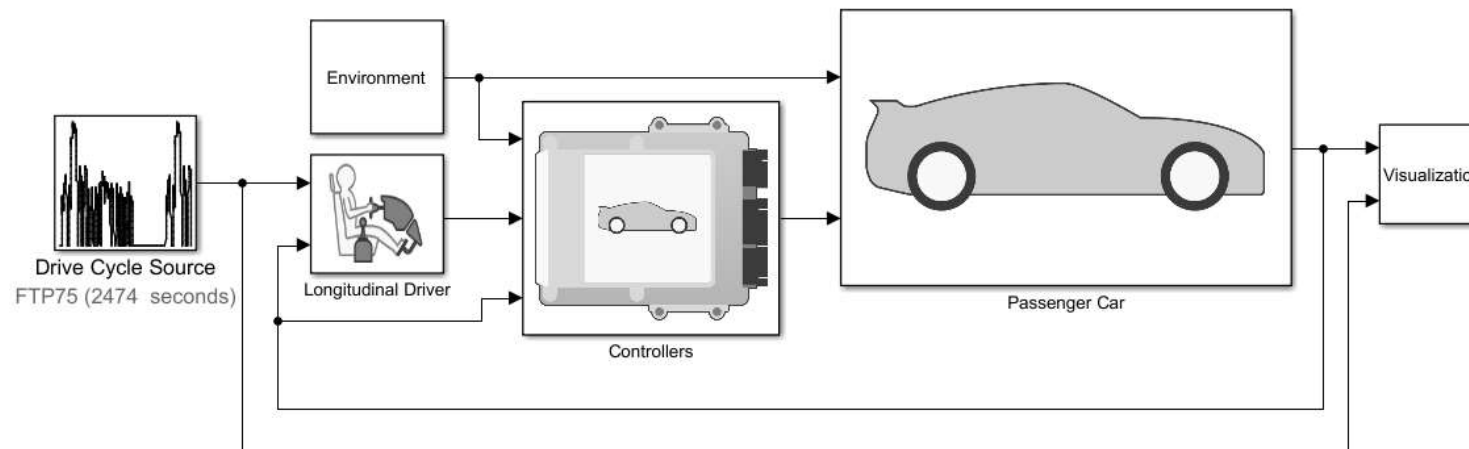
- Add your own subsystem variants to the existing vehicle models
  - Simulink-based
  - Simscape-based
  - S-function
  - FMI Interface



# Session Key Take Away

Full vehicle simulation model addresses the new challenges posed by key automotive trends.

- Reference application model from Powertrain Blockset can be used as a starting point for:
  - Design optimization studies
  - Multi-domain simulation via Simscape
  - Component controller design and parametrization
  - Hardware-in-the-loop (HIL) testing



## Scenario- 2



**Autonomous  
Driving**



**Chassis Control | Vehicle Dynamics | Automated Driving**

## What if we can...

- Model and simulate vehicle dynamics in a virtual 3D environment
- Ride & handling: Characterize vehicle performance under standard driving maneuvers
- Chassis controls: Design and test chassis control systems
- ADAS / AD: Create virtual 3D test ground for ADAS and automated driving features



Ride & handling



Chassis controls



ADAS / AD

# Challenges faced by Chassis Controls, Vehicle Dynamics & ADAS Engineers

- Quickly achieving a good vehicle design with limited number of prototype builds
- Verifying system behavior for conditions that are too time consuming or risky to test on the road
- Single simulation environment for design and verification of controls



# Challenges faced by Chassis Controls, Vehicle Dynamics & ADAS Engineers

- **Quickly achieving a good vehicle design with limited number of prototype builds**
- Verifying system behavior for conditions that are too time consuming or risky to test on the road
- Single simulation environment for design and verification of controls

