Vehicle Dynamics Blockset™ Release Notes

MATLAB®&SIMULINK®

MathWorks®

How to Contact MathWorks

Latest news:	`www.mathworks.com`
Sales and services:	`www.mathworks.com/sales_and_services`
User community:	`www.mathworks.com/matlabcentral`
Technical support:	`www.mathworks.com/support/contact_us`
Phone:	508-647-7000

The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

Vehicle Dynamics Blockset™ Release Notes

Trademarks

Patents

Contents

R2023a

R2022b

R2022a

R2021b

R2021a

R2020b

R2020a

R2019b

R2019a

R2018b

R2018a

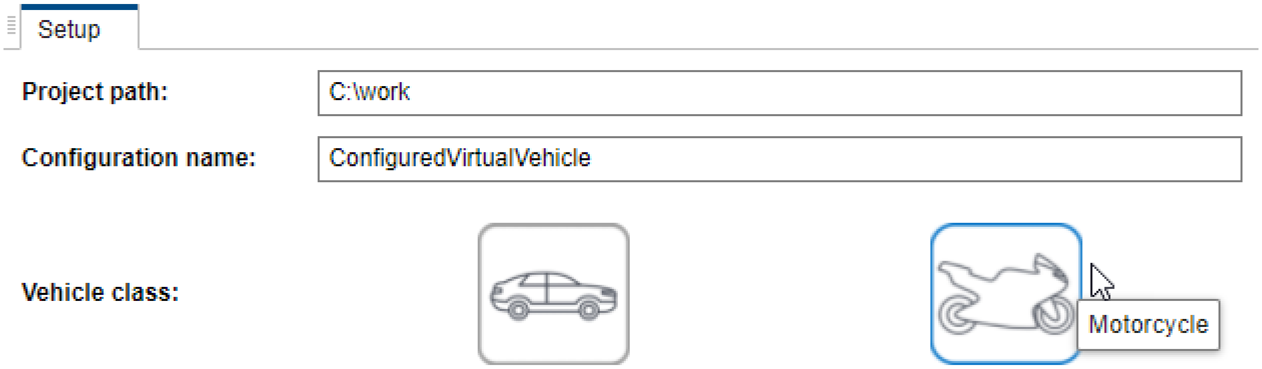
R2023a

Virtual Vehicle Composer: Configure, build, and analyze a motorcycle

If you have Simscape™ and Simscape add-ons, you can use the **Virtual Vehicle Composer** app to configure, build, and analyze a motorcycle. The app provides the Simscape subsystem templates for longitudinal (in-plane) or lateral (out-of-plane) dynamics vehicle analysis.

In the **Virtual Vehicle Composer**, to configure a motorcycle:

1. Set **Vehicle class** to `Motorcycle`.



Project path: C:\work

Configuration name: ConfiguredVirtualVehicle

Vehicle class: Motorcycle

2. Set **Powertrain architecture** to `Conventional Motorcycle` or `Electric Motorcycle with Chain`.
3. Set **Vehicle dynamics** to `In-plane motorcycle dynamics` or `Out-of-plane motorcycle dynamics`.

To configure and build the motorcycle, you need these Simscape add-ons:

- Simscape Driveline™
- Simscape Electrical™
- Simscape Fluids™
- Simscape Multibody™

Virtual Vehicle Composer: Autogenerate static calibrations

In R2023a, **Virtual Vehicle Composer** can calibrate these block types using an imported dataset when building a vehicle:

- CI engine
- SI engine
- Electric motor

Block to Calibrate	Options
CI Mapped Engine	• On the **Setup** tab, select `Conventional Vehicle`. • On the **Data and Calibration** tab, select **Powertrain** > **Engine**. • In the **Engine** list, select `CI Mapped Engine`. • Select `Calibrate from Data`. For more information, see "Calibrate Mapped CI Engine Using Data"
SI Mapped Engine	• On the **Setup** tab, select `Hybrid Electric Vehicle P2`. • On the **Data and Calibration** tab, select **Powertrain** > **Engine**. • In the **Engine** list, select `SI Mapped Engine`. • Select `Calibrate from Data`. For more information, see "Calibrate Mapped SI Engine Using Data".
Mapped Electric Motor	• On the **Setup** tab, select `Electric Vehicle 1EM`. • On the **Data and Calibration** tab, select **Powertrain** > **Electrical System** > **Electric Machine 2**. • In the **Electric Machine 1** list, select `Electric Vehicle 1EM`. • Select `Calibrate from Data`. For more information, see "Calibrate Mapped Electric Motors Using Data".

