

The background of the image is a blurred crowd of people. In the foreground, a hand is visible, wearing a silver bracelet with a large gemstone. Overlaid on the left side of the image is a large, stylized graphic composed of several overlapping triangles in shades of blue and orange. The text 'MATLAB EXPO 2018 KOREA' is positioned to the right of this graphic.

MATLAB EXPO 2018
KOREA

MATLAB EXPO 2018

[Subtrack 2]

Vehicle Dynamics Blockset 소개

김종헌 부장



Agenda

- What is Vehicle Dynamics Blockset?
- How can I use it?

Agenda

- **What is Vehicle Dynamics Blockset?**
- How can I use it?

Background

■ Context

- Automotive OEM's and Tier 1 suppliers must assess vehicle's dynamic performance
 - Will the vehicle roll over?
 - What's the stopping distance of the vehicle?
 - Do the stability controls perform adequately?
- Answer questions by building prototypes and / or running simulations



■ Challenges

- Prototypes are expensive, so must achieve a good design as early as possible
- Specialized vehicle dynamics simulation software is quite expensive and difficult to use
- Integrating 3rd party vehicle dynamics software with Simulink controls is cumbersome

Vehicle Dynamics Blockset

New product (R2018a)

- Model and simulate vehicle dynamics in a virtual 3D environment
- 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

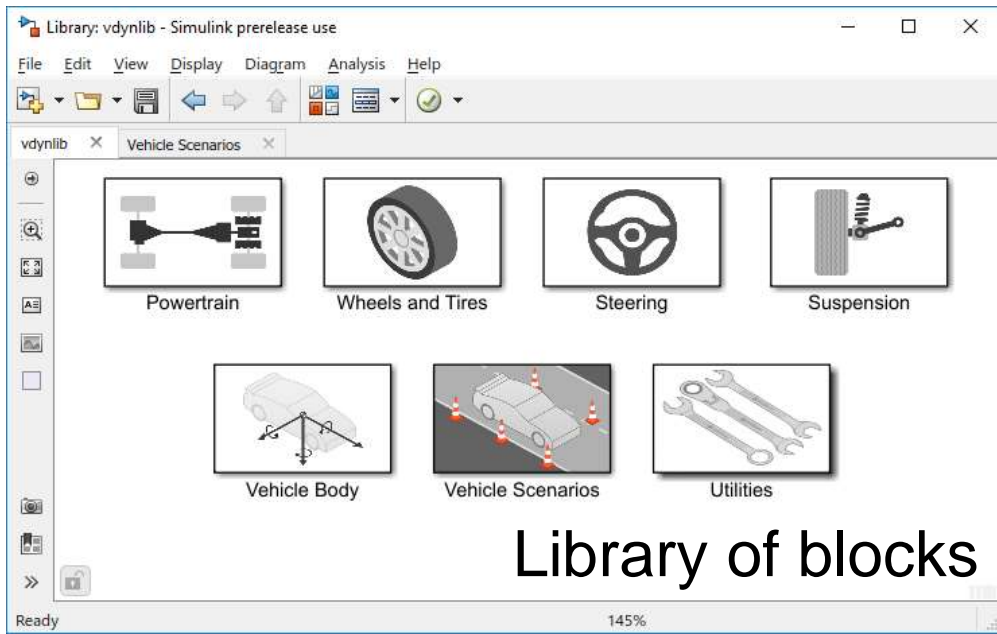


Chassis controls

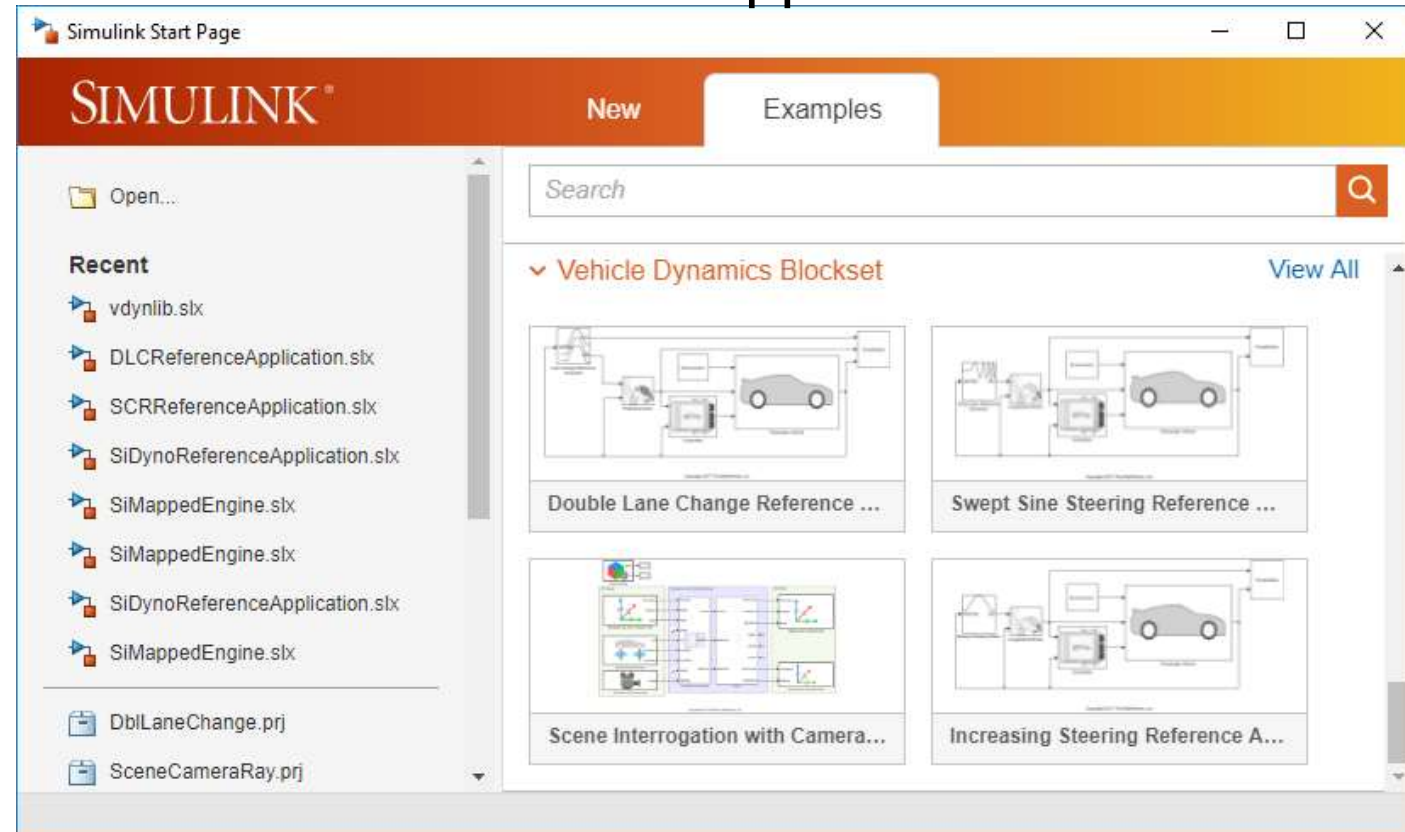


ADAS / AD

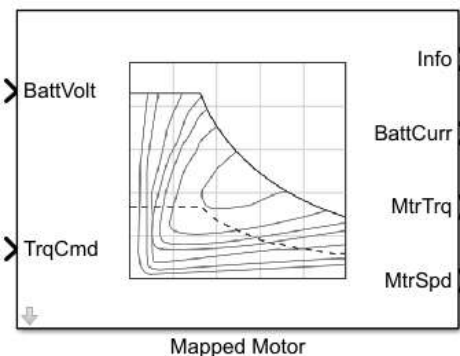
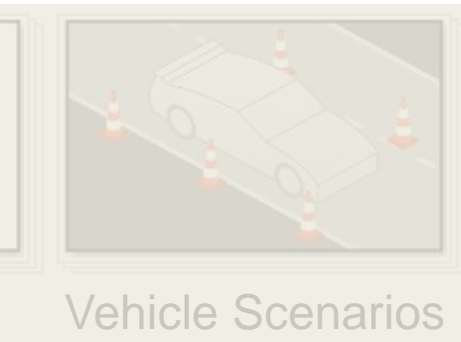
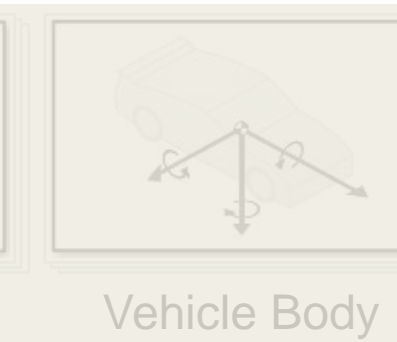
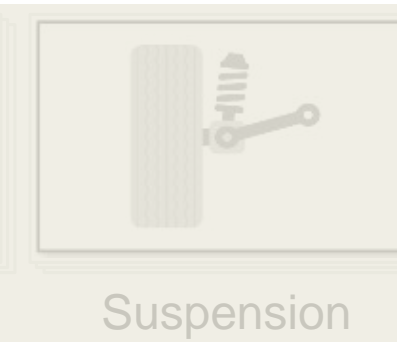
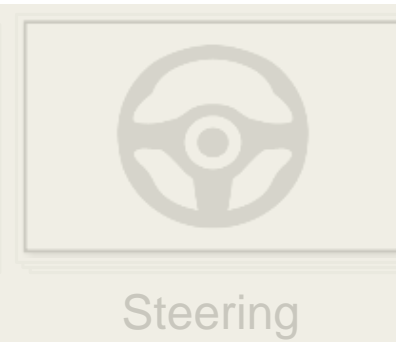
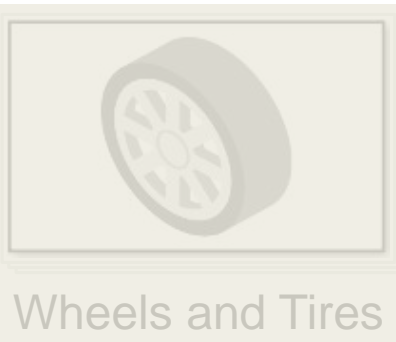
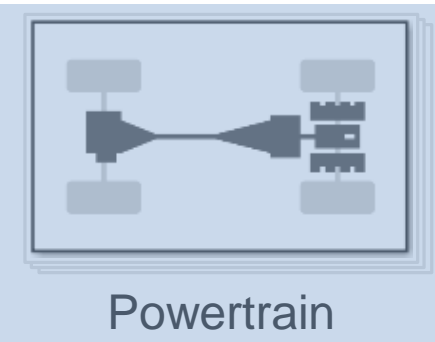
Vehicle Dynamics Blockset Features