## Ride and Handling Study: Double Lane Change at 30 mph

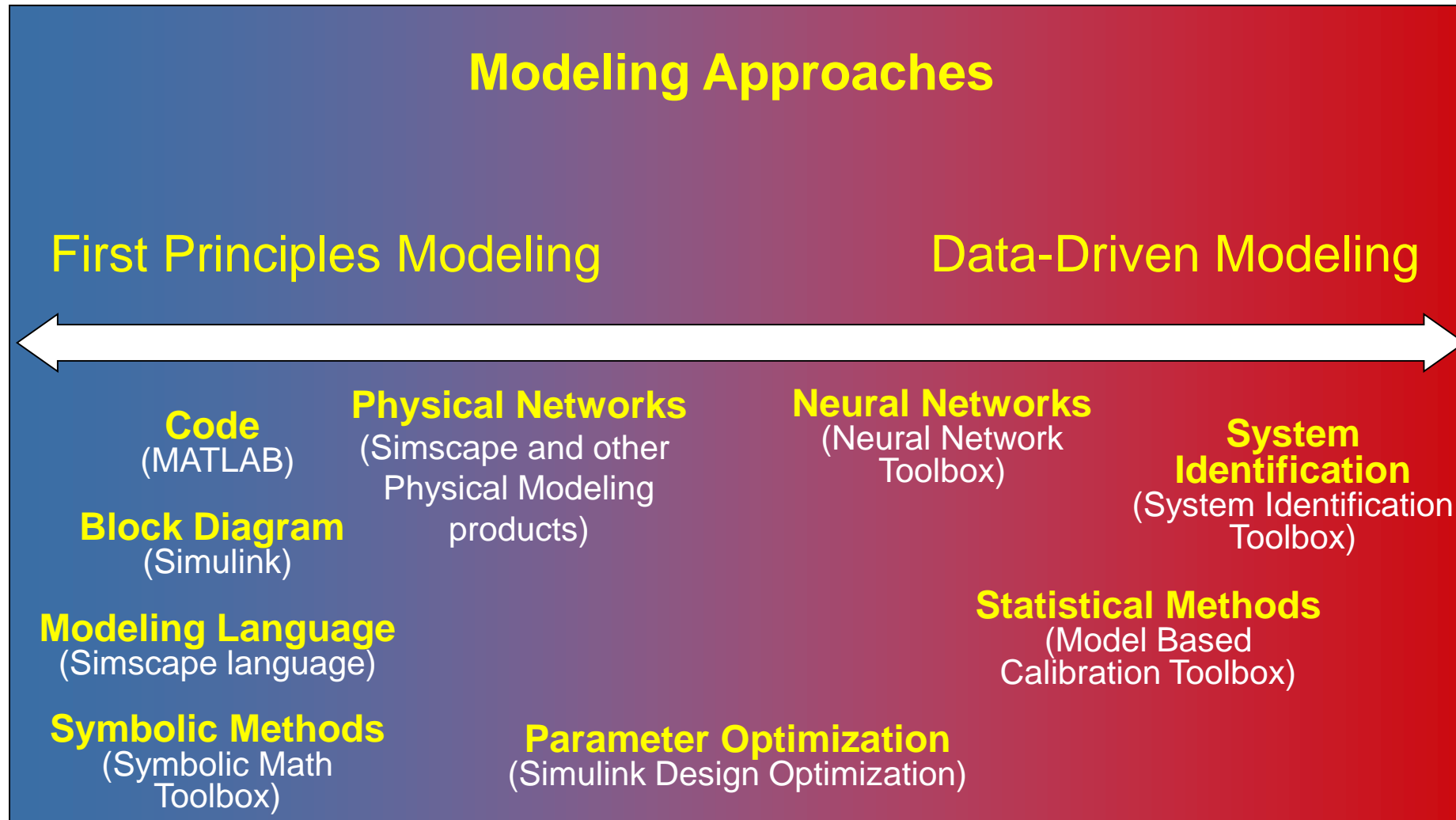


- Test the obstacle avoidance performance of a vehicle as per ISO 3888-2

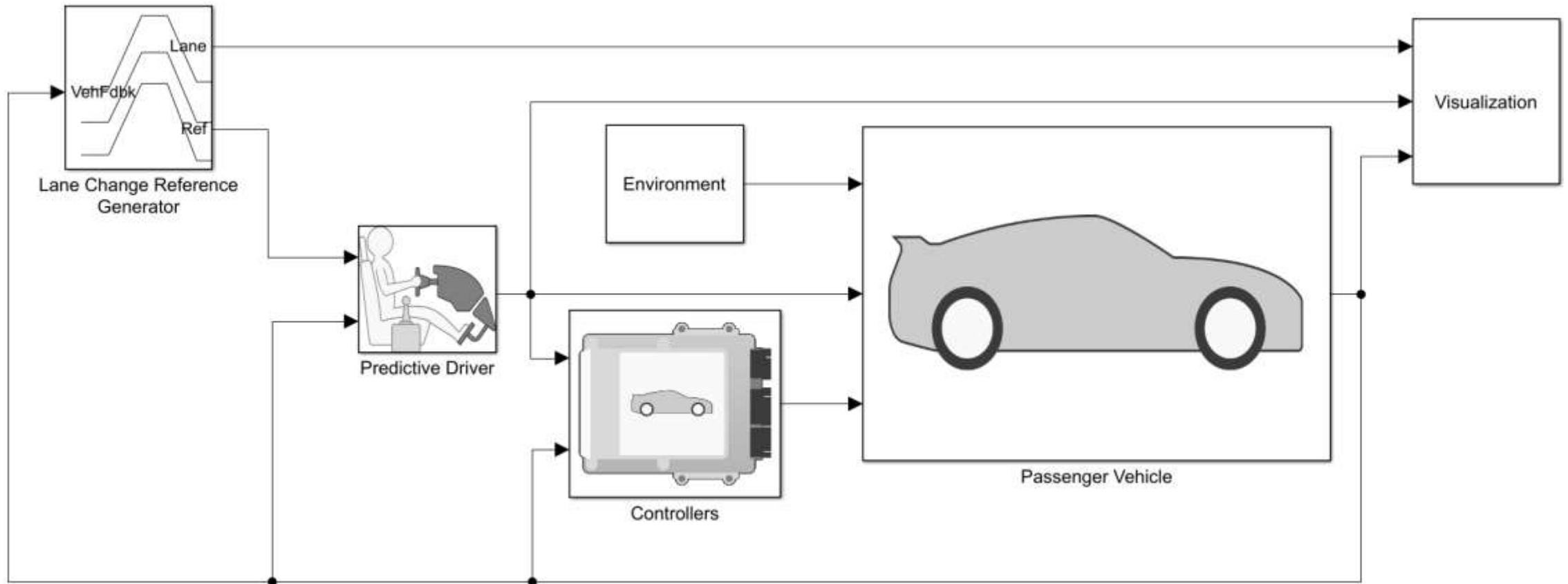
In the test, the driver:

- Accelerates until vehicle hits a target velocity
- Releases the accelerator pedal
- Turns steering wheel to follow path into the left lane
- Turns steering wheel to follow path back into the right lane

# Modeling Dynamic Systems in the Simulink Environment

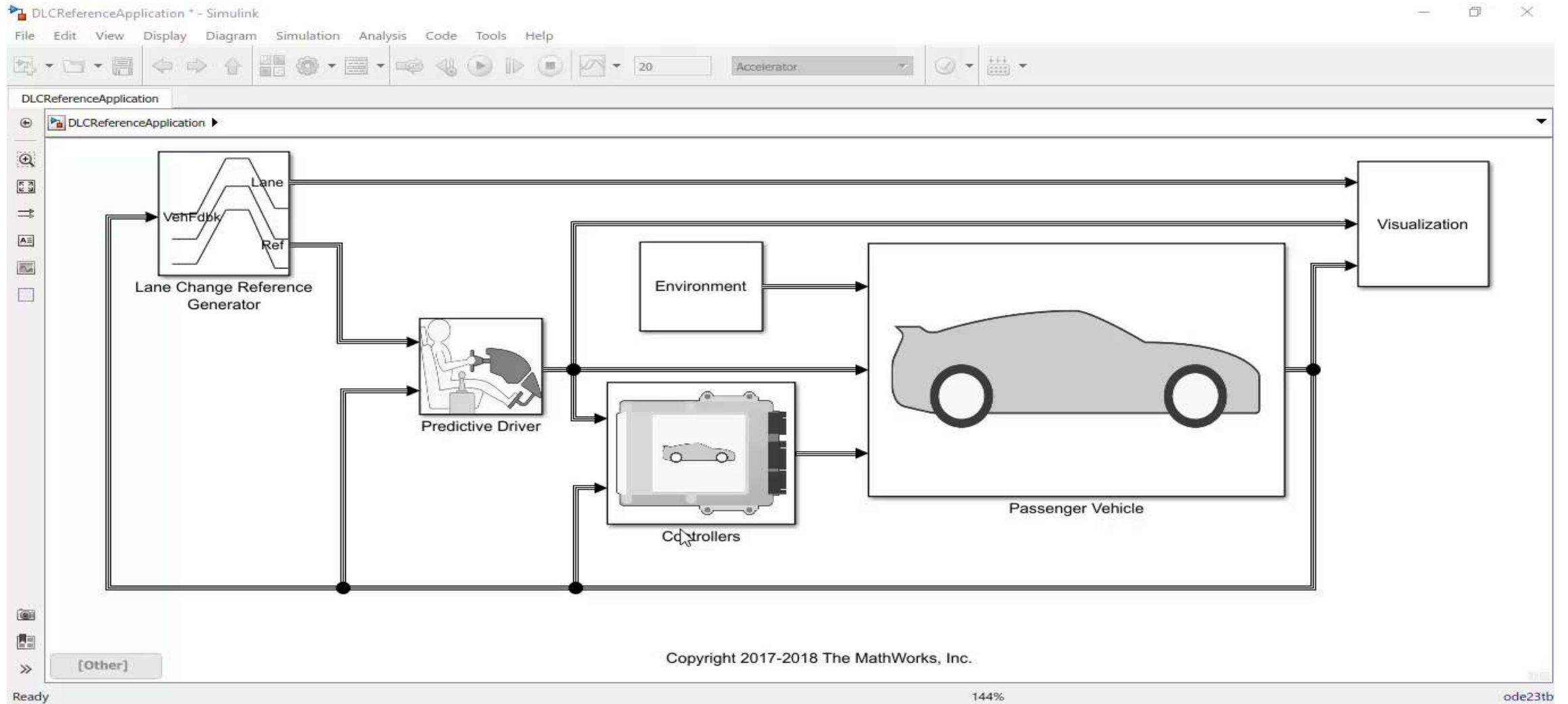


# Need for a good starting point to build good plant/controller models



***Lower the barrier to entry for Model-Based Design***

# Demo : Double Lane Change Reference Application



# Challenges faced by Chassis Controls, Vehicle Dynamics & ADAS Engineers

- Quickly achieving a good vehicle design with limited number of prototype builds
- **Verifying system behavior for conditions that are too time consuming or risky to test on the road**
- **Single simulation environment for design and verification of controls**