Static calibrations require the Model-Based Calibration Toolbox™.

Virtual Vehicle Composer: Resize mapped motors

Use the **Virtual Vehicle Composer** app to resize a motor based on high-level specifications. For more information, see "Resize Mapped Motor".

Tire Block: Implement combined lateral and longitudinal slip based on Dugoff model

The Wheels and Tires library now contains the Dugoff Wheel 2DOF block. The block requires fewer parameters than the Combined Slip Wheel 2DOF block. If you do not have the tire coefficients needed by the Magic Formula, consider using the Dugoff Wheel 2DOF block for applications that might require tire models that calculate lateral forces.

The Dugoff Wheel 2DOF block implements a simplified tire with combined lateral and longitudinal slip capability. The block includes:

- Rotational and vertical degrees-of-freedom
- Aligning, rolling, and overturning moments

The vehicle maneuver reference applications contain a subsystem variant with the Dugoff Wheel 2DOF block. For more information, see "Vehicle Reference Applications".

Reference Applications: Improved path tracking for constant radius and double lane change maneuvers

The driver model in the constant radius and double-lane change maneuver reference applications implement a predictive Stanley controller that improves path-tracking. The predictive Stanley controller minimizes the lateral position and angle errors of the current path with respect to a reference path. For more information, see:

- “Constant Radius Maneuver”
- “Double-Lane Change Maneuver”

By default, the reference applications use predictive Stanley driver control. Previously, the maneuvers did not include the predictive Stanley control option.

Simulation 3D Vehicle Enhancements: Configure formula student vehicles and one-axle trailers

You can configure these simulation 3D vehicle blocks with additional meshes. To implement the new vehicles in the 3D simulation environment, use the **Type** parameter.

Block	Type Parameter
Simulation 3D Vehicle Simulation 3D Vehicle with Ground Following	`Formula student vehicle`
Simulation 3D Trailer	`One-axle trailer`

Simulation 3D Environment Upgrade: Run 3D simulations using Unreal Engine 4.27

The simulation 3D environment that comes installed with Vehicle Dynamics Blockset™ is an Unreal Engine® application. The application has been updated to Unreal Engine 4.27. Previously, the toolbox used Unreal Engine 4.26.

For information about using Unreal Engine to create custom scenes, see “Customize 3D Scenes for Vehicle Dynamics Simulations” and “Unreal Engine Simulation Environment Requirements and Limitations”.

Compatibility Considerations

If your Simulink® model uses an Unreal Engine executable or project developed using a prior release of the Vehicle Dynamics Blockset Interface for Unreal Engine 4 Projects support package, the simulation may return an error. To migrate the project to be compatible with the R2023a version of the support package, see “Migrate Projects Developed Using Prior Support Packages”.

Examples: Generate skidpad test

With this example, you can generate a formula vehicle and test it on a skidpad similar to those in student competitions held by Formula Student, Formula SAE, and other organizations internationally. Vehicle Dynamics Blockset and Simulink 3D Animation™ are required.

For more information, see “Generate Skidpad Test”.