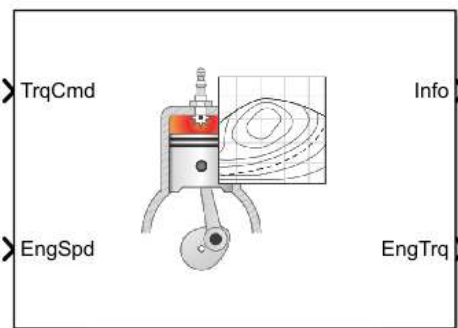
Pre-built reference applications



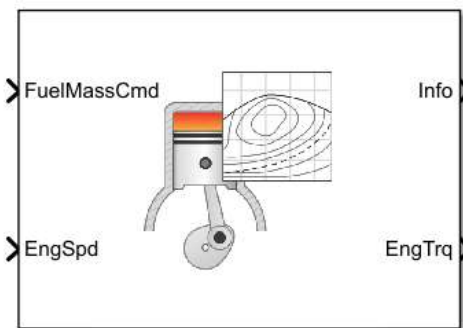
Block Library: Powertrain



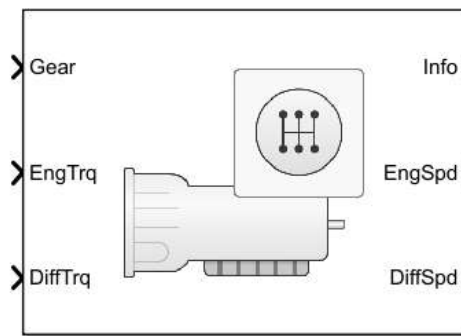
Mapped Motor



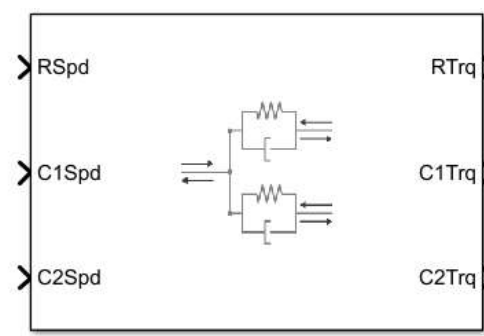
Mapped SI Engine



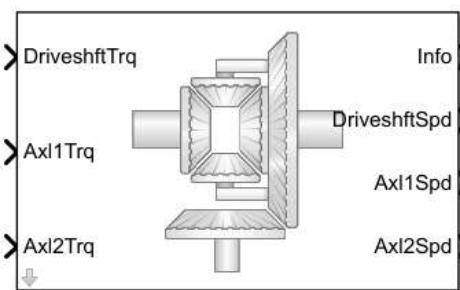
Mapped CI Engine



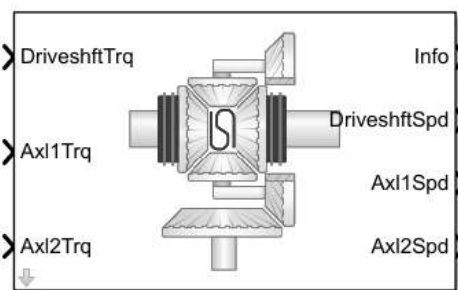
Ideal Fixed Gear Transmission



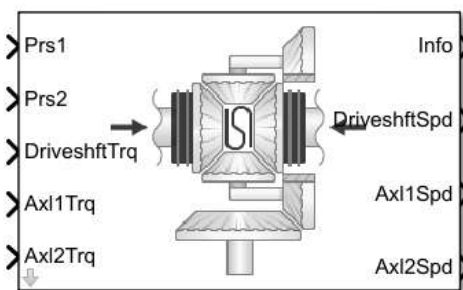
Split Torsional Compliance



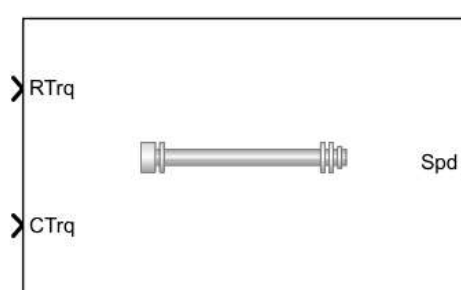
Open Differential



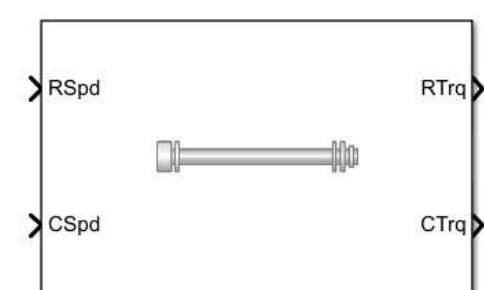
Limited Slip Differential



Active Differential

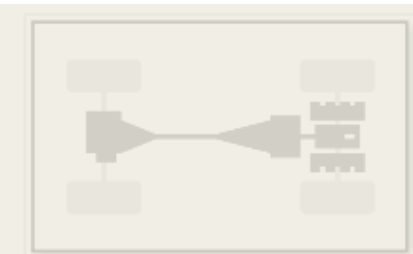


Rotational Inertia



Torsional Compliance

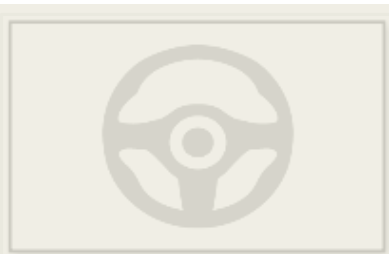
Block Library: Wheels and Tires



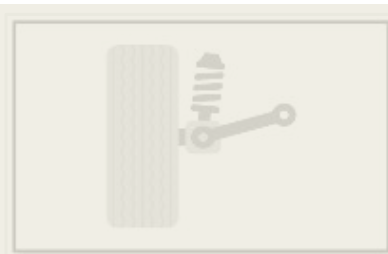
Powertrain



Wheels and Tires



Steering



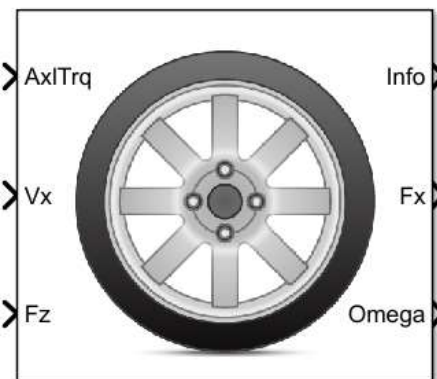
Suspension



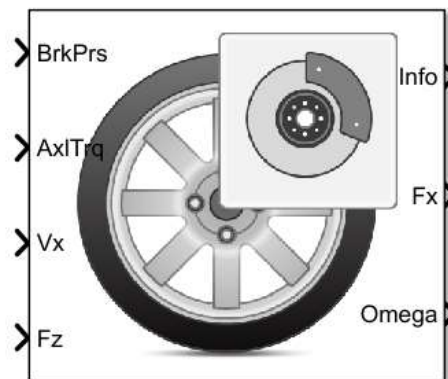
Vehicle Body



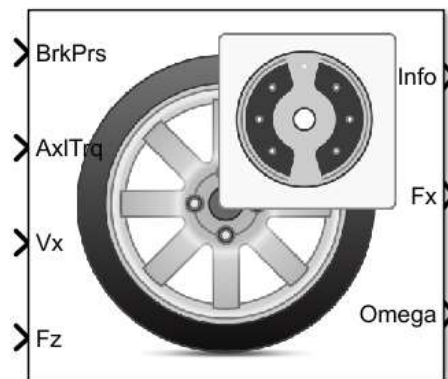
Vehicle Scenarios



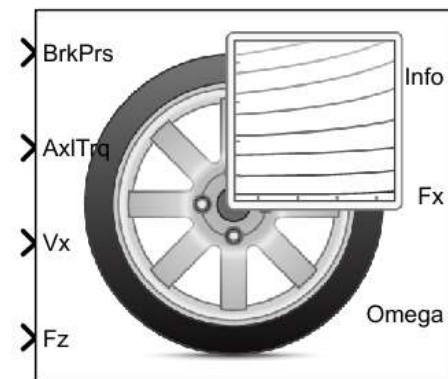
Longitudinal Wheel - No Brake



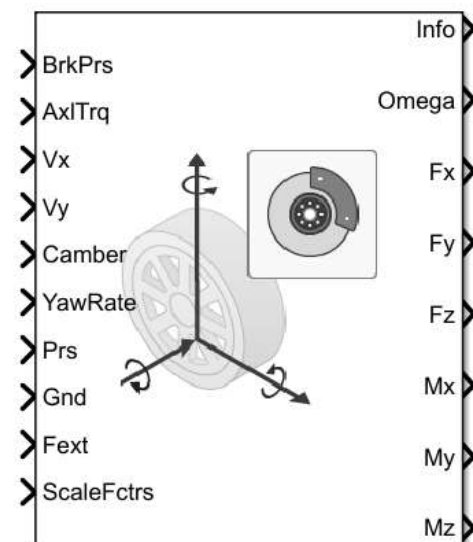
Longitudinal Wheel - Disc Brake



Longitudinal Wheel - Drum Brake

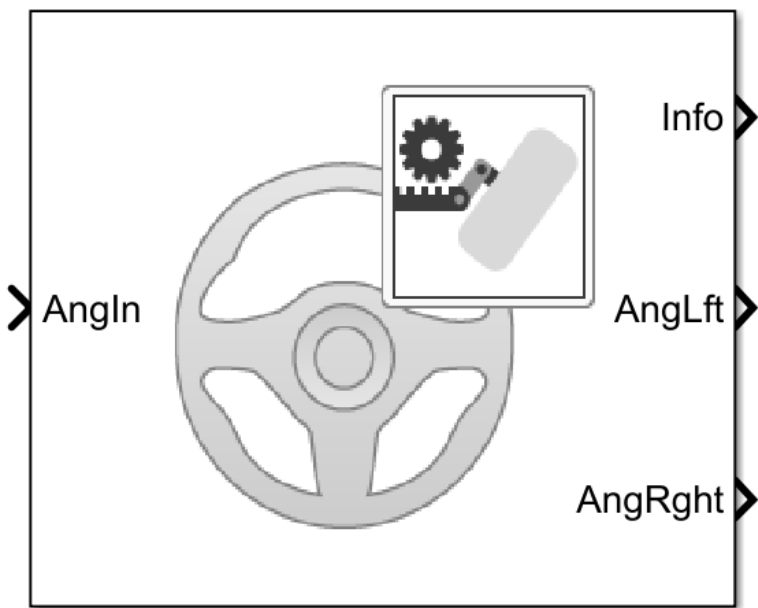
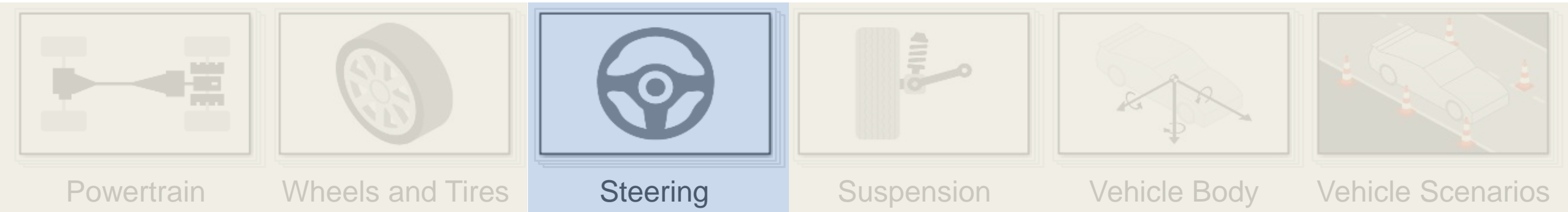


Longitudinal Wheel - Mapped Brake

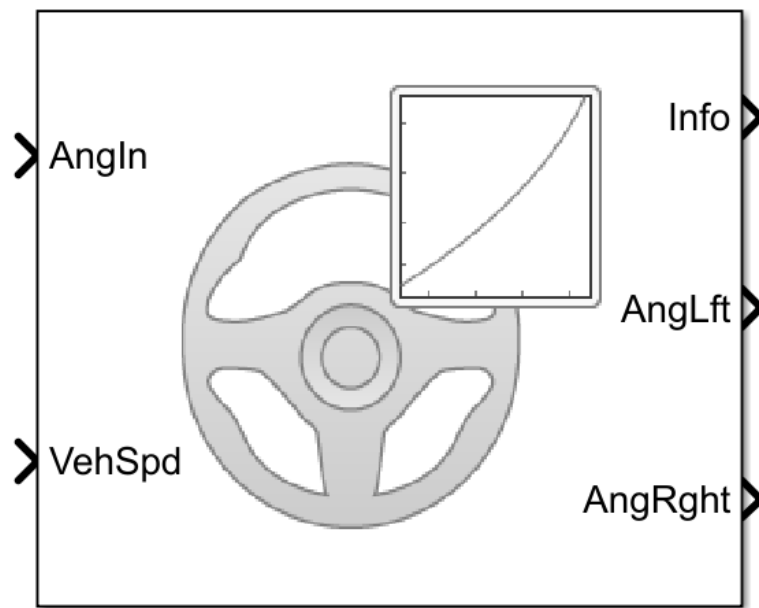


Combined Slip Wheel 2DOF

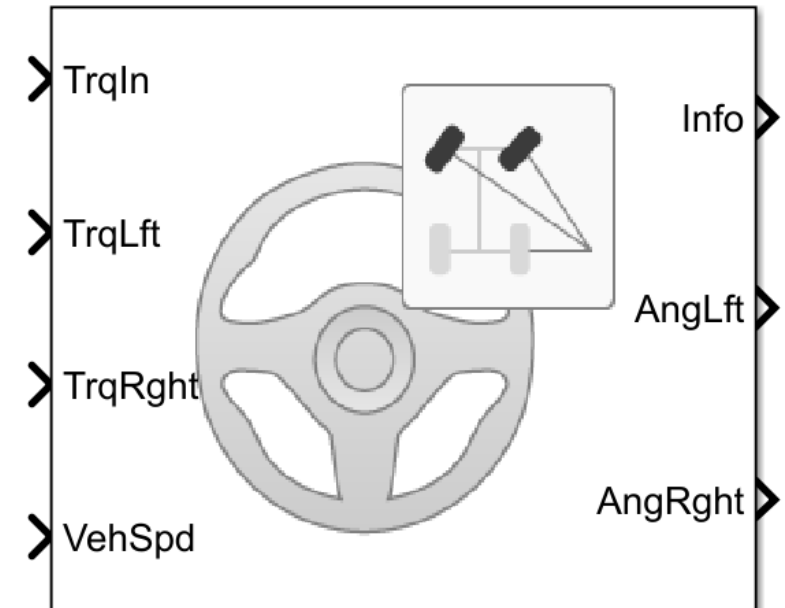
Block Library: Steering



Kinematic Steering



Mapped Steering



Dynamic Steering

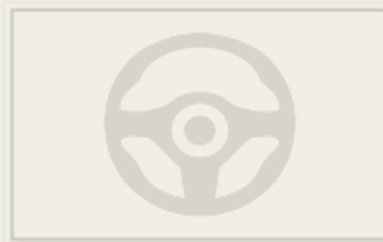
Block Library: Suspension



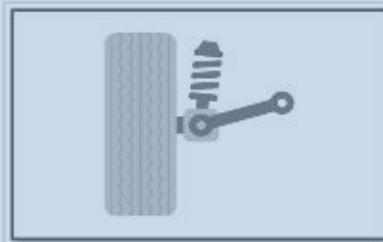
Powertrain



Wheels and Tires



Steering



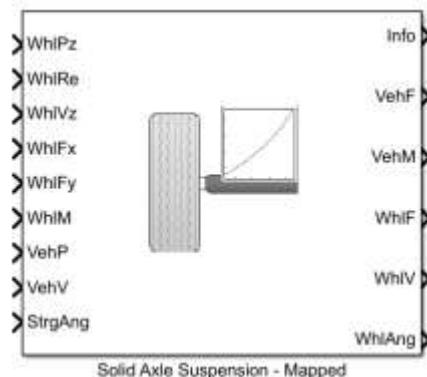
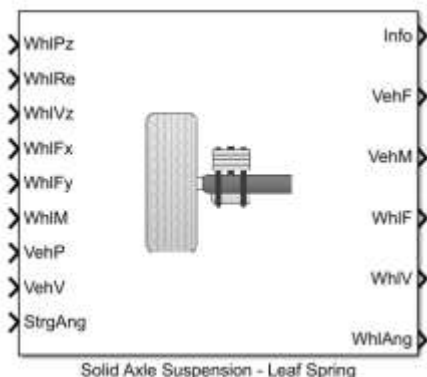
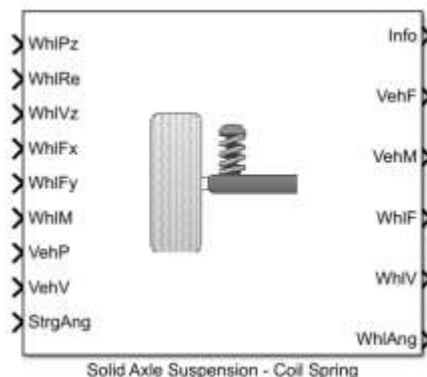
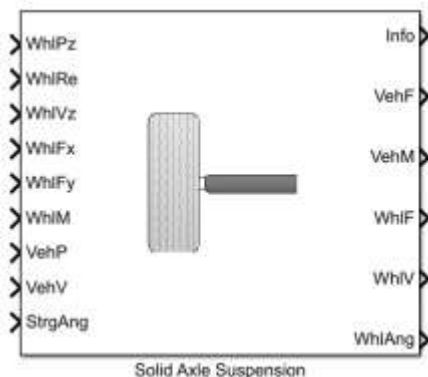
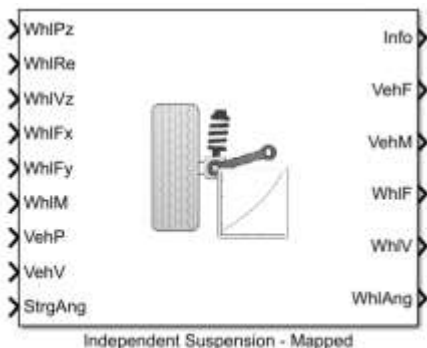
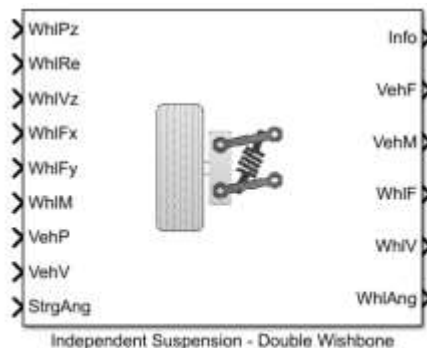
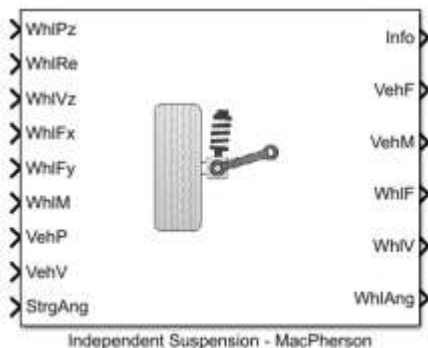
Suspension