# Game Engine Co-Simulation

## Simulink

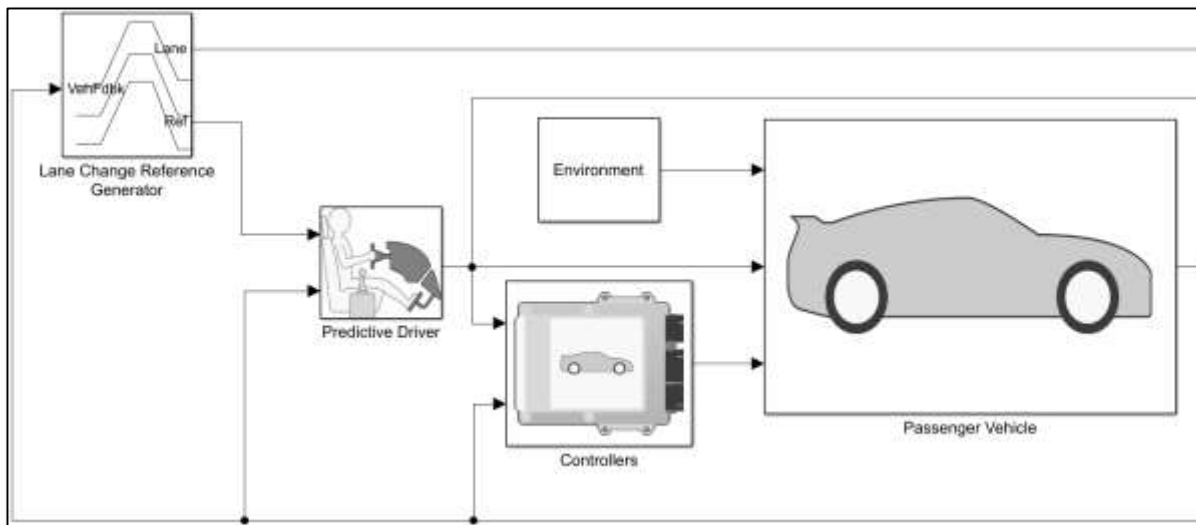
- Physics of vehicle
- Initialization of game engine camera

vehicle / camera location

## Unreal Engine

- Rendering / lighting
- Physics of non-Simulink objects
- Collision detection

camera image, ground height, ...





# Stop Sign Detection and Braking

The screenshot displays a Simulink model for a vehicle control system, titled "SCRRReferenceApplication/Controller and Display - Simulink". The main workspace shows two primary blocks: "Controller" and "Stop Sign Detector".

**Controller Block:** This block processes reference signals. It includes:
 

- Inputs: `single(1)`, `single(1)`, `single(0)`, `single(-0.1)` (Left), and `single(0.1)` (Right).
- Logic: A gain block of `.5` produces `ThrCmd`. A switch selects between "Straight" and "Right" steering commands, which are then processed by gain blocks of `0` and `10` to produce `SteerCmd`.
- Output: A switch selects between `ThrCmd` and `SteerCmd` based on a `isnan` block (set to `true`).