R2022b

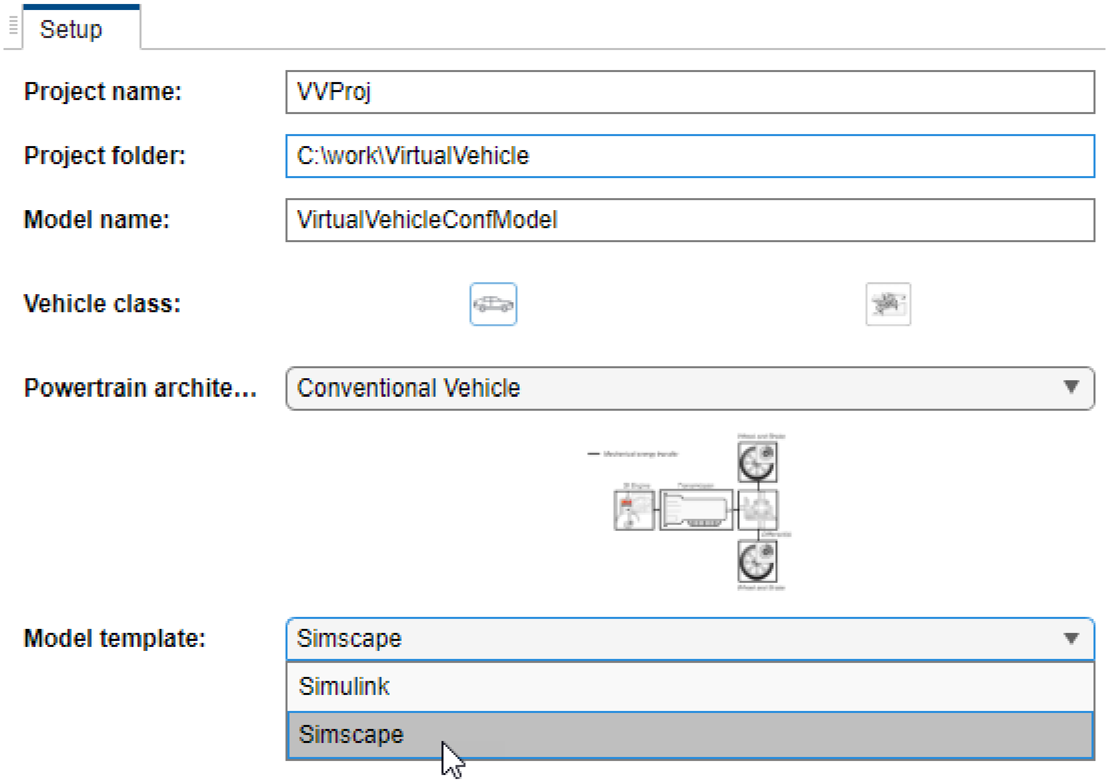
Virtual Vehicle Composer App: Configure vehicles with Simscape subsystems

If you have these Simscape products, you can use the **Virtual Vehicle Composer** app to configure the vehicle plant with Simscape subsystems.

- Simscape Driveline
- Simscape Electrical

The app provides the Simscape subsystem templates for longitudinal vehicle analysis.

When you build your virtual vehicle, on the **Setup** tab, set **Model template** to `Simscape`.



Project name: VVProj

Project folder: C:\work\VirtualVehicle

Model name: VirtualVehicleConfModel

Vehicle class:

Powertrain archite... Conventional Vehicle

Model template: Simscape

Previously, the **Virtual Vehicle Composer** app provided only Simulink subsystem templates.

Simulation 3D Ray Tracer Block: Implement ray tracing in Unreal Engine

The Simulation 3D Ray Tracer block implements ray tracing in the Unreal Engine. Use the block to get positions, surface normals, surface identifiers, and distances for objects in the scene. You can specify block parameters that configure the ray origins, directions, and lengths to adjust the ray trace sensor pattern for your scene and test scenario.

For the minimum hardware required to use the block, see Unreal Engine Simulation Environment Requirements and Limitations.

Terrain Sensor Blocks: Use ray tracing to detect ground terrain

Starting R2022b, the Simulation 3D Terrain Sensor and Vehicle Terrain Sensor blocks include multipoint sensing for more accurate results over irregular terrain.

Simulation 3D Physics Vehicle Block: Control a vehicle in Unreal Engine

Use the Simulation 3D Physics Vehicle block to implement a controllable 6DOF vehicle in the Unreal Engine 3D environment.

For the minimum hardware required to use the block, see Unreal Engine Simulation Environment Requirements and Limitations.

Simulation 3D Scene Configuration Block: Specify ASAM OpenDRIVE file for lane detections

Select the Simulation 3D Scene Configuration block parameter **Select ASAM OpenDRIVE file** to specify an ASAM OpenDRIVE® file. You will need an ASAM OpenDRIVE file if you want to perform any lane detection applications with custom scenes using the Simulation 3D Vision Detection Generator (Automated Driving Toolbox) block.