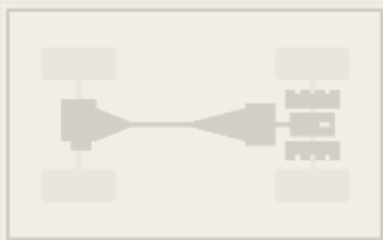
Vehicle Body



Vehicle Scenarios



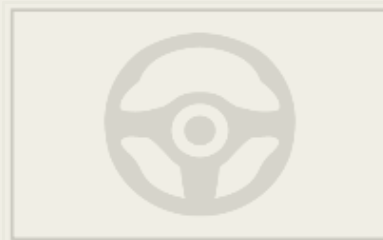
Block Library: Vehicle Body



Powertrain



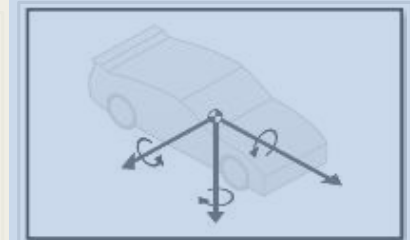
Wheels and Tires



Steering



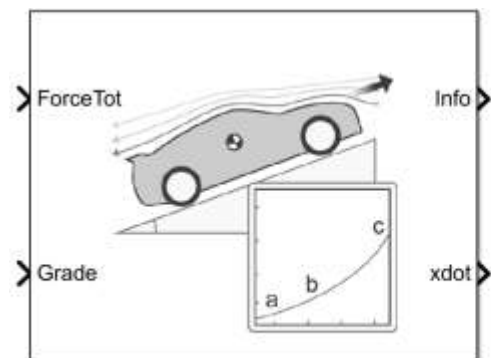
Suspension



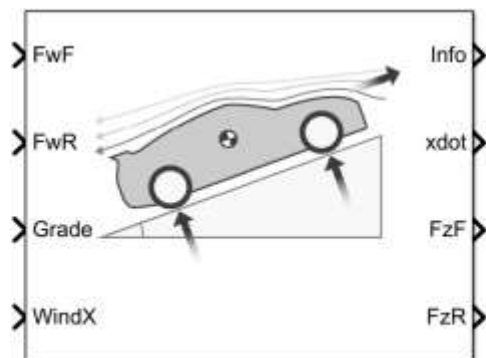
Vehicle Body



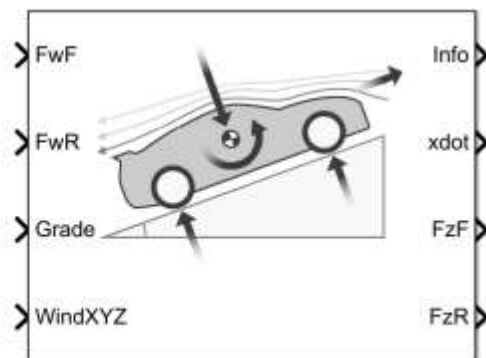
Vehicle Scenarios



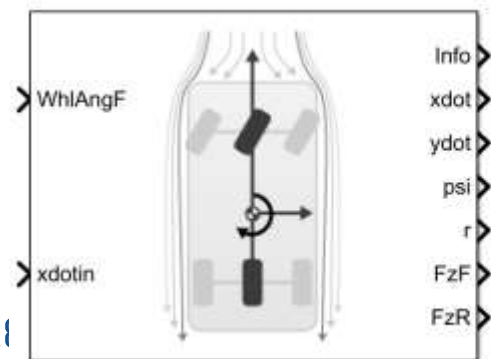
Vehicle Body Total Road Load



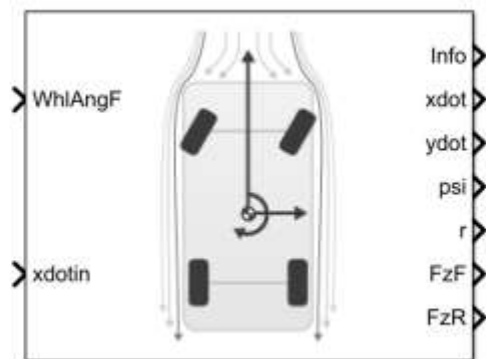
Vehicle Body 1DOF Longitudinal



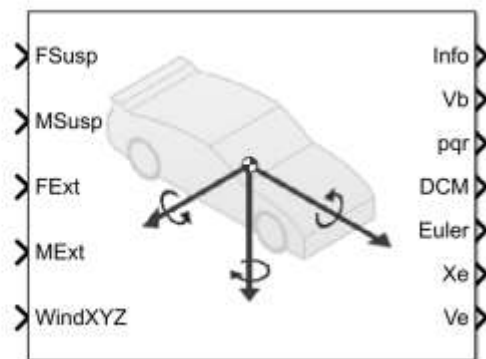
Vehicle Body 3DOF Longitudinal



Vehicle Body 3DOF Single Track

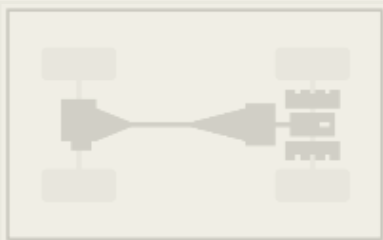


Vehicle Body 3DOF Dual Track



Vehicle Body 6DOF

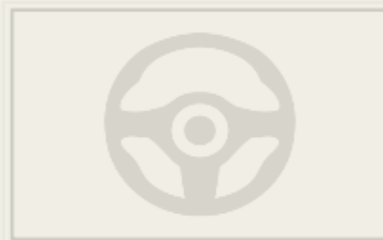
Block Library: Vehicle Scenarios



Powertrain



Wheels and Tires



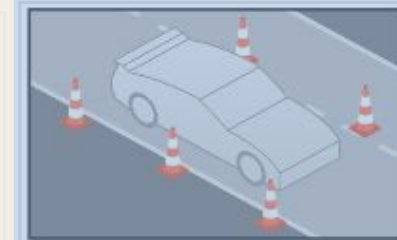
Steering



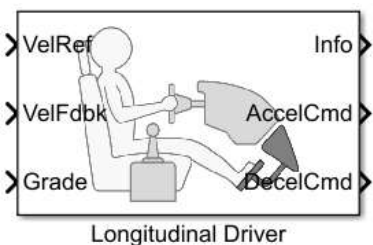
Suspension



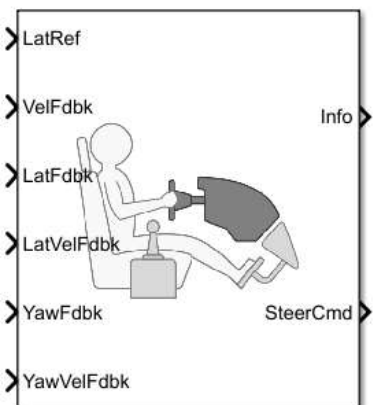
Vehicle Body



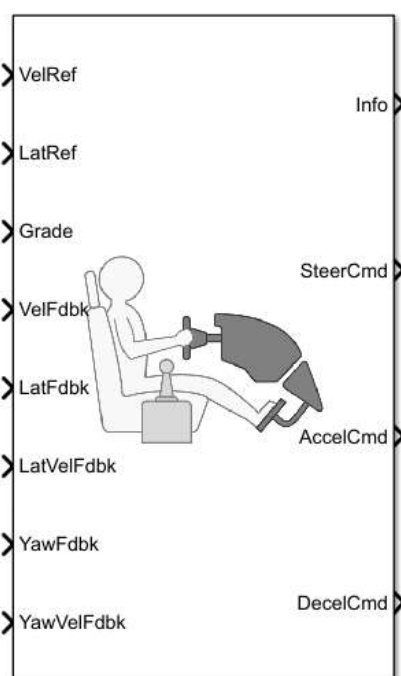
Vehicle Scenarios



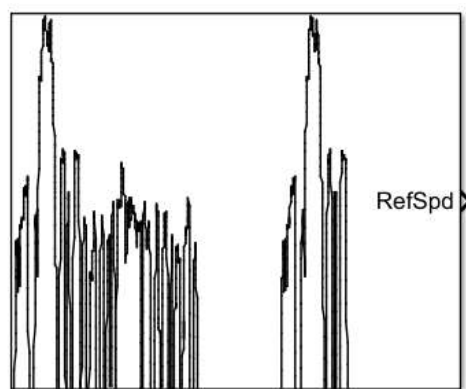
Longitudinal Driver



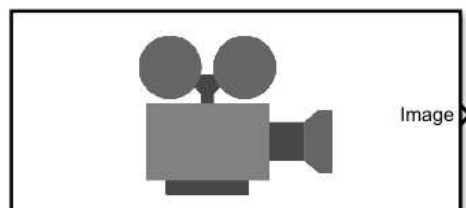
Lateral Driver



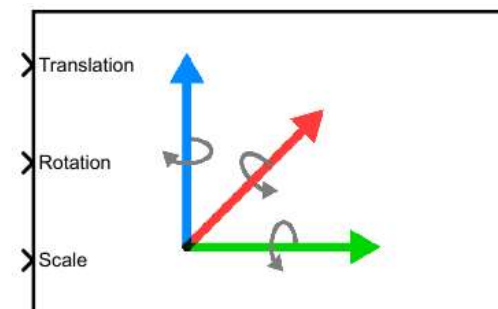
Predictive Driver



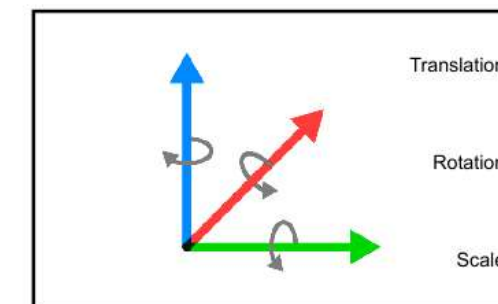
Drive Cycle Source
FTP75 (2474 seconds)



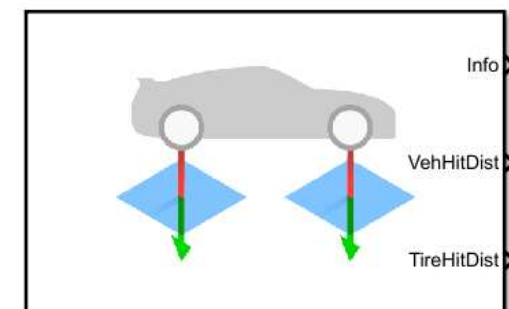
Simulation 3D Camera Get



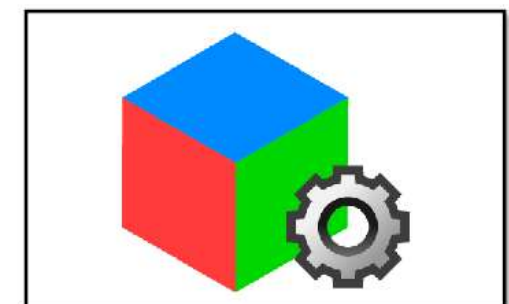
Simulation 3D Actor Transform Set



Simulation 3D Actor Transform Get



Vehicle Terrain Sensor



Simulation 3D Config

Game Engine Co-Simulation

Simulink

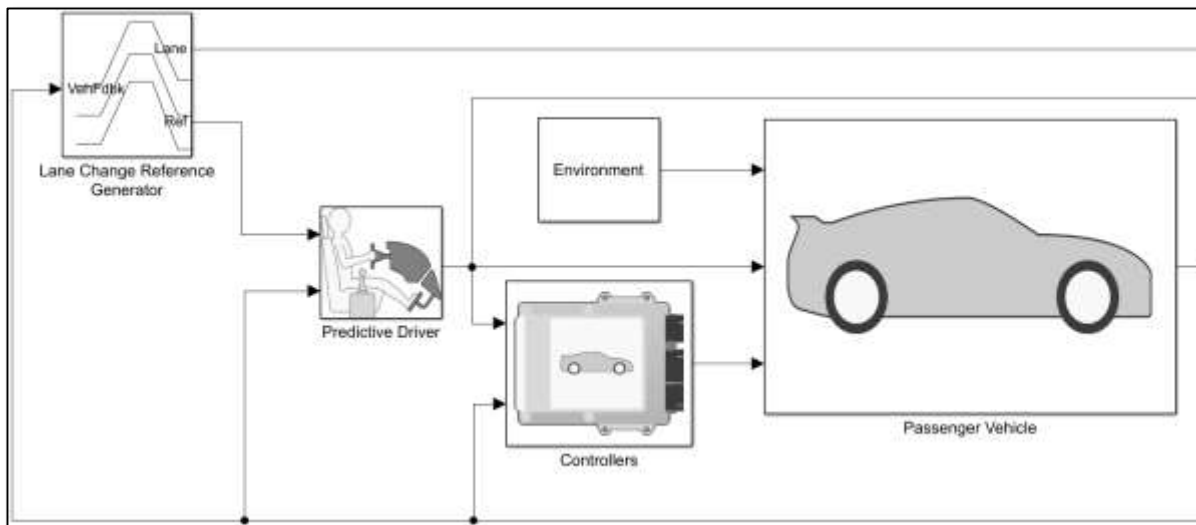
- Physics of vehicle
- Initialization of game engine camera

vehicle / camera location

Unreal Engine

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

camera image, ground height, ...

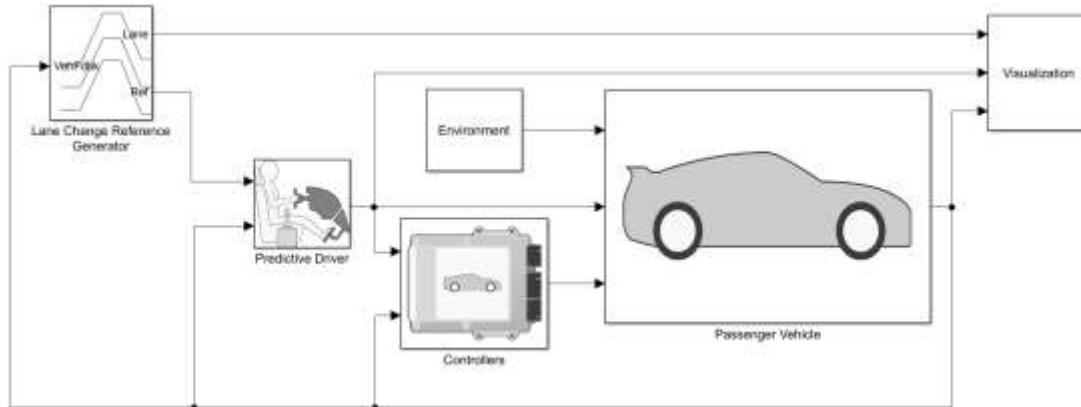


Reference Applications

Vehicle Maneuvers

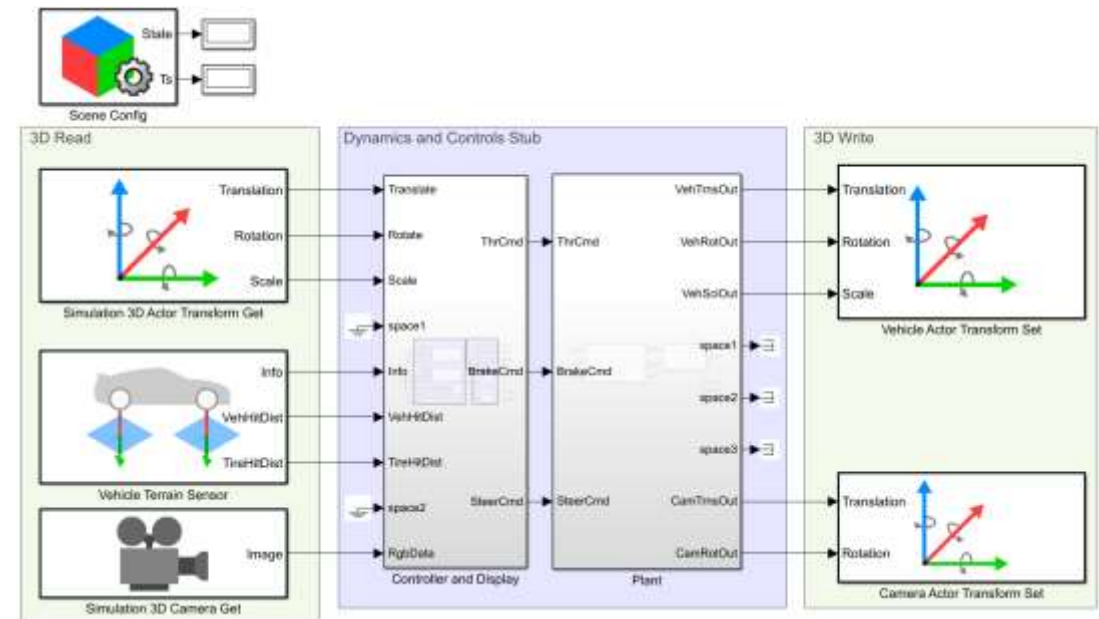
Analyze ride and handling on driving maneuvers such as:

- Double-lane change
- Swept sine steering
- Slowly increasing steering



Scene Interrogation

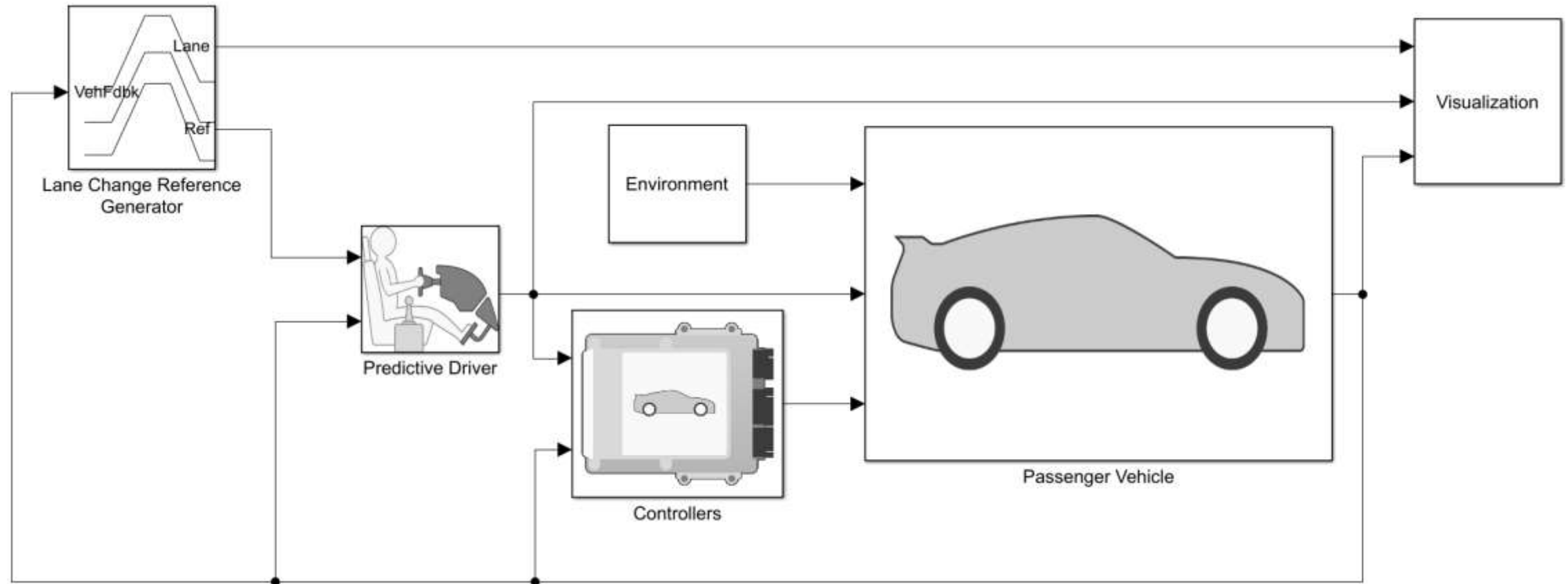
Configure the interface to the 3D environment