**Stop Sign Detector Block:** This block handles image processing:
 

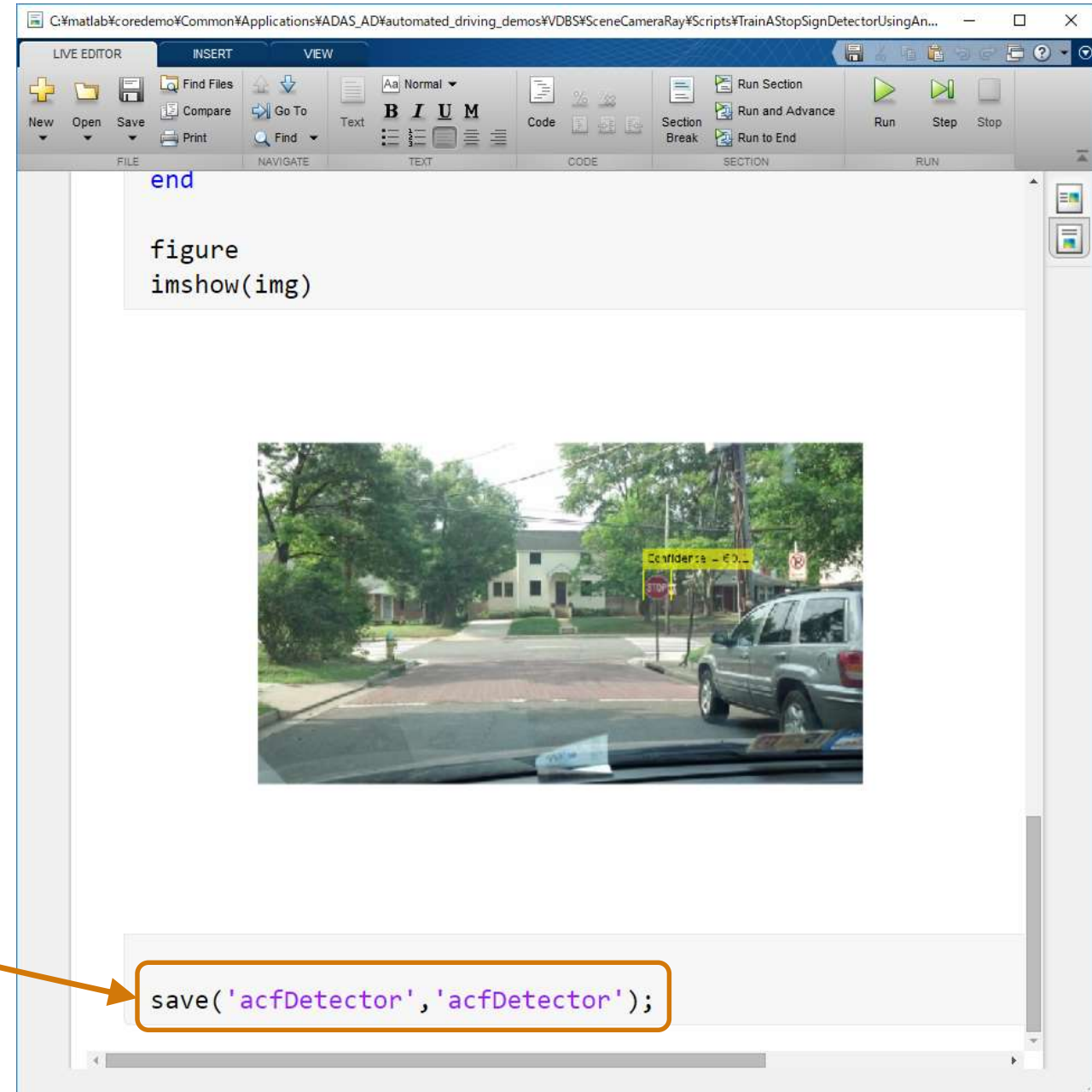
- Input: `imgData`.
- Process: `Convert Image to double` → `img StopSignDetector` (outputting `bbox` and `score`) → `Image Draw Rectangles` (outputting `Pts`) → `Image To Video Display`.

**Viewer Scope:** A graph window showing a plot with a y-axis from 0 to 1 and an x-axis from 0 to 4.5. The plot area is currently empty. The status bar indicates "Running" and "Sample based T=3.570".

**To Video Display:** A window showing a 3D perspective view of a road with a car in the distance. The status bar at the bottom indicates "Running", "80%", "t=3.600", "72%", and "FixedStepDiscrete".

# Training Stop Sign Detector

- Train a stop sign detector as an ACF object detector
- The detector is trained based on [the CVST example](#) and saved as a MAT-file



```
end

figure
imshow(img)

save('acfDetector', 'acfDetector');
```

The screenshot shows the MATLAB Live Editor interface. The code editor contains the following code:

```
end

figure
imshow(img)

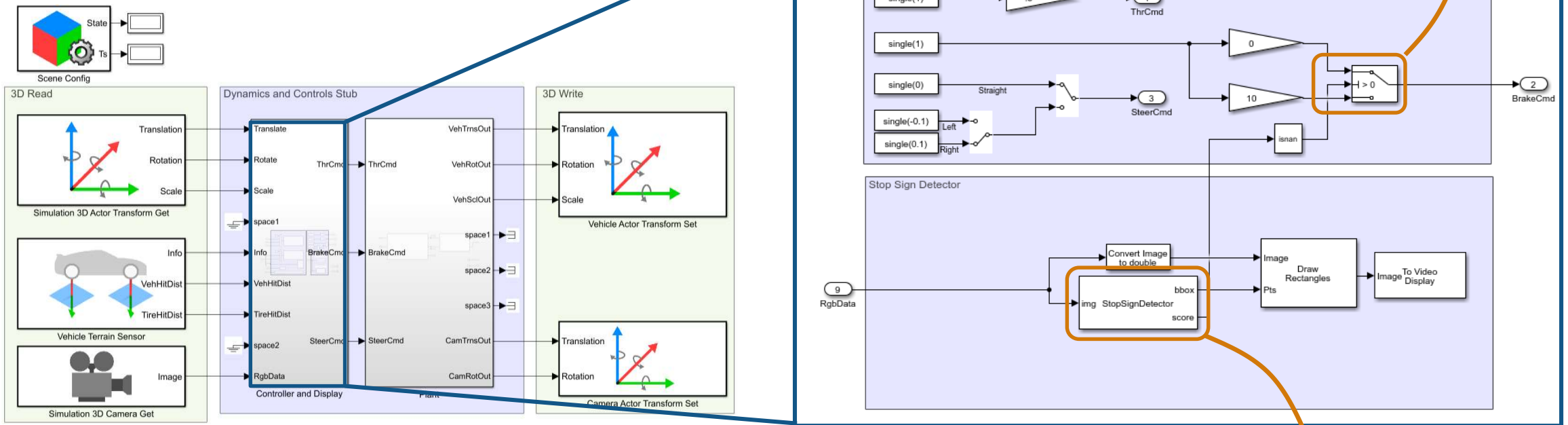
save('acfDetector', 'acfDetector');
```