Simulation 3D Scene Configuration: Use new MATLAB API to download maps locally from the server

Starting in R2022b, you can use the `sim3d.maps` class and its object functions to download prebuilt Unreal Engine scenes and access them directly from the Simulation 3D Scene Configuration block.

- The `sim3d.maps.Map.download` function downloads a map from the server and makes the map locally available.
- The `sim3d.maps.Map.delete` function deletes the map that you download from the server.
- The `sim3d.maps.Map.server` function lists all the maps on the server that are available for download.
- The `sim3d.maps.Map.local` function lists the locally available maps downloaded from the server.

Steering System: Implement dynamic steering with Ackerman, steering shaft, and input options

The Vehicle Dynamics Blockset block library now contains the Steering System block. Use the block to implement a dynamic steering system, including:

- Ackerman percentage that adjusts the ideal Ackerman outside wheel angle. You can use an input port or parameters to specify a constant or table of Ackerman percentages.
- Single or double Cardan joint to model the intermediate steering shaft.
- Friction and compliance effects.
- Input ports for power assistance, Ackerman steering, and kingpin moments.

Tire Blocks: Specify brake, rotational, and inertial characteristics for each wheel

Starting in R2022b, you can use the Combined Slip Wheel 2DOF and Fiala Wheel 2DOF blocks to specify brake, rotational, and inertial characteristics for each wheel on your vehicle. Previously, the blocks used the same parameters for all wheels.

Tire Blocks: Ply steer and turn slip effects in Magic Formula implementation

To use ply steer and turn slip effects in the Magic Formula implementation of these blocks, select the **Ply steer** and **Turn slip** parameters.

- Combined Slip Wheel 2DOF
- Combined Slip Wheel CPI
- Combined Slip Wheel STI

By default, the blocks include ply steer and turn slip in the Magic Formula equations. The equations are fit to flat-belt test data and predict a number of tire effects, including ply steer and turn slip. Consider removing the effects if:

- Your test data does not include ply steer or turn slip data.
- Your analysis does not require ply steer or turn slip effects.

K and C Blocks: Specify AWD, FWD, or RWD drivetrain

The Twist-Beam Suspension - K and C and Independent Suspension - K and C blocks have a **Drivetrain type** parameter that allows you specify the drivetrain:

- `AWD` – All-wheel drive
- `FWD` – Front-wheel drive
- `RWD` – Rear-wheel drive

Previously, the Independent Suspension - K and C block implemented only front-wheel drive.

Twist-Beam Suspension - K and C Block: Implement kinematics and compliance effects on independent and twist-beam suspensions

The Twist-Beam Suspension - K and C block implements the kinematics and compliance (K and C) test suspension characteristics measured from simulated or actual laboratory suspension tests for:

- Independent suspensions on front axles with two wheels
- Twist-beam suspensions on rear axles with two wheels

Suspension Blocks: Parameter name change to specify wheels per axle

Use the **Number of wheels by axle, NumWhlsByAxl** parameter to specify the number of wheels per axle in these blocks.

- Independent Suspension - Double Wishbone
- Independent Suspension - K and C
- Independent Suspension - MacPherson
- Independent Suspension - Mapped
- Solid Axle Suspension
- Solid Axle Suspension - Coil Spring
- Solid Axle Suspension - Leaf Spring
- Solid Axle Suspension - Mapped

Compatibility Considerations

The **Number of tracks by axle, NumTracksByAxl** parameter is renamed to **Number of wheels by axle, NumWhlsByAxl**.

Braking Control Performance Test: Assess commercial truck braking control at 1/2 g deceleration

Use the Three-Axle Tractor Towing a Three-Axle Trailer example in ride and handling analysis and chassis control development of a commercial truck during a braking test. Specifically, use the braking test to assess the ability of the service braking control system to stop the commercial truck at .5 g deceleration without activating the anti-lock braking system (ABS).

Maneuver Reference Applications: Enhanced tire-to-ground contact model

These maneuver reference applications implement a contact model that accounts for the reaction force and torque between the tire and ground. Previously, the contact model accounted for only the reaction force.