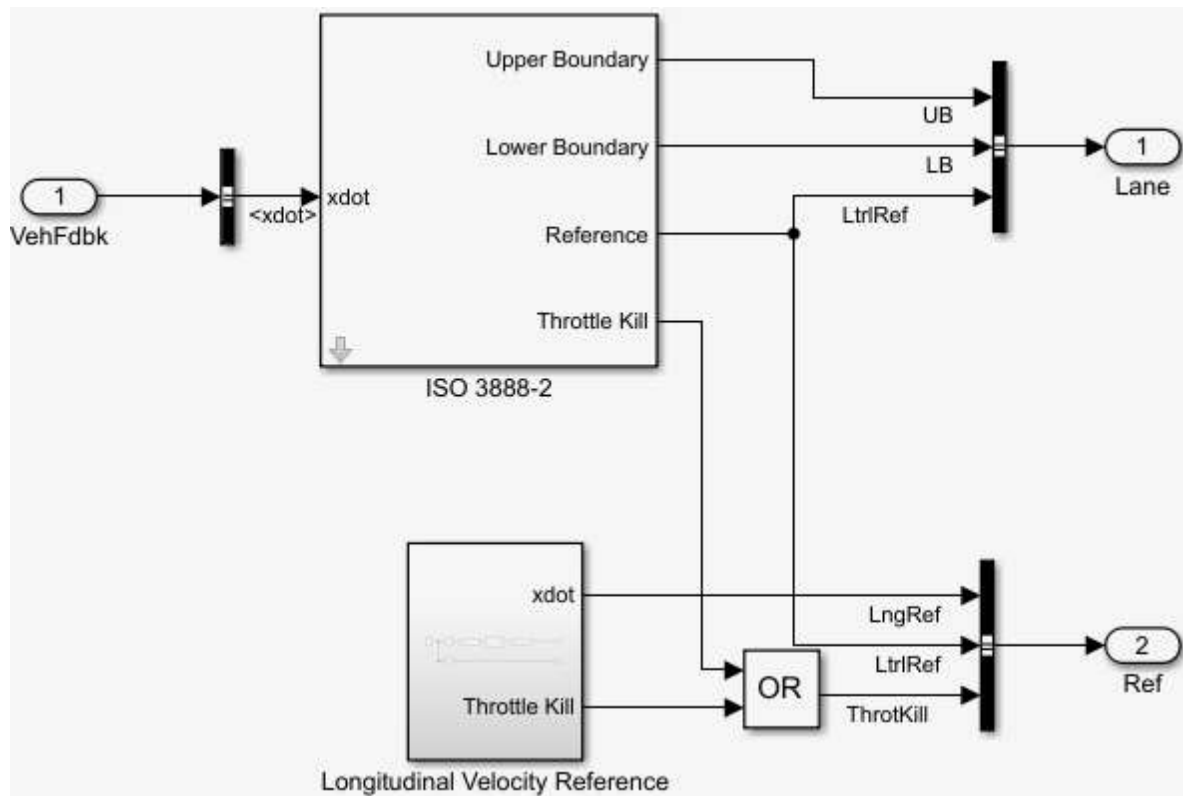
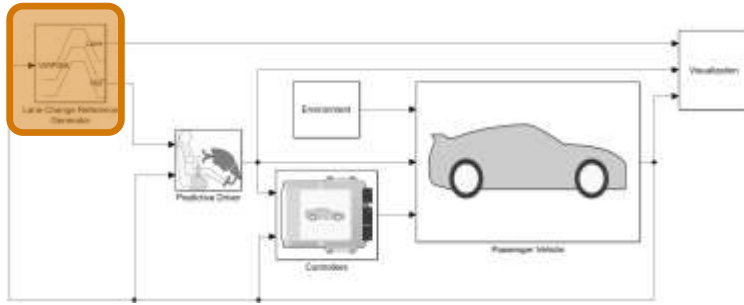
Agenda

- What is Vehicle Dynamics Blockset?
 - **How can I use it?**
 - Ride and handling analysis
 - Chassis controls development
 - ADAS / AD testing
 - Hardware-In-the-Loop Testing
- ➔ – Assess longitudinal / lateral dynamics

Reference Application: Double Lane Change



Reference Application: Double Lane Change (Maneuver)



Block Parameters: Lane Change Reference Generator

Double Lane Change Source (mask) (link)

Once the target longitudinal velocity is achieved, this block will command a zero acceleration signal and generate a lateral reference trajectory as a function of longitudinal displacement. Signals indicating the left and right lane boundaries are also generated as a function of the specified track width. An additional distance may be prescribed after the target longitudinal velocity prior to beginning the maneuver.

Parameters

Maneuver start time, t_{start} [s]:

Longitudinal entrance velocity setpoint, x_{dot_ref} :

Longitudinal entrance velocity setpoint units, $x_{dotUnit}$ []:

Distance after target speed to begin reference, x_{start} [m]:

Vehicle width, $vehW$ [m]:

Lateral offset, $latoff$ [m]:

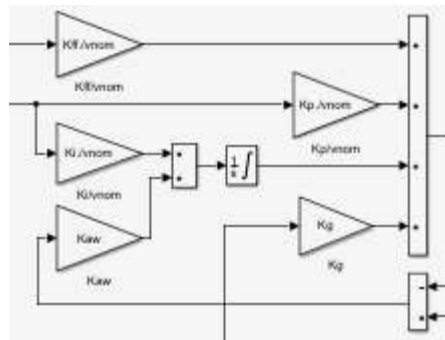
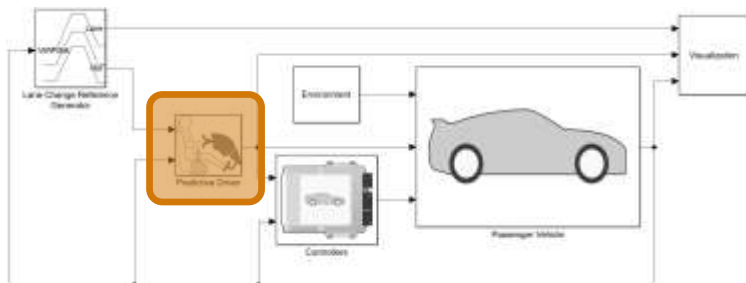
Use reference data

Lateral reference position breakpoints [m]:

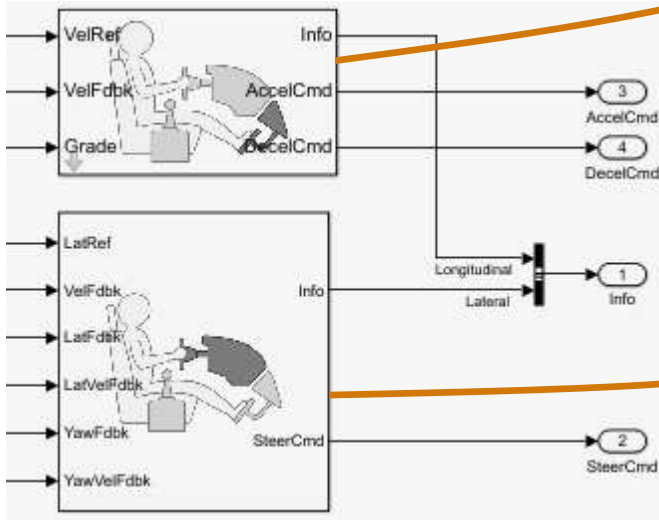
Lateral reference data, $latRef$ [m]:

Set target velocity and lateral position

Reference Application: Double Lane Change (Driver)



PI controller sets throttle / brake command



Previewed Path Input

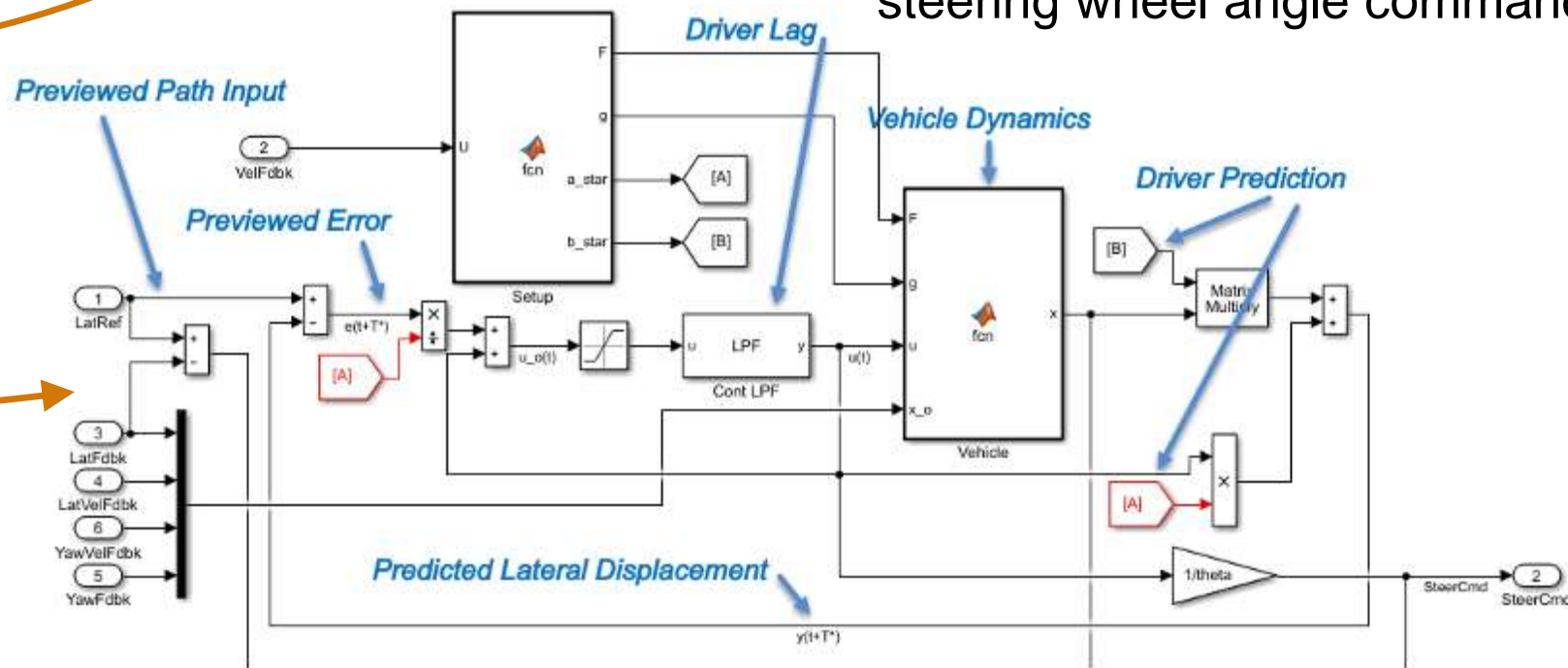
Previewed Error

Driver Lag

Vehicle Dynamics

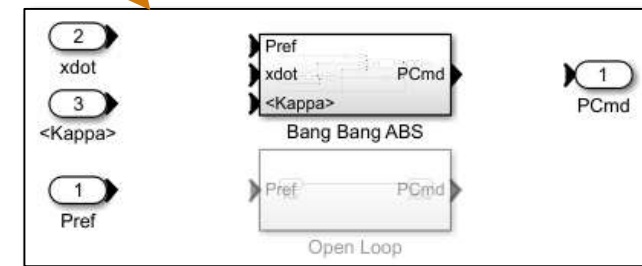
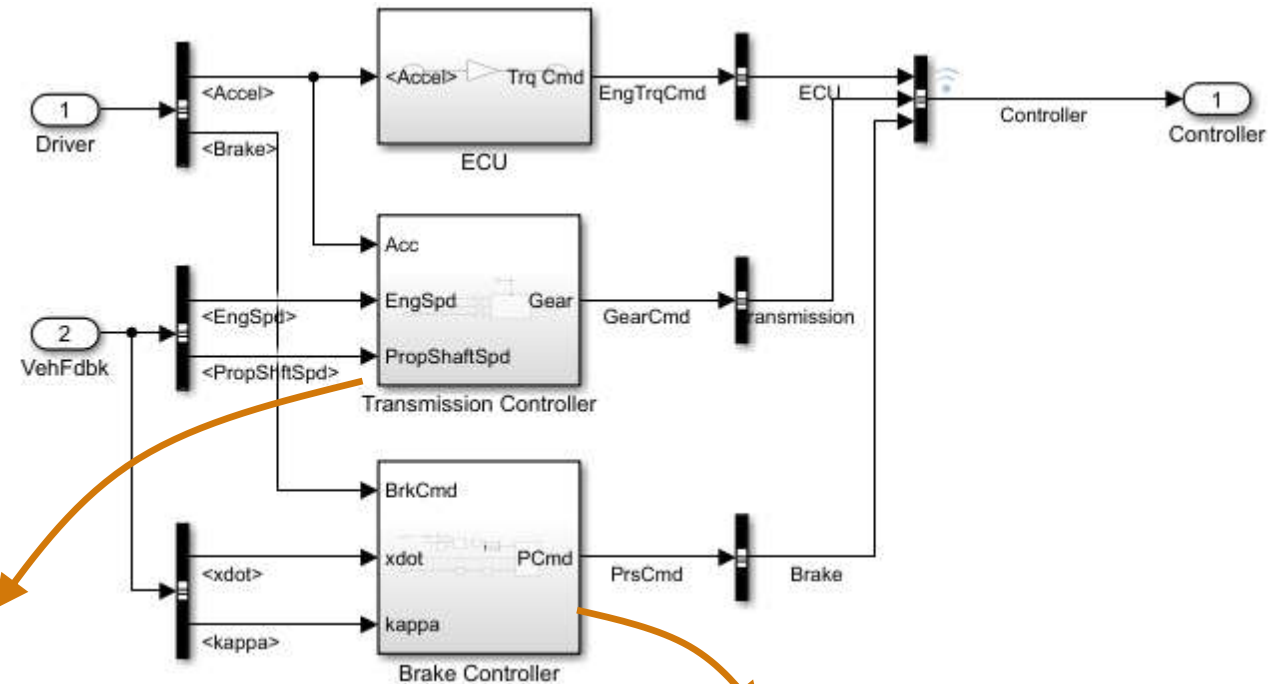
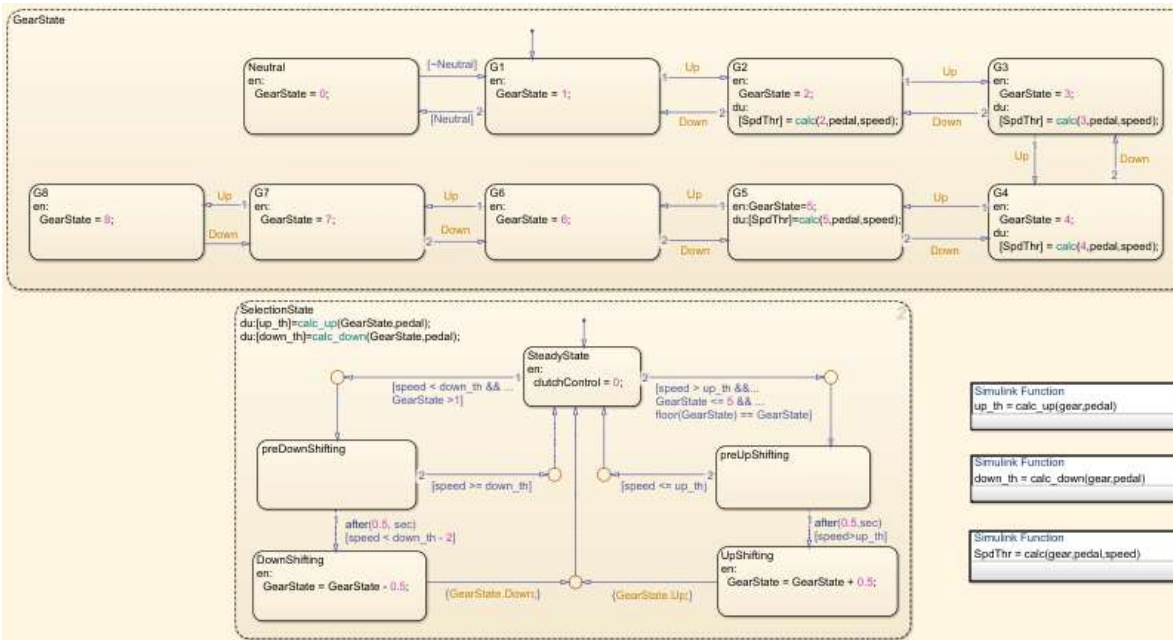
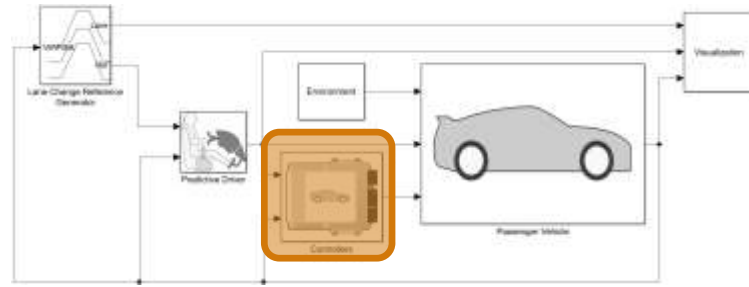
Driver Prediction