An orange arrow points from the text "saved as a MAT-file" in the list to the `save('acfDetector', 'acfDetector');` line in the code editor. Below the code editor, there is a preview window showing a street scene with a stop sign and a car. The stop sign is highlighted with a yellow bounding box.

# Implementing Braking Logic

- Start with Scene Interrogation reference application
- Add braking logic to stop when the stop sign appears

Add switching logic



Add stop sign detector as MATLAB System Object

# ADAS / AD Testing: Virtual 3D Scene



Camera sensor sends video to Simulink

Synthetic video used for testing vision-based algorithms (e.g., lane detection)



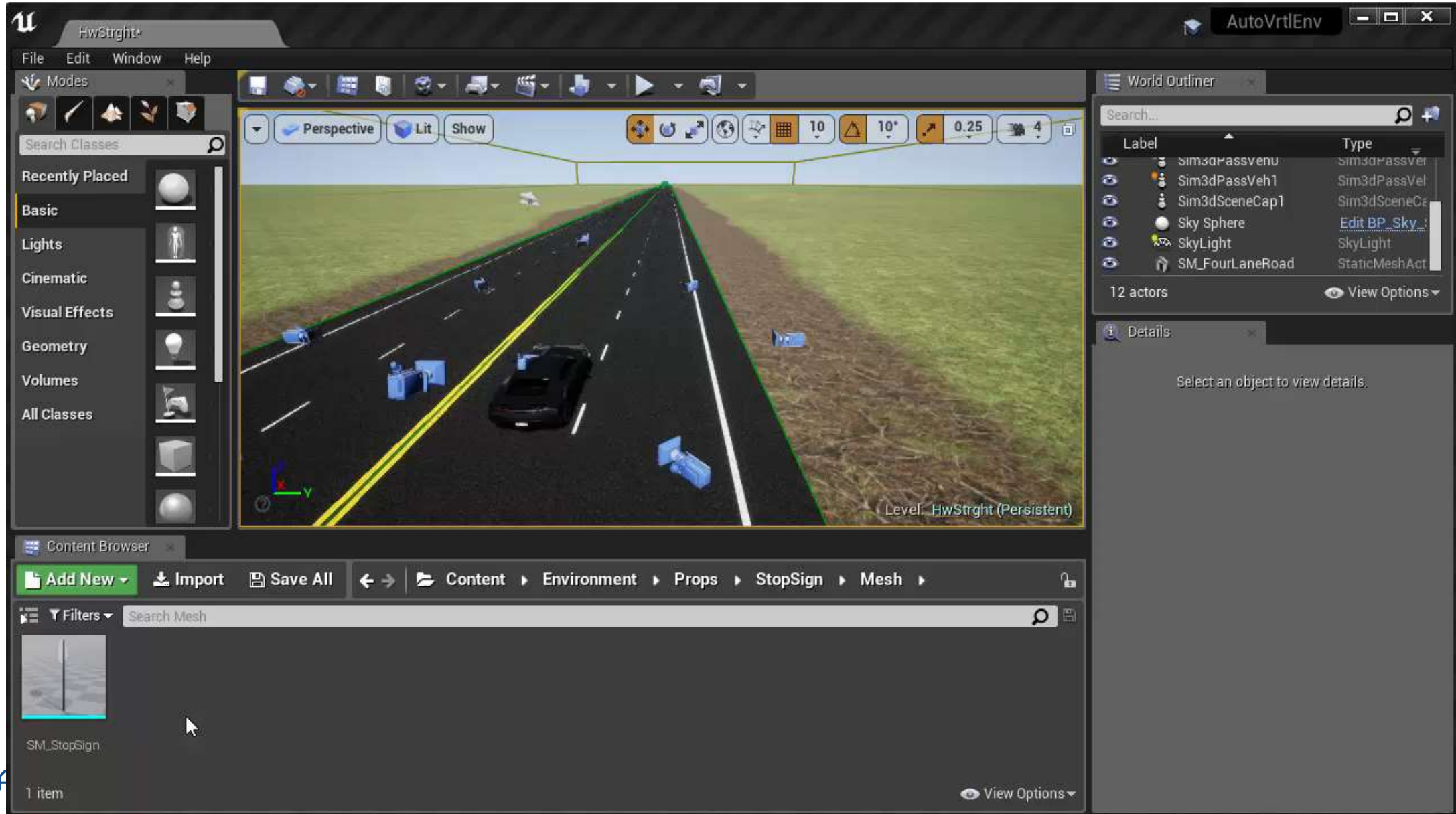
# Changing the Lighting to Night Conditions