- Constant Radius Reference Application
- Double Lane Change Reference Application
- Increasing Steering Reference Application
- Swept-Sine Steering Reference Application
- Braking Test Reference Application

Support Package Installation: Use copyExampleSim3DProject to configure environment for scene customization

To configure your environment so that you can customize scenes, install the Vehicle Dynamics Blockset Interface for Unreal Engine 4 Projects support package. The support package includes these components:

- An Unreal® project, defined in `AutoVrtlEnv.uproject`, and its associated files. The project includes editable versions of the prebuilt 3D scenes that you can select from the **Scene name** parameter of the Simulation 3D Scene Configuration block.

- Three plugins: `MathWorkSimulation`, `RoadRunnerMaterials`, and `MathWorksAutomotiveContent`. These plugins establish the connection between MATLAB® and the Unreal Editor and are required for co-simulation.

Starting in R2022b, you can use the `copyExampleSim3dProject` function to copy the support package components to a folder on your local machine. For example, this code copies the files to `C:\project`.

```
sim3d.utils.copyExampleSim3dProject("C:\project");
```

For more information, see Customize 3D Scenes for Vehicle Dynamics Simulations.

Compatibility Considerations

Previously, the configuration steps to set up the environment were contained in the product documentation. The steps:

- Copied the `MathWorksSimulation`, `RoadRunnerMaterials`, and `MathWorksAutomotiveContent` plugin folders to the Unreal plugin folder.

  In R2022b, `copyExampleSim3dProject` copies the plugins to a specified folder.
- Copied the support package folder containing the `AutoVrtlEnv.uproject` files to a specified folder.

  In R2022b, `copyExampleSim3dProject` also copies the `AutoVrtlEnv.uproject` files to a specified folder.

Functionality being removed or changed

Updated scenes

Behavior change

Starting from R2022b, these scenes in the Unreal Engine 3D environment are rendered using RoadRunner. As a result, the locations of scene objects, including cones and parked vehicles, are moved from their pre-R2022b locations.

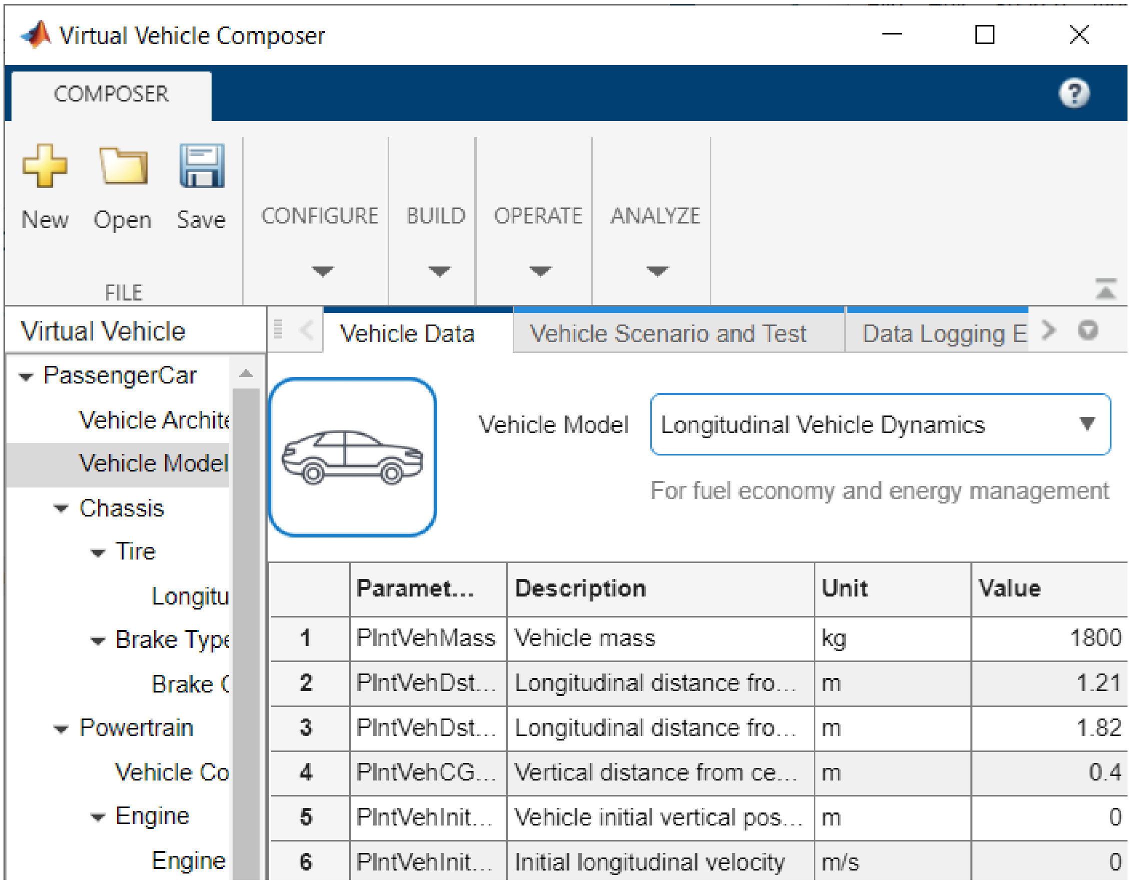
- Curved Road
- Double Lane Change
- Open Surface
- Parking Lot
- Straight Road
- US City Block
- US Highway