Predicted Lateral Displacement



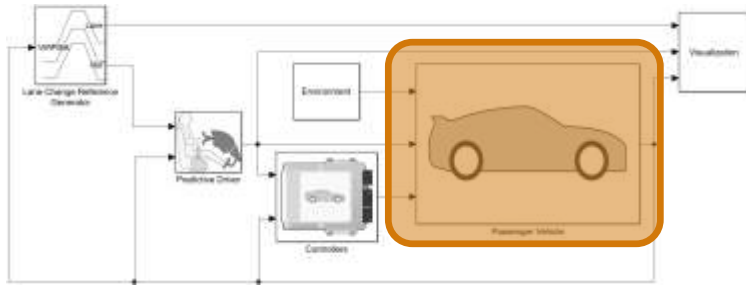
Predictive driver model sets steering wheel angle command

Reference Application: Double Lane Change (Controllers)

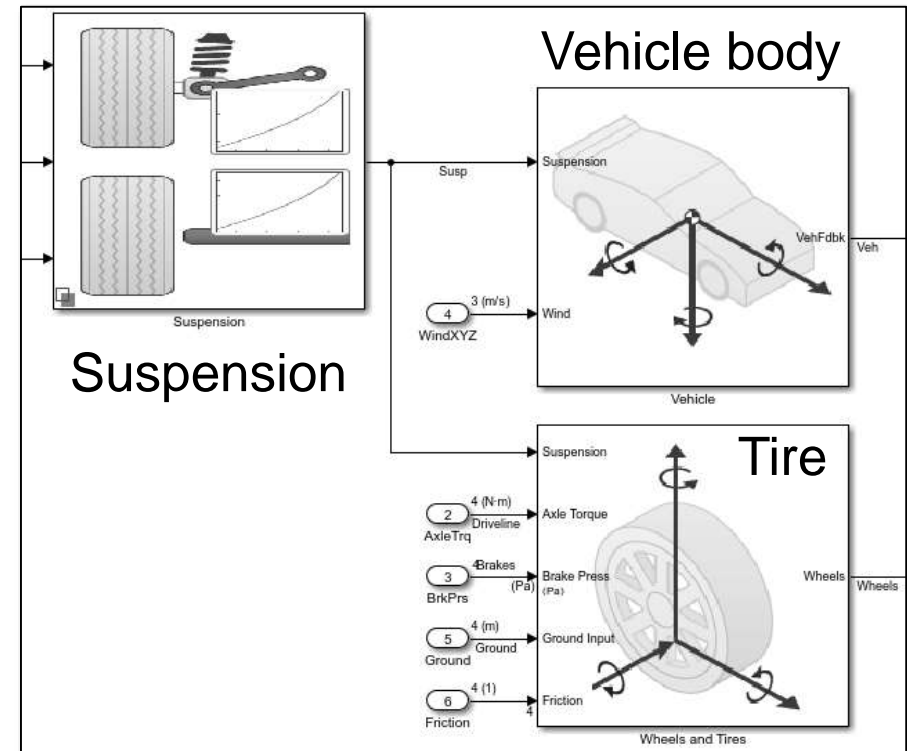
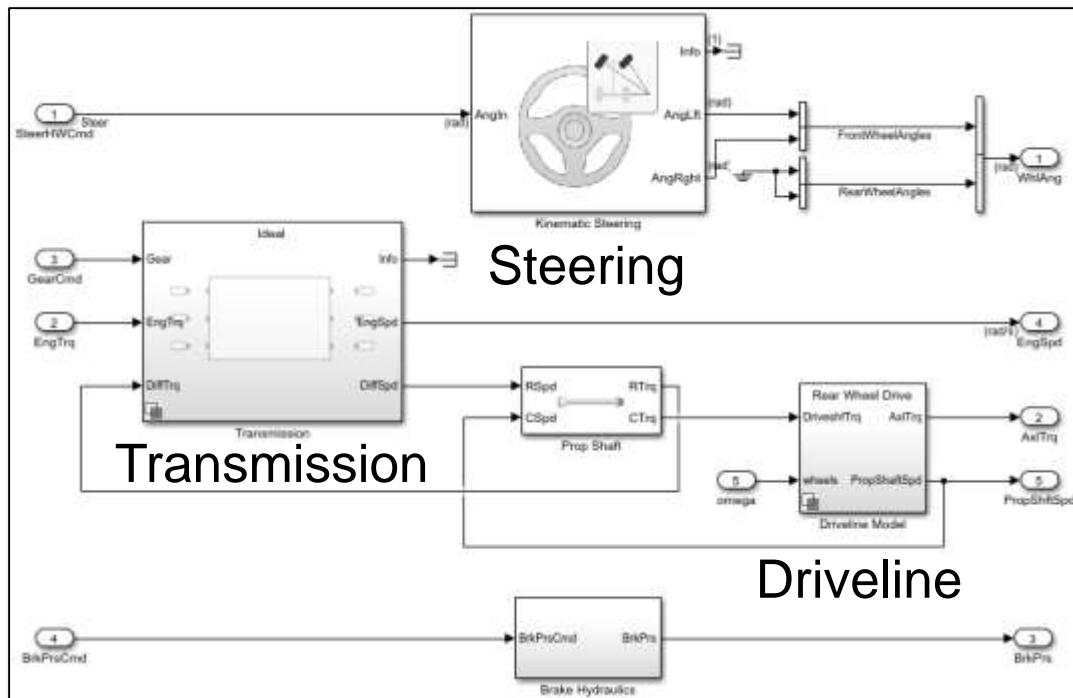
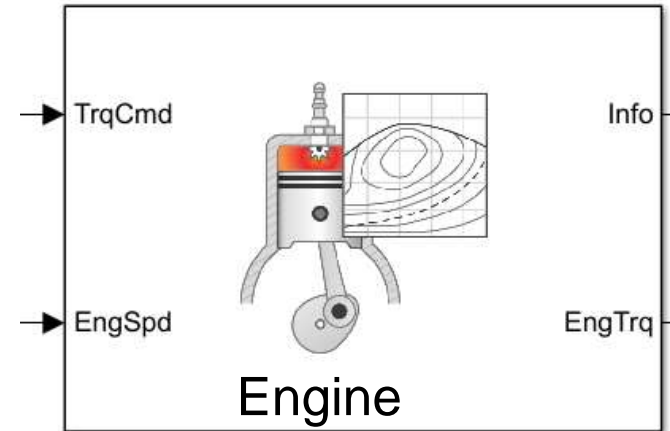


- Basic controllers provided for engine, transmission and brakes
- Incorporate your own variants, as needed

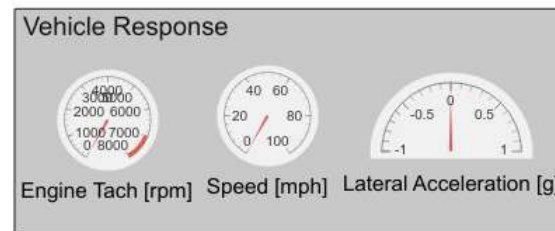
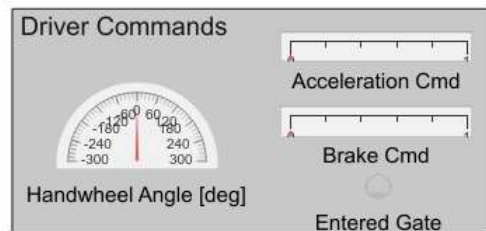
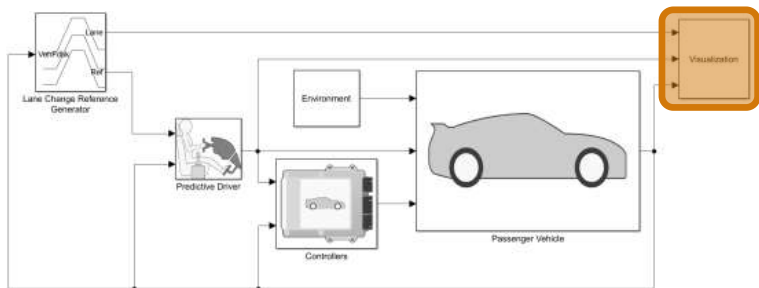
Reference Application: Double Lane Change (Plant)



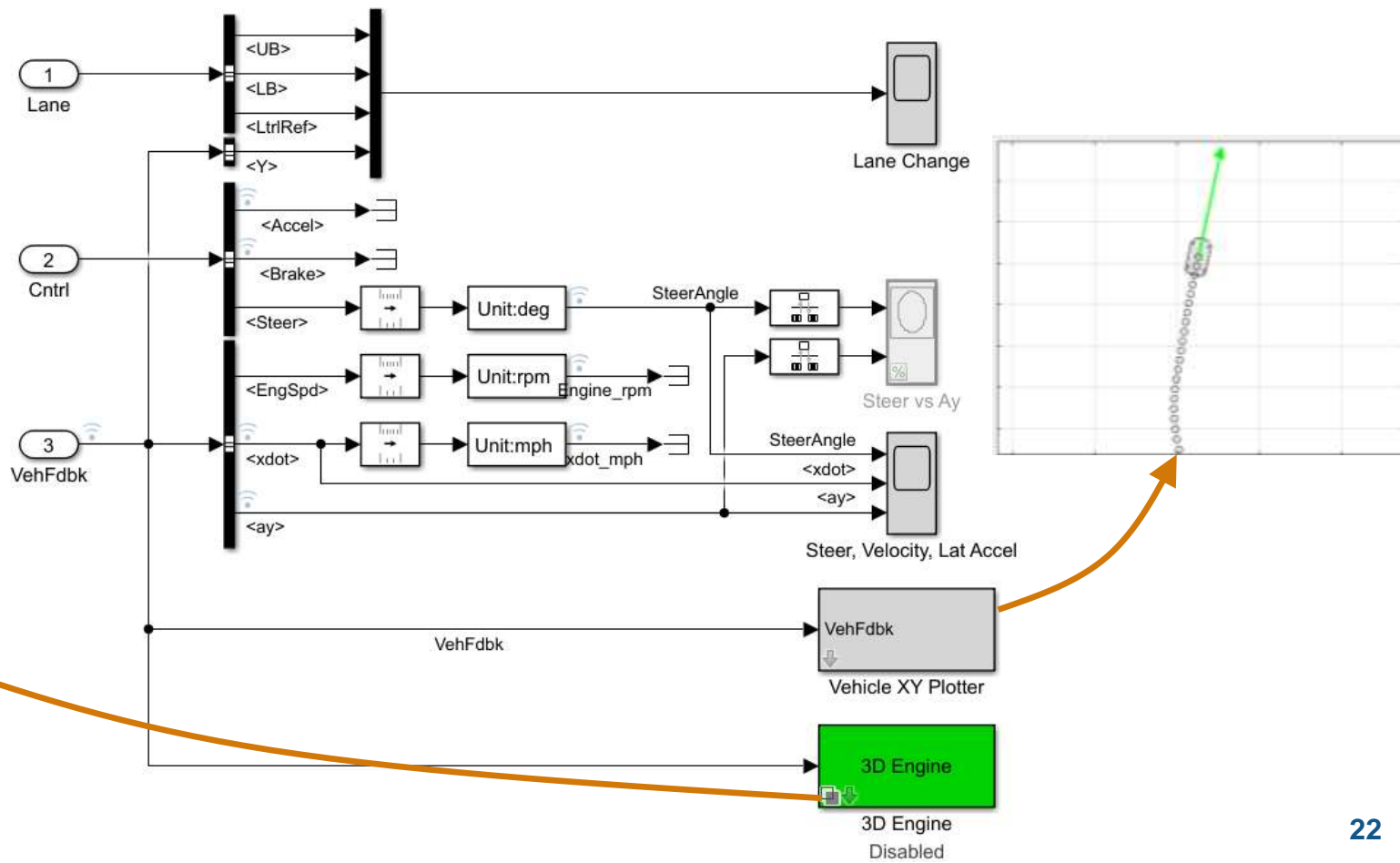
- Use default plant model provided
- Select variants of interest
- Customize subsystems



Reference Application: Double Lane Change (Visualization)



- Scopes, gauges, plotters, logs
- 3D engine interface



Ride and Handling Study: Double Lane Change



At 30 mph

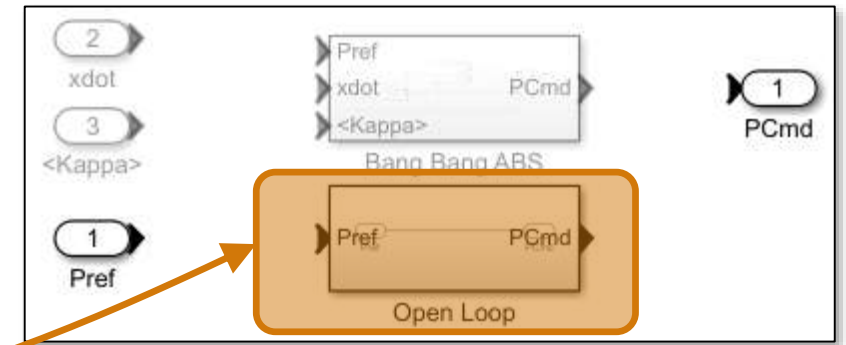
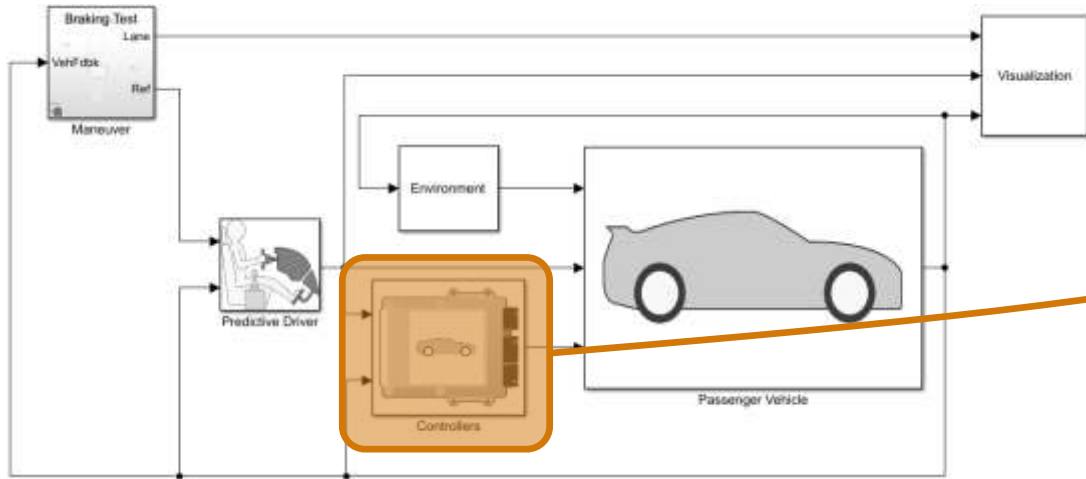


At 50 mph

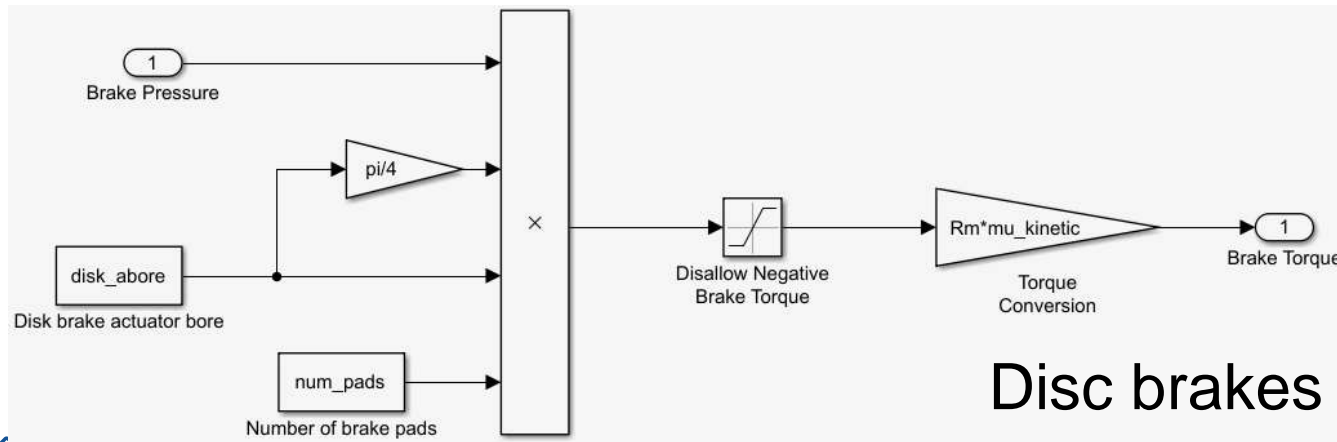
Agenda

- What is Vehicle Dynamics Blockset?
 - **How can I use it?**
 - Ride and handling analysis
 - Chassis controls development
 - ADAS / AD testing
 - Hardware-In-the-Loop Testing
- ➔ – Perform closed-loop testing