The image shows a Simulink workspace with the following components:

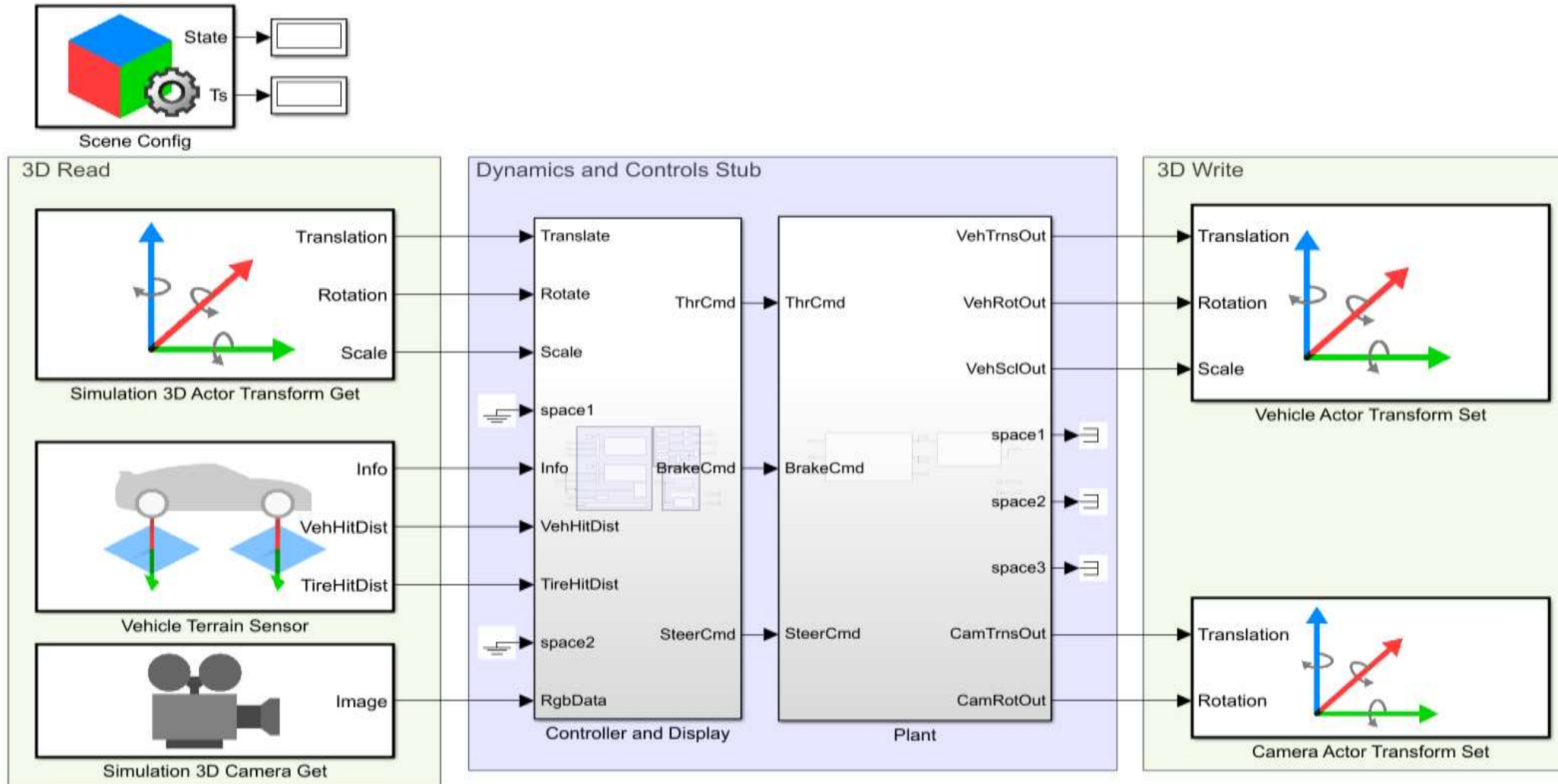
- SCRReferenceApplication/Controller and Display - Simulink**: The main workspace containing:
  - Controller**: A block diagram with inputs for throttle and steering, gain blocks (0.5, 0, 10), and a switch for steering direction (Straight, Left, Right).
  - Stop Sign Detector**: A block diagram that takes image data, converts it to double, uses a `StopSignDetector` block to find bounding boxes and scores, and then draws rectangles on the image for display.
- Viewer Scope**: A window showing a plot with a y-axis from 0.9 to 1.1 and an x-axis from 0 to 4.5. The plot area is currently black, indicating no data is being displayed.
- To Video Display**: A window showing a first-person view of a road at night. The road is dark with white dashed lines. A red car is visible in the distance. The sky is dark with some light on the horizon.