3

R2022a

Virtual Vehicle Composer App: Configure, build, and analyze a virtual vehicle

The **Virtual Vehicle Composer** app enables you to configure and build a virtual vehicle that you can use for system-level performance analysis, including ride and handling evaluation, chassis controls development, software integration testing, and hardware-in-the-loop (HIL) testing. To get started, see Get Started with the Virtual Vehicle Composer.



Vehicle Model: Longitudinal Vehicle Dynamics

For fuel economy and energy management

	Paramet...	Description	Unit	Value
1	PlntVehMass	Vehicle mass	kg	1800
2	PlntVehDst...	Longitudinal distance fro...	m	1.21
3	PlntVehDst...	Longitudinal distance fro...	m	1.82
4	PlntVehCG...	Vertical distance from ce...	m	0.4
5	PlntVehInit...	Vehicle initial vertical pos...	m	0
6	PlntVehInit...	Initial longitudinal velocity	m/s	0

Simulation 3D Blocks: Implement multipoint terrain sensing in Unreal Engine

The Simulation 3D Terrain Sensor block implements multipoint terrain sensing in Unreal Engine. Use the block for contact modeling at high vehicle velocities over terrain changes, including speed bumps. The block implements ray tracing to detect the terrain below the tires. Use the block parameters to:

- Sense the terrain under any simulation 3D vehicle actor in the scene, including actors created by the Simulation 3D Vehicle and Simulation 3D Motorcycle blocks.
- Configure the ray origins, directions, and lengths to adjust the terrain sensor pattern for your scene and test scenario.

The Vehicle Terrain Sensor block implements *single*-point terrain sensing in Unreal Engine.

For the minimum hardware required to use the block, see Unreal Engine Simulation Environment Requirements and Limitations.

Simulation 3D Environment Upgrade: Run 3D simulations using Unreal Engine 4.26

The 3D visualization engine that comes installed with Vehicle Dynamics Blockset has been updated to Unreal Engine 4.26. Previously, the toolbox used Unreal Engine 4.25.

For information about using Unreal Engine to create custom scenes, see Customize 3D Scenes for Vehicle Dynamics Simulations and Unreal Engine Simulation Environment Requirements and Limitations.

Compatibility Considerations

If your Simulink model uses an Unreal Engine executable or project developed using a prior release of the Vehicle Dynamics Blockset Interface for Unreal Engine 4 Projects support package, the simulation may return an error. To migrate the project so that it is compatible with the R2022a version of the support package, see Migrate Projects Developed Using Prior Support Packages.

Simulation 3D Motorcycle Vehicle Enhancements: Control lights

You can configure the Simulation 3D Motorcycle block to control the lights.

1. On the block **Light Controls** tab, select **Enable light controls**.
2. Use the enabled parameters to specify the motorcycle lights for:
   - Headlights
   - Brake lights
   - Turn signal lights
3. Connect Boolean light control signals to the `Signal lights` input port.

Simulation 3D Camera Performance Improvements: Run cameras at improved speeds during Unreal Engine simulation

The Simulation 3D Camera Get block now has improved simulation performance and runs at higher frame rates. This table shows the increase in frames per second (FPS) for each camera in an Unreal Engine simulation.