Chassis Controls Study: Braking Test

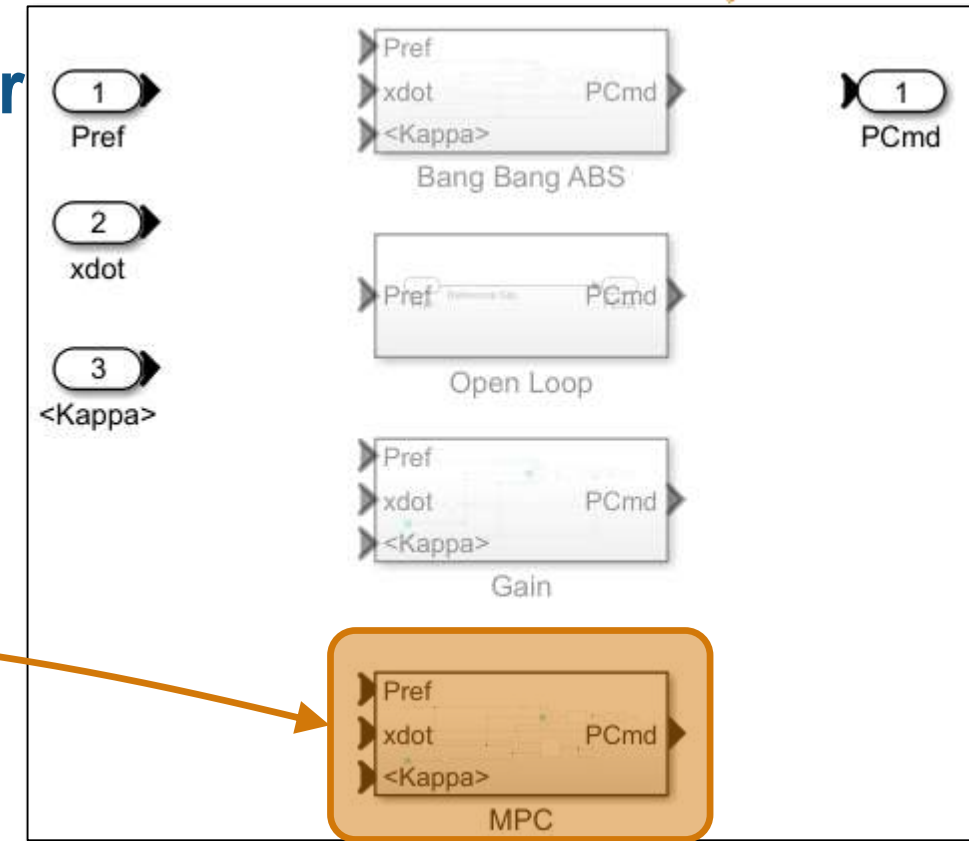
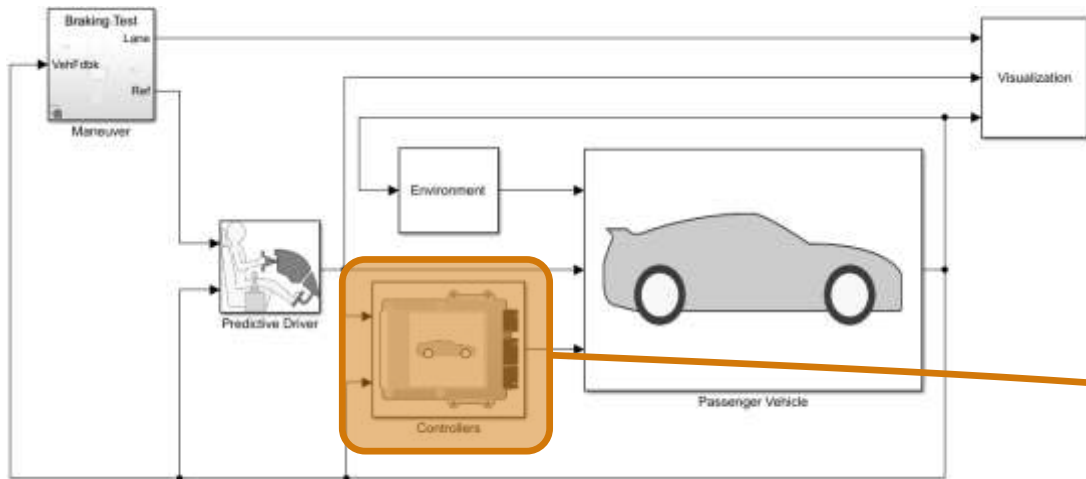


Open loop brake controller simply passes through brake pressure command

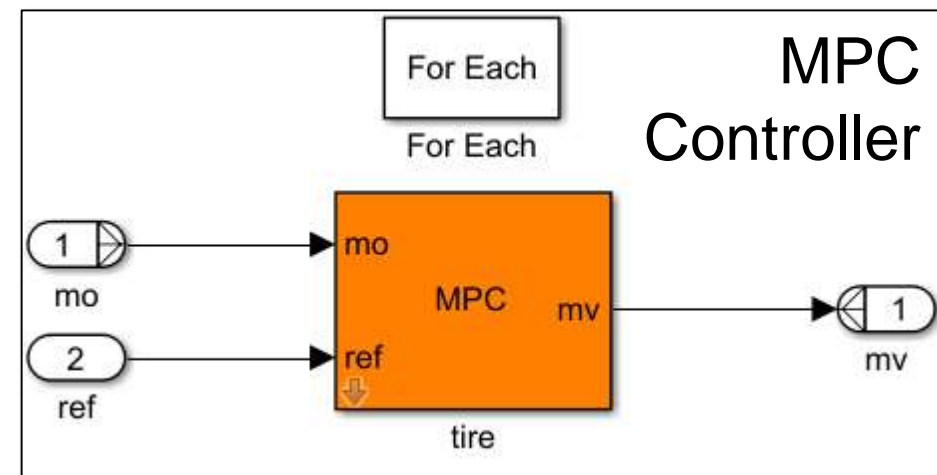


Disc brakes

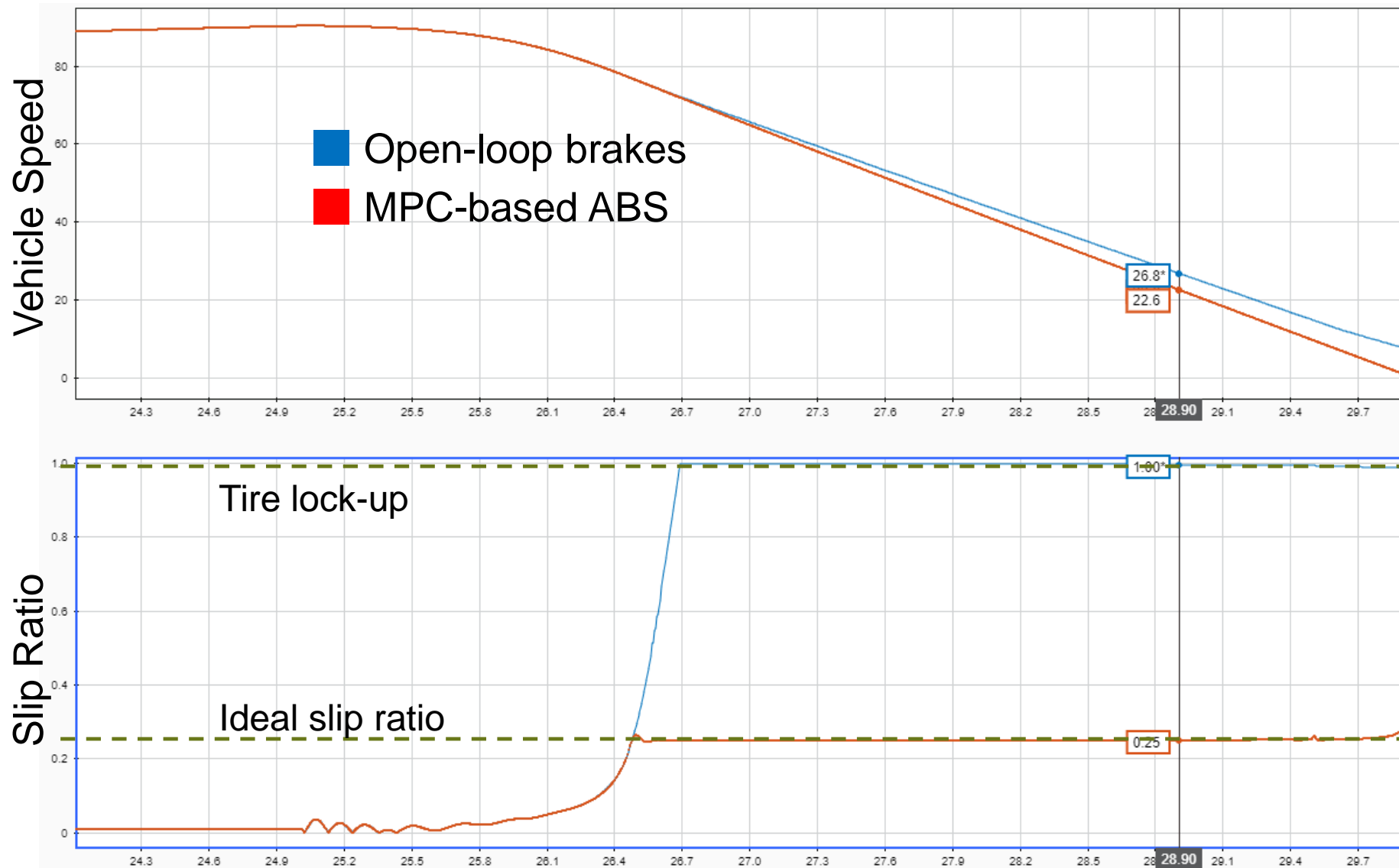
Chassis Controls Study: ABS Controller



- Added custom MPC variant to brake controller subsystem
- At each time step, finds optimal brake pressure for target slip ratio



Chassis Controls Study: ABS Controller

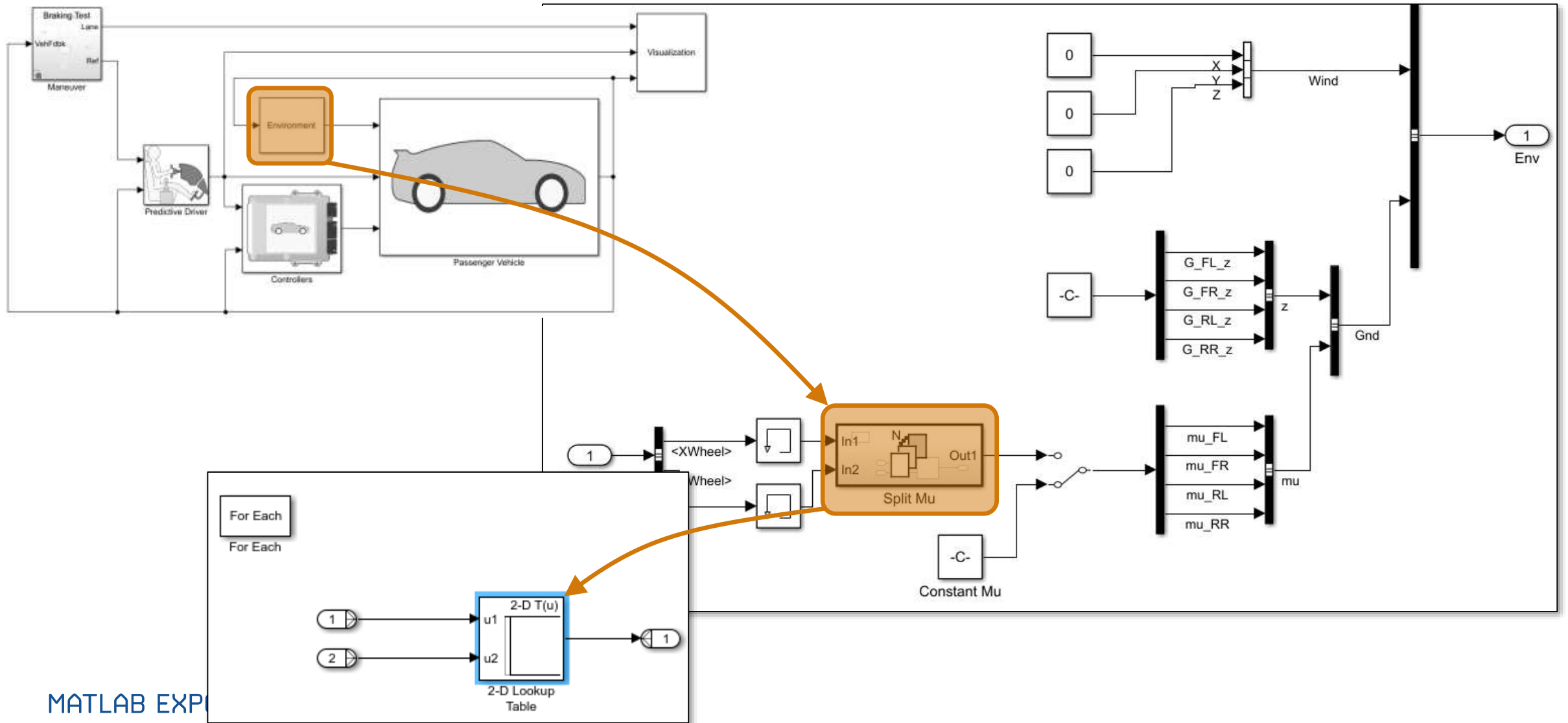


Chassis Controls Study: Braking Test

Green: Open-loop brake, white : with ABS




Split Mu Test



Chassis Controls Study: Split Mu Test



Agenda

- What is Vehicle Dynamics Blockset?
 - **How can I use it?**
 - Ride and handling analysis
 - Chassis controls development
 - ADAS / AD testing
 - Hardware-In-the-Loop Testing
-  – Test in a virtual 3D environment

ADAS / AD Testing: Virtual 3D Scene



Camera sensor sends video to Simulink

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

Stop Sign Detection and Braking

The image shows a Simulink simulation environment for a stop sign detection and braking system. The main workspace contains two primary blocks: a 'Controller' and a 'Stop Sign Detector'.

Controller Block: This block processes inputs to generate control signals. It features several gain blocks: a gain of 0.5 for throttle control, a gain of 0 for steering control, and a gain of 10 for another control signal. A switch block selects between 'Straight', 'Left', and 'Right' steering commands. The outputs are labeled 'ThrCmd' and 'SteerCmd'. A 'Isnan' block is set to 'False'.

Stop Sign Detector Block: This block handles image processing. It starts with 'Convert Image to double', followed by an 'img StopSignDetector' block that outputs 'bbox' and 'score'. The 'bbox' output is used by an 'Image Draw Rectangles' block, which also receives 'Pts' input. The final output is 'Image To Video Display'.

Viewer Scope: A plot window showing a signal that remains at 0 until approximately t=4.1, where it abruptly jumps to 1. The x-axis ranges from 0 to 4.5, and the y-axis ranges from 0 to 1. The status bar indicates 'Running' and 'Sample based T=4.170'.

To Video Display: A 3D simulation window showing a car driving on a road towards a stop sign. The status bar at the bottom indicates 'Running', 't=4.200', and 'FixedStepDiscrete'.