At the bottom of the Simulink interface, the status bar shows "Running" and "FixedStepDiscrete".

# Editing Support Package Scene to Add Stop Sign



# Configuring the interface to the 3D environment





# Customizing Scene with Support Package

- Create your own scenes with Unreal Editor and our Simulink plug-in
- Unreal Editor project files available in our Support Package:  
[“Vehicle Dynamics Blockset interface for Unreal Engine 4”](#)



The screenshot displays the MathWorks File Exchange interface. At the top, the MathWorks logo is on the left, and navigation links for Products, Solutions, Academia, Support, Community, and Events are on the right. Below this is a blue header bar with 'File Exchange' on the left and a search box labeled 'Search File Exchange' on the right. A secondary navigation bar contains links for MATLAB Central, Files, Authors, Tags, Comments, My File Exchange, and Submit. The main content area features a file listing for 'Vehicle Dynamics Blockset interface for Unreal Engine 4', which includes a thumbnail image of a car on a road, the file version (1.0, 15.1 KB), the author (MathWorks Automotive Community Profile), and a brief description: 'Simulink integration for Unreal Engine 4'.

# Model and simulate vehicle dynamics in a virtual 3D environment: Vehicle Dynamics Blockset

- Use Vehicle Dynamics Blockset for:
  - Ride & handling: characterize vehicle performance under standard driving maneuvers
  - Chassis controls: design and test chassis control systems
  - ADAS / AD: create virtual 3D test ground for ADAS and automated driving features



Ride & handling



Chassis controls

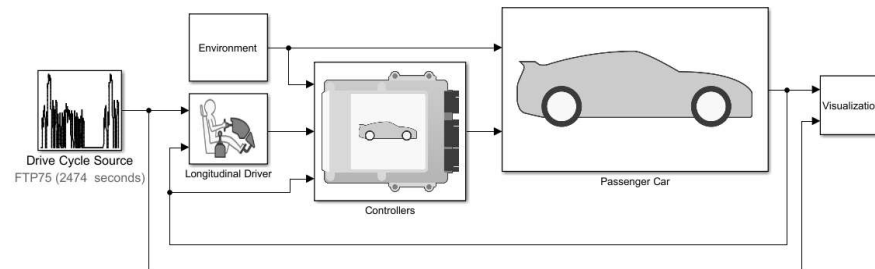


ADAS / AD

# Session Key Take Away

Full vehicle simulation model addresses the new challenges posed by key automotive trends.

- **MathWorks provides vertical products to serve automotive industry, including**
  - **Powertrain Blockset: powertrain controls, fuel economy and performance simulation**



- **Vehicle Dynamics Blockset: ride and handling, chassis controls, AD / ADAS testing**



## Call to action

- In case you are working on EV applications, you can get started with [Exploring the Reference Application Model of an Electric Vehicle](#)
- Else if you are working on Vehicle Dynamics/Chassis Controls/Automated Driving, you can get started with [Building a Vehicle Dynamics Model](#)
- After identifying a problem statement ,you can evaluate the new products using a 30 day trial license

# Thank You

- **Share your experience with MATLAB & Simulink on Social Media**
  - Use #MATLABEXPO
  - I use #MATLAB because..... Attending #MATLABEXPO
  - Examples
    - I use #MATLAB because it helps me be a data scientist! Attending #MATLABEXPO
    - Learning new capabilities in #MATLAB and #Simulink at #MATLABEXPO.
- **Share your session feedback:**

Please fill in your feedback for this session in the feedback form

## Speaker Details

Email: [Prasanna.Deshpande@mathworks.in](mailto:Prasanna.Deshpande@mathworks.in)

LinkedIn: <https://in.linkedin.com/in/deshprasan>

Twitter: [@InfPrasanna](https://twitter.com/InfPrasanna)

## Speaker Details

Email: [Vijayalayan.R@mathworks.in](mailto:Vijayalayan.R@mathworks.in)

LinkedIn: <https://in.linkedin.com/in/Vijayalayanr>

Twitter: [@r\\_Vijayalayan](https://twitter.com/_r_Vijayalayan)