Number of Cameras in Simulation	Frame Rate per Camera (R2021b)	Frame Rate per Camera (R2022a)	Percent Improvement per Camera
1	75.85 FPS	88.45 FPS	16.6%
4	30.51 FPS	32.80 FPS	7.5%

These simulations were timed on a Windows 10, Intel® Xeon® W-2133 CPU @ 3.60 GHz, with 64 GB of RAM and a GPU with 8 GB of on-board RAM.

These improvements enable you to run cameras at real-time speeds, provided that your system meets the requirements specified by the Unreal Engine Simulation Environment Requirements and Limitations.

Suspension Blocks: Implement kinematics and compliance test characteristics in vehicle dynamic performance analysis

Use the Independent Suspension - K and C block to implement kinematics and compliance (K & C) test suspension characteristics measured from simulated or actual laboratory suspension tests. The block models the kinematics and compliance effects of four independent suspensions on a vehicle with two axles and two tracks per axle. You can specify suspension test parameters, including:

- Shock force
- Bounce, roll, and steer tests
- Longitudinal compliance braking test
- Lateral and aligning torque compliance-opposed test

These reference applications include a Kinematics and Compliance Independent Suspension variant that implements the Independent Suspension - K and C block.

- Constant Radius Maneuver
- Double-Lane Change Maneuver
- Slowly Increasing Steering Maneuver
- Swept-Sine Steering Maneuver
- Braking Test

Steering Blocks: Enhanced Ackerman and mapped steering implementation

The steering blocks include these options.

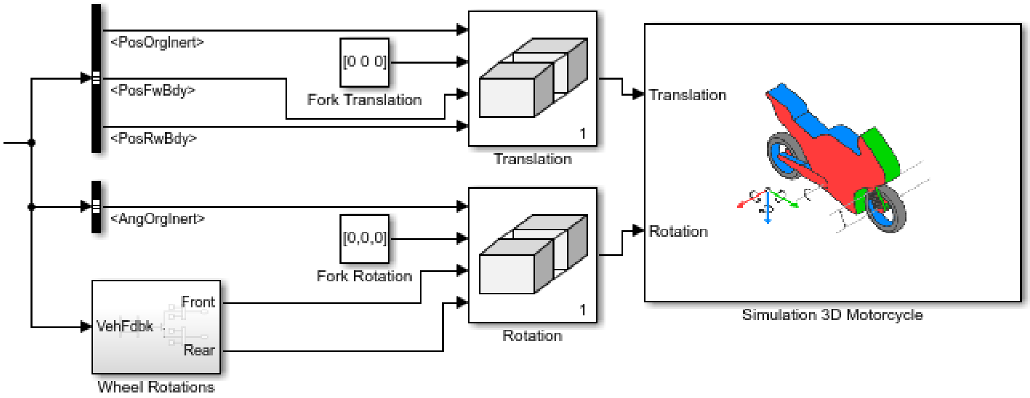
Block	Enhancement
Kinematic Steering	Block uses an Ackerman percentage to adjust the ideal Ackerman outside wheel angle. Use an input port or parameters to specify a constant or table of Ackerman percentages. Previously, the block did not adjust the ideal Ackerman wheel angles.
Mapped Steering	Block implements tables that are functions of rack travel displacement. To determine the rack displacement, you can specify a constant gear ratio or a ratio that is a function of the steering angle. Previously, the block implemented only tables that are functions of the steering angle.

Motorcycle Body Longitudinal In-Plane Block: Enhanced signals to provide Unreal Engine environment with motion

To cosimulate in the Unreal Engine 3D environment and provide a motorcycle with the motion calculated by the Motorcycle Body Longitudinal In-Plane block:

1. Add a Simulation 3D Motorcycle block to your model.
2. Route the Info bus port signals to the Simulation 3D Motorcycle input ports `Translation` and `Rotation`.

   - `PosOrgInert`
   - `PosFwBdy`
   - `PosRwBdy`
   - `AngOrgInert`



Previously, the Motorcycle Body Longitudinal In-Plane block did not contain signals to easily provide the Simulation 3D Motorcycle with the motorcycle motion.

Functionality being removed or changed

Updated Large Parking Lot scene

Behavior change

Starting from R2022a, the Large Parking Lot scene in the Unreal Engine 3D environment is rendered using RoadRunner. As a result, the locations of scene objects, including cones and parked vehicles, are moved from their pre-R2022a locations.