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](#)”



MathWorks® Products Solutions Academia Support Community Events

File Exchange

Search File Exchange

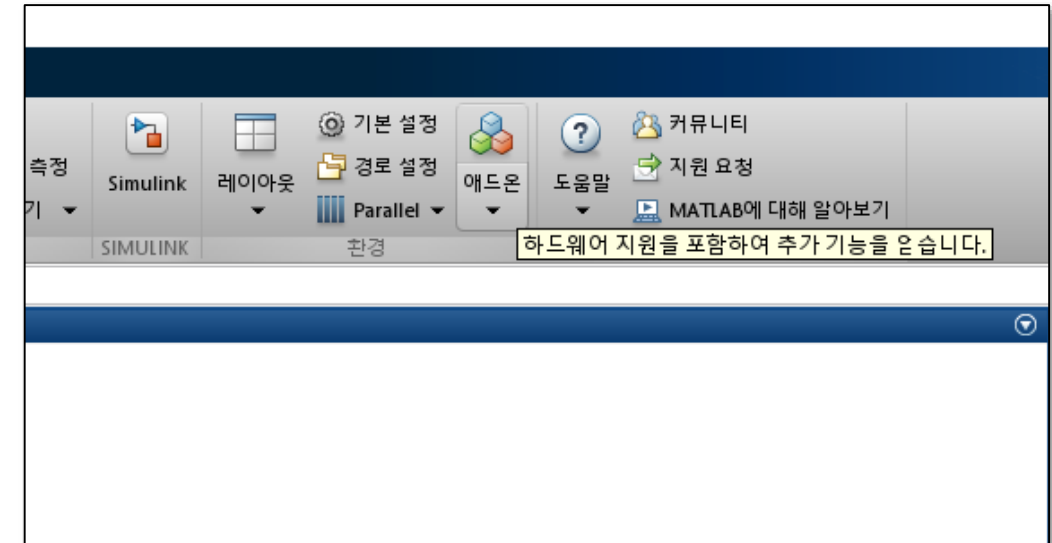
MATLAB Central ▾ Files Authors Tags Comments My File Exchange Submit About



Vehicle Dynamics Blockset interface for Unreal Engine 4

version 1.0 (15.1 KB) by MathWorks Automotive Community Profile

Simulink integration for Unreal Engine 4

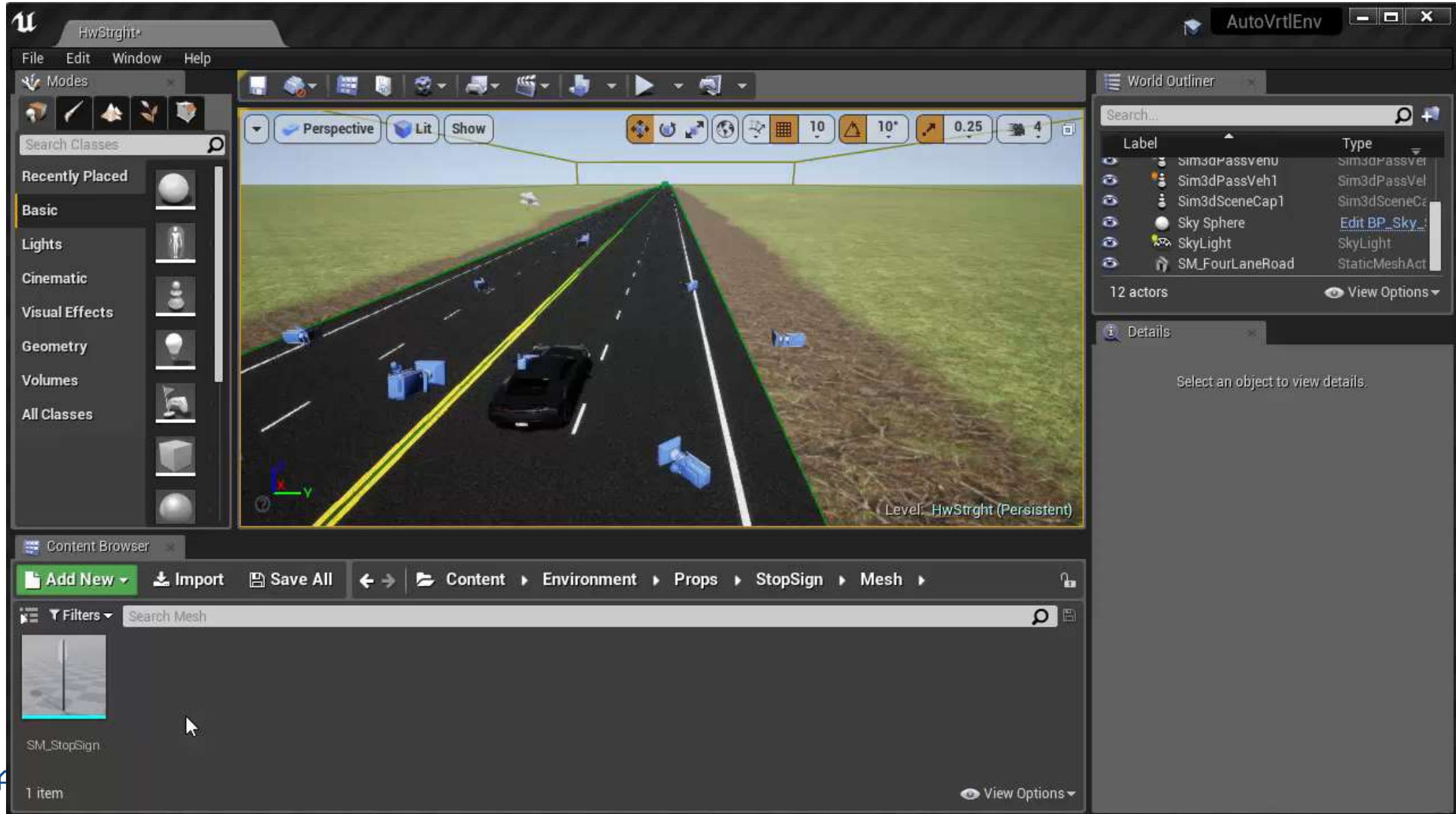


Simulink 레이아웃 기본 설정 경로 설정 애드온 도움말 커뮤니티 지원 요청

SIMULINK 환경

하드웨어 지원을 포함하여 추가 기능을 은습니다.

Editing Support Package Scene to Add Stop Sign



Changing the Lighting to Night Conditions

The screenshot displays the Simulink environment for a project named "SCRRReferenceApplication/Controller and Display". The main workspace shows a Simulink block diagram with two primary sections:

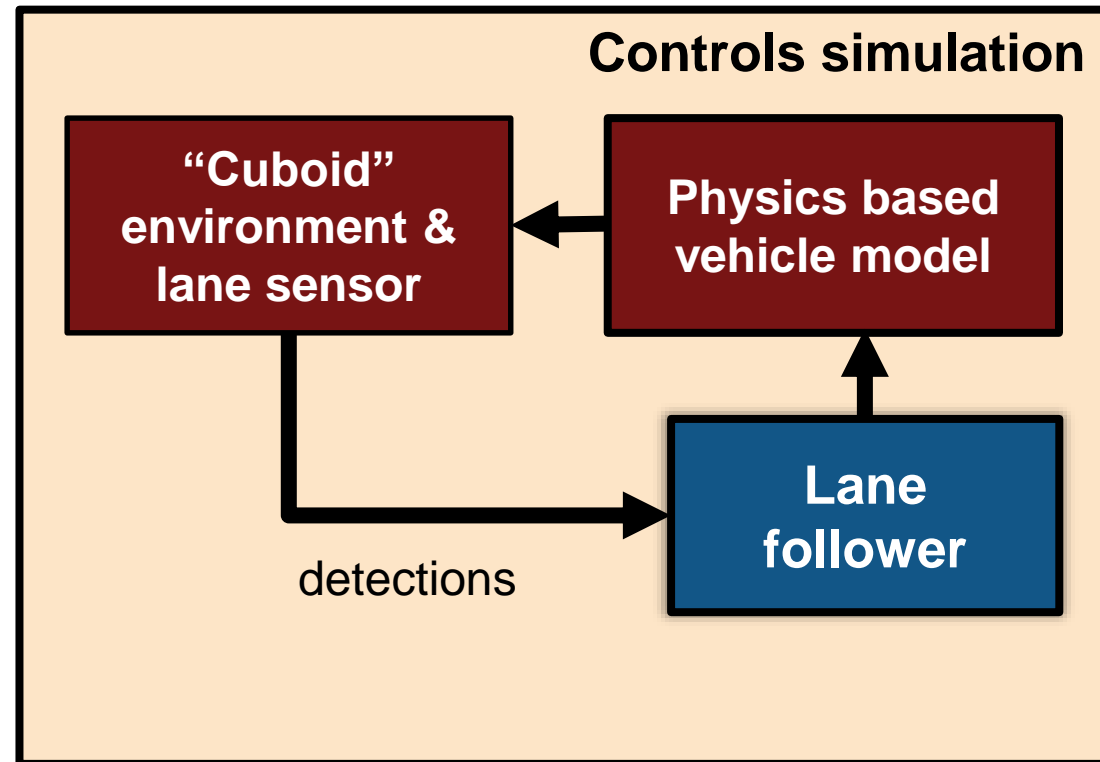
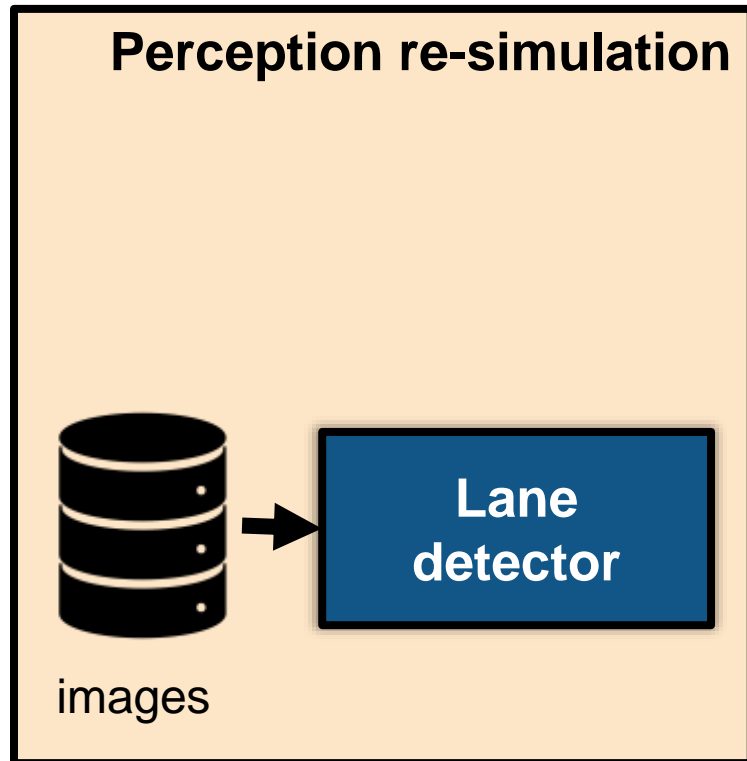
- Controller:** This section contains several gain blocks (0.5, 0, 10) and a switch block. It receives inputs from "single(1)", "single(0)", and "single(-0.1)" blocks. The switch block is controlled by "Left" and "Right" inputs. The controller outputs "ThrCmd" and "SteerCmd".
- Stop Sign Detector:** This section is highlighted with a blue border. It takes an image input and processes it through "Convert Image to double", "img StopSignDetector", and "Image Draw Rectangles" blocks. The "img StopSignDetector" block outputs a "score" and a "bbox". The "Image Draw Rectangles" block outputs "Pts". The final output is "Image To Video Display".

Two windows are overlaid on the Simulink interface:

- Viewer Scope:** A plot window showing a signal that is constant at 1.0 until approximately 4.2 seconds, where it drops sharply to 0.95. The x-axis ranges from 0 to 4.5, and the y-axis ranges from 0.9 to 1.1.
- To Video Display:** A video window showing a first-person perspective of a vehicle at night. A red stop sign is visible on the right side of the road. The video is being displayed at 80% zoom.

At the bottom of the Simulink window, the status bar indicates "Running" and "FixedStepDiscrete".

Perception algorithms are typically developed with different workflows than control algorithms



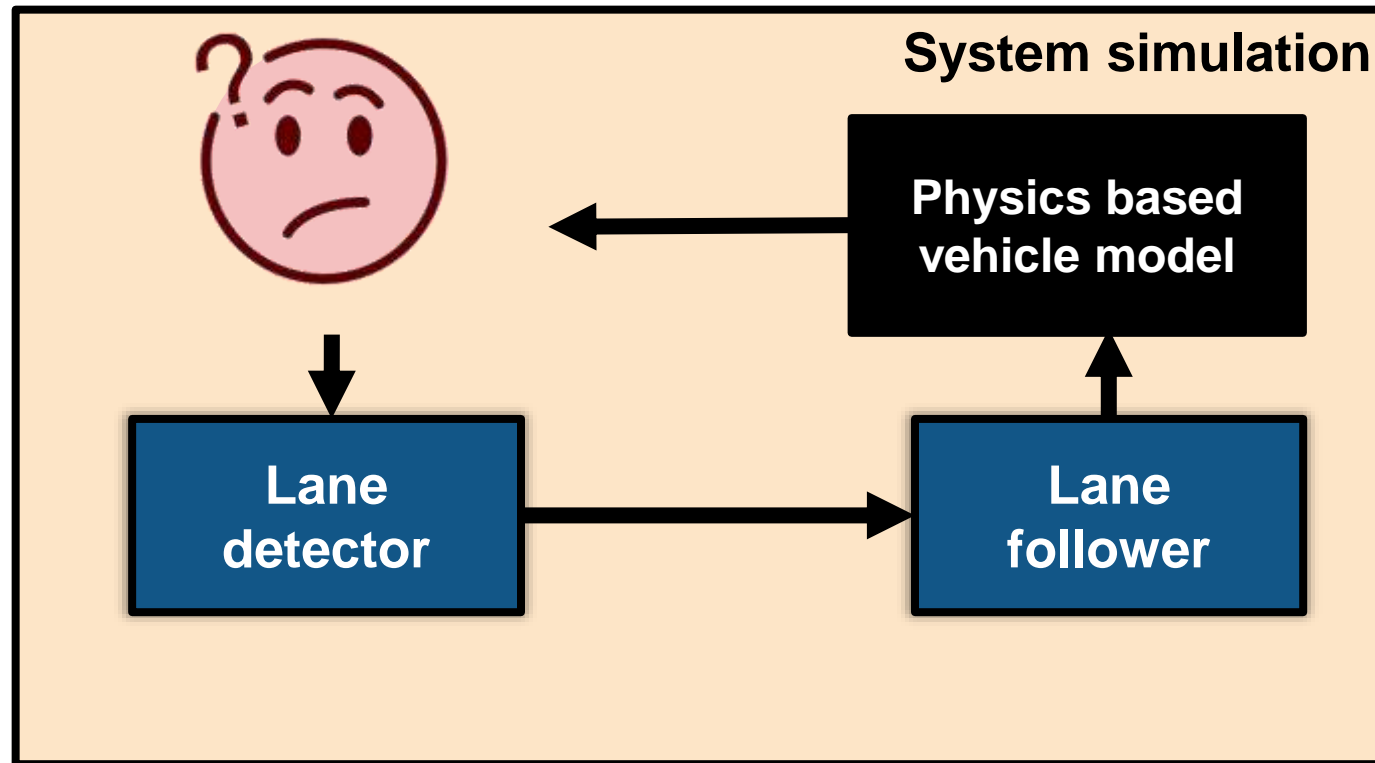
Perception
engineer

MATLAB EXPO 2018



Controls
engineer

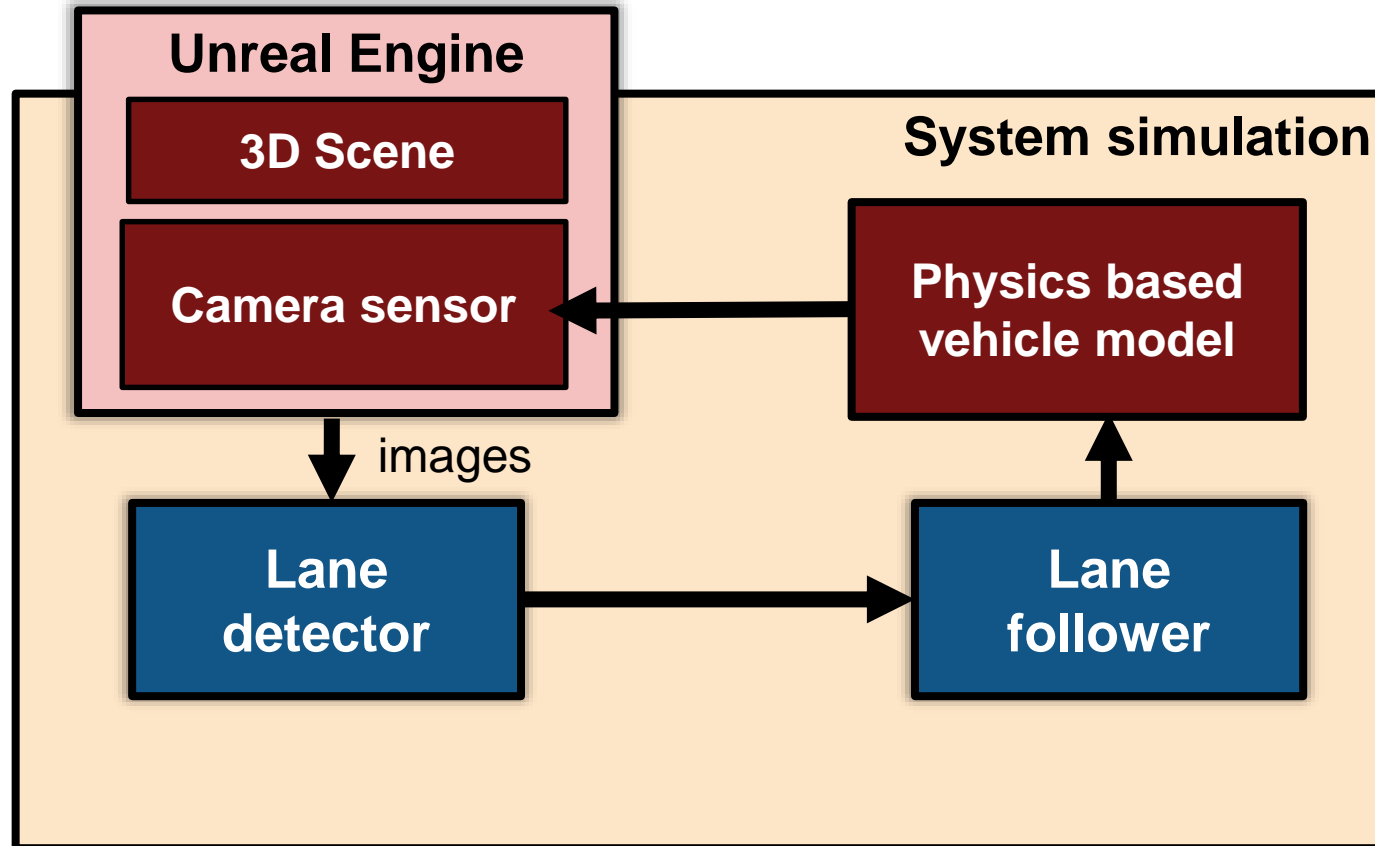
What is required to combine lane detector and follower components into a system level simulation



Systems engineer



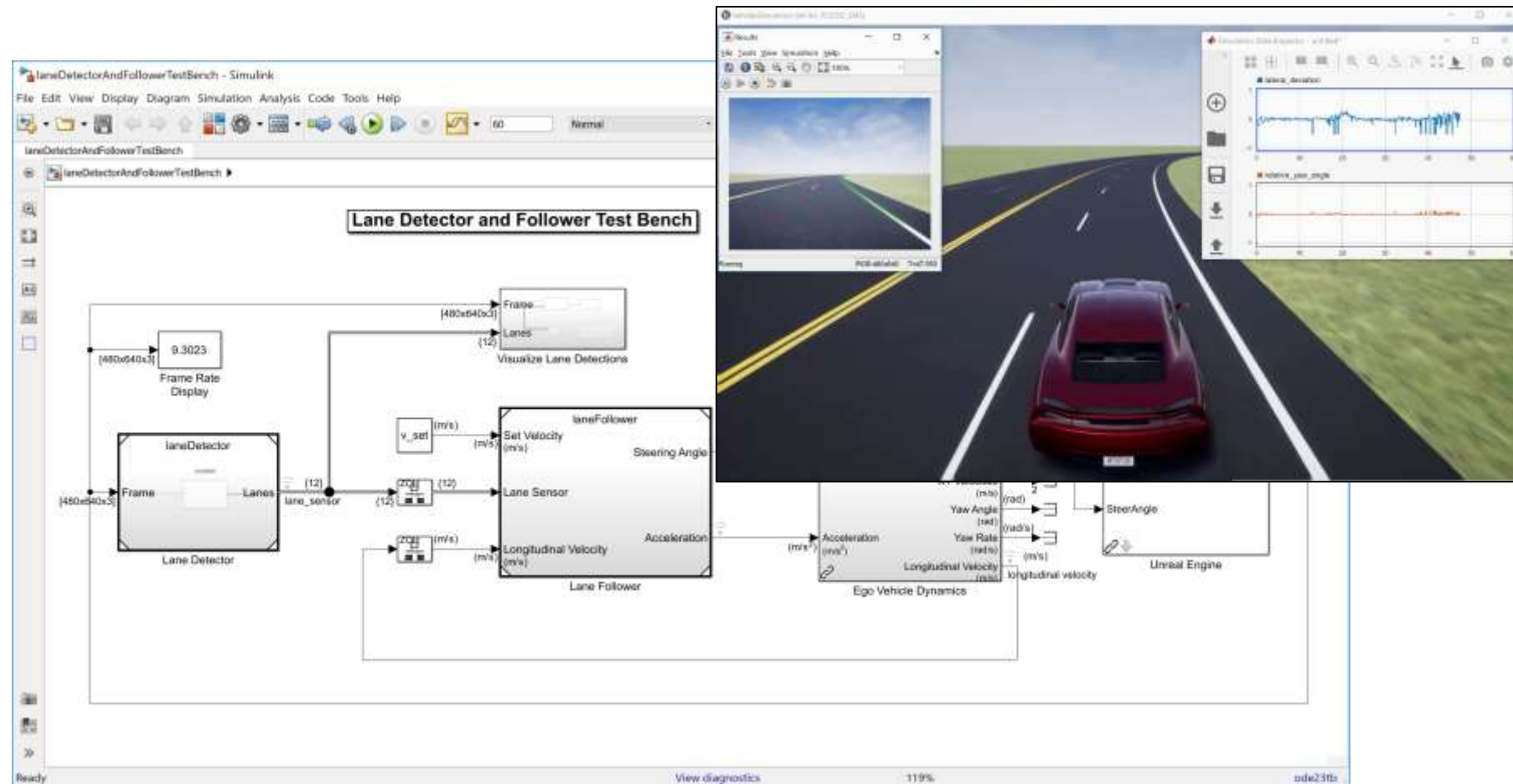
Lane detector and follower system



Systems
engineer



Lane detector and follower system test bench simulates vehicle dynamics with Unreal Engine to synthesize camera images

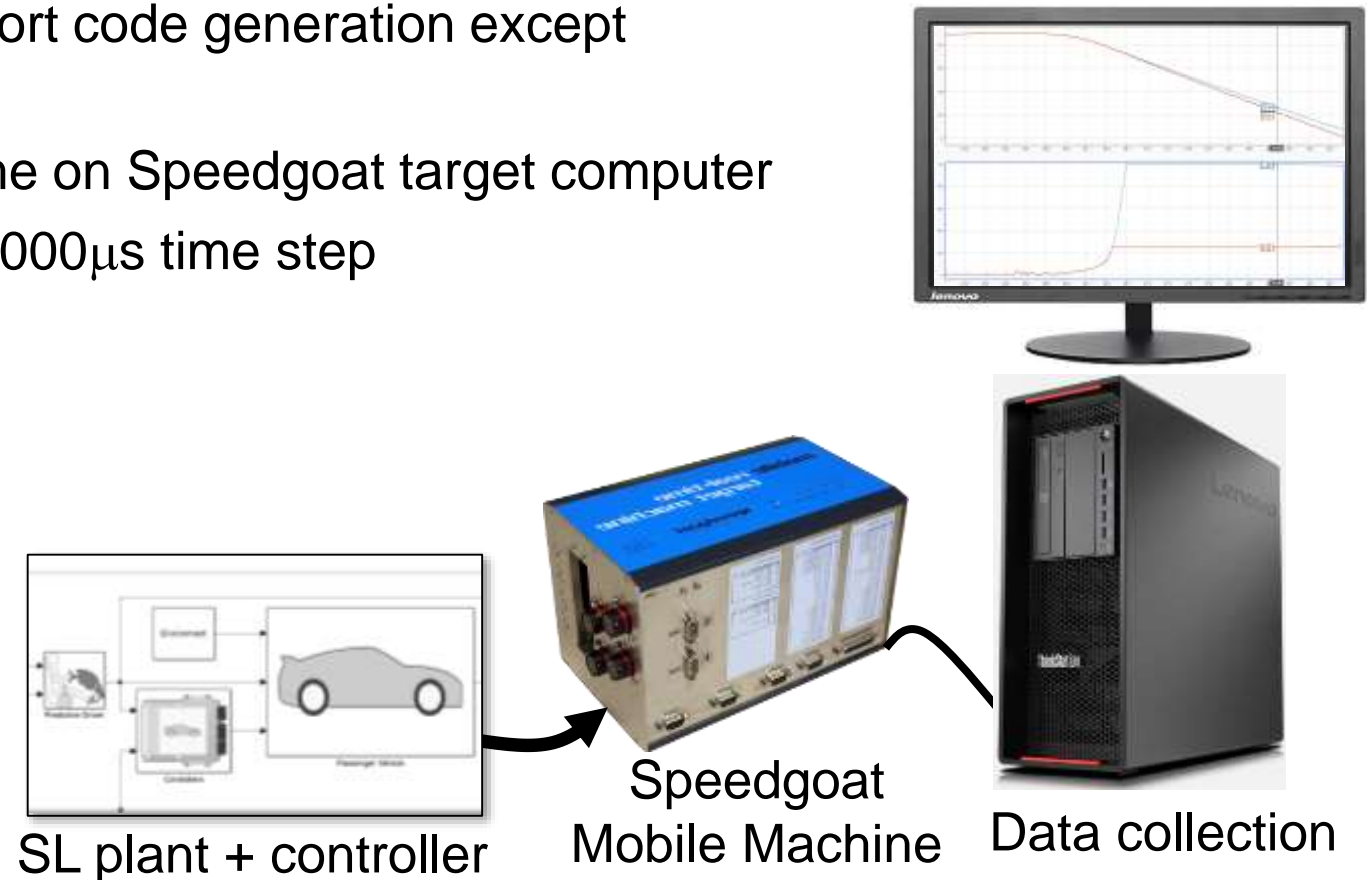


Agenda

- What is Vehicle Dynamics Blockset?
- **How can I use it?**
 - Ride and handling analysis
 - Chassis controls development
 - ADAS / AD testing
 - Hardware-In-the-Loop Testing

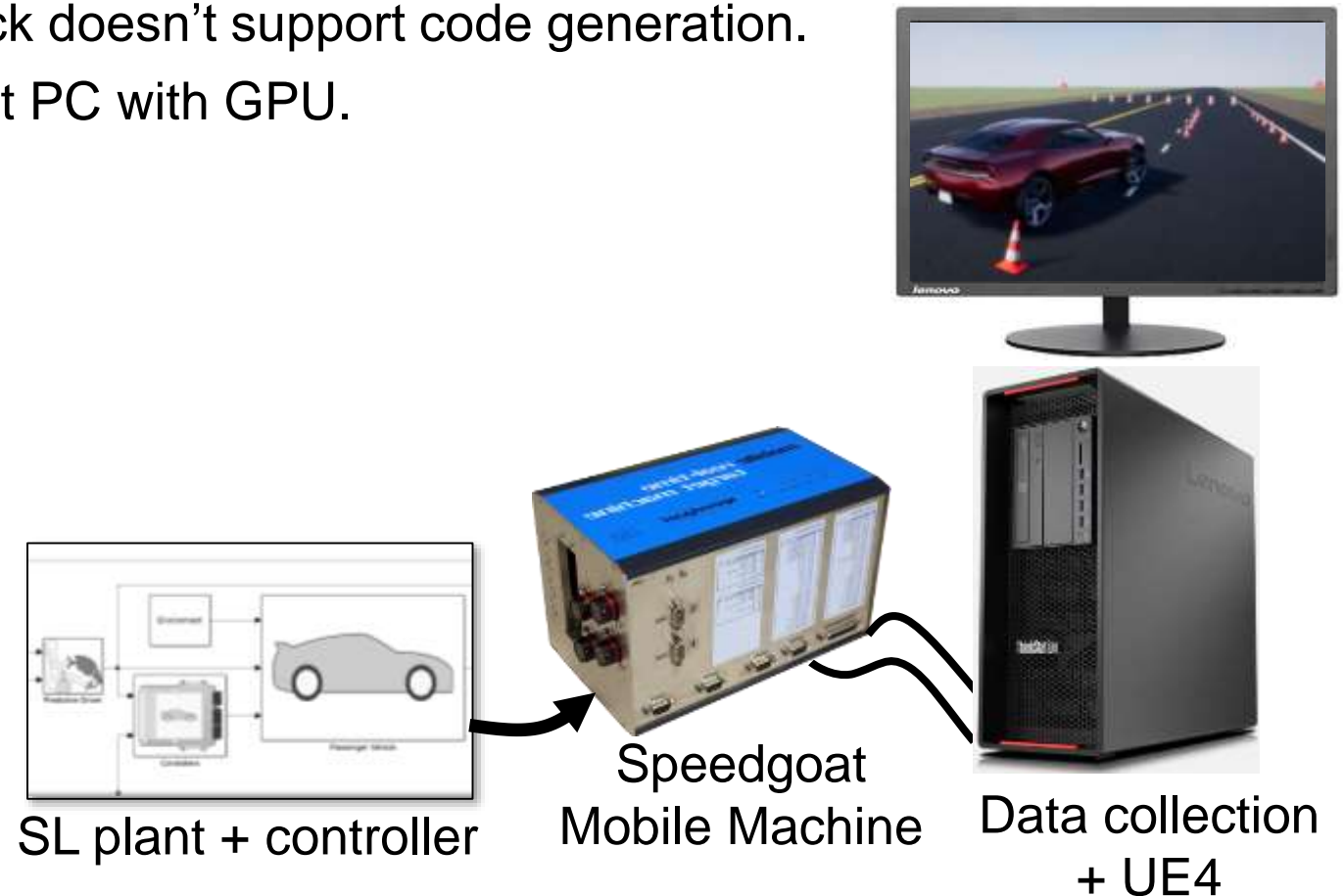
HIL Testing

- Do these models run on HIL simulators?
 - Yes!, All blocks in VDBS support code generation except visualization block
 - Tested with Simulink Real-Time on Speedgoat target computer
 - $\sim 270\mu\text{s}$ turn-around time for $1000\mu\text{s}$ time step



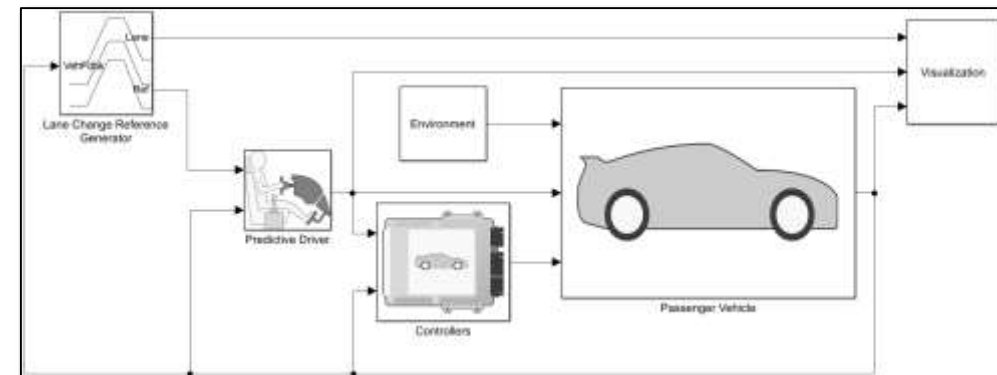
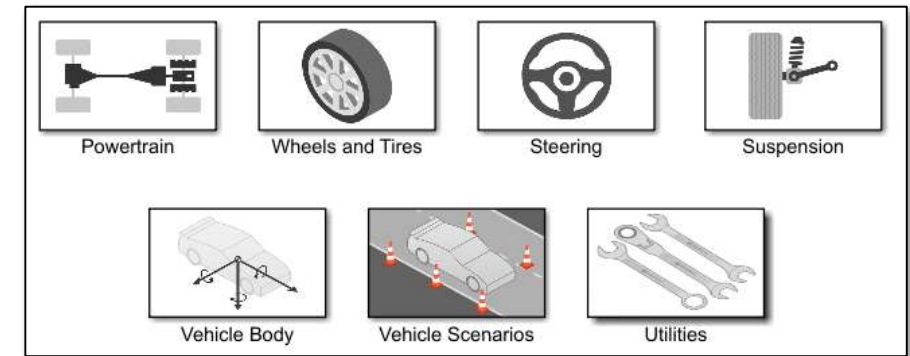
HIL Testing with UE

- Can you perform HIL testing with Unreal Engine running?
 - Yes!, but UE visualization block doesn't support code generation.
 - Unreal Engine can run on host PC with GPU.



Summary

- Vehicle Dynamics Blockset provides:
 - Open and documented library of component and subsystem models
 - Pre-built vehicle models that you can parameterize and customize
 - Fast-running models that are ready for HIL deployment
 - Interface to Unreal Engine



Frequently Asked Questions: Hardware

- What hardware is required to run these models?
 - Simulink only: reference applications run faster than real-time on a modern laptop
 - With 3D engine enabled: Need a good GPU (tested on 1080 nVidia graphics card with 8 GB on-board RAM)



- Do we support Mac / Linux?
 - The Simulink models will run on any platform that Simulink supports
 - The UE4 games are compiled for Windows only, so Mac and Linux users must run in Simulink-only mode (for now)