R2021b

Motorcycle Reference Application: Evaluate longitudinal motorcycle braking performance

Use the motorcycle reference application to evaluate longitudinal motorcycle braking performance. You can create your own versions to establish a framework to test that your motorcycle design meets the design requirements under normal and extreme driving conditions. Use this reference application in ride and handling studies and chassis controls development to characterize braking systems, including anti-lock braking system (ABS) control algorithms.

1. To create and open a working copy of the braking test reference application, enter:

   ```
   vdynblksMotoLongBrakingStart
   ```

   By default, the reference application is configured with an ABS controller.
2. Open the Straight Maneuver Reference Generator block. On the **Maneuver Parameters** tab, specify the braking test parameters.
3. Optionally, on the **Tracking Parameters** tab, select **Enable fault tracking before braking**. Use the parameters to specify fault conditions before braking during a split-mu test. If the vehicle speed, steering angle, or yaw rate is not within the allowable range before braking, the block sets a fault condition.
4. Optionally, to view the vehicle response, enable 3D visualization. In the Visualization subsystem, set the 3D Engine block parameter **3D Engine** to **Enabled**. View the vehicle response in the `AutoVrtlEnv` window.

For more information, see Longitudinal Motorcycle Braking Test.

Motorcycle Blocks: Implement in-plane longitudinal motorcycles, including chain dynamics

The Vehicle Dynamics Blockset library now contains blocks that implement an in-plane longitudinal motorcycle and motorcycle chain.

Block	Description
Motorcycle Body Longitudinal In-Plane	Implements a longitudinal in-plane motorcycle body model to calculate longitudinal, vertical, and pitch motion. The block accounts for: • Mass of the frame, rear arm, front upper fork, front lower fork, front wheel, and rear wheel • In-plane dynamic effects of the frame, front lower fork, front wheel, rear wheel, rear suspension, front suspension, rear wheel damper, rear arm, and chain • External forces, external moments, and aerodynamic drag • Road incline • Weight distribution between the axles due to acceleration
Motorcycle Chain	Implements the dynamic effects of a motorcycle chain on the Motorcycle Body Longitudinal In-Plane block, including dynamic tension and moment drive coupling.

For a reference application that uses the blocks, see Longitudinal Motorcycle Braking Test.

Tire Blocks: Implement Magic Formula models compliant with CPI and STI standards

The Wheels and Tires library now contains Magic Formula tire blocks that comply with the contact point interface (CPI) and standard tire interface (STI) Tydex tire standard. The blocks are suitable for applications that require combined lateral slip, for example, in lateral motion and yaw stability studies.

Block	Description
Combined Slip Wheel CPI	Implements the longitudinal and lateral behavior of a wheel characterized by the Magic Formula that complies with the CPI input and output standard. For a reference application that uses the block, see Longitudinal Motorcycle Braking Test.
Combined Slip Wheel STI	Implements the longitudinal and lateral behavior of a wheel characterized by the Magic Formula that complies with the STI input and output standard. For a reference application that uses the block, see Kinematics and Compliance Virtual Test Laboratory.

Powertrain Blocks: Implement transfer cases in drivetrain models and simple engines using basic specifications

The Powertrain library now contains blocks that implement transfer cases in drivetrain models and simple engines using basic specifications.

Block	Description
Transfer Case	Implements a transfer case to find an optimal torque-split ratio for the vehicle. Compare the vehicle performance and fuel consumption in different driving cycles.
Simple Engine	Implements a steady-state engine model using a torque curve table, fuel specific gravity, and fuel efficiency parameters. Use with simple published engine data or internal company sources to facilitate hardware-in-the-loop (HIL) engine control design and vehicle-level fuel economy and performance simulations.

Trailer Blocks: Implement 3DOF and 6DOF trailers with one axle

Use the Trailer Body 3DOF and Trailer Body 6DOF blocks to implement one-axle trailers. Previously, you could configure the blocks only for two or three axles.

For examples of tractors towing trailers, see:

- Two-Axle Tractor Towing a One-Axle Trailer
- Two-Axle Tractor Towing a Two-Axle Trailer
- Three-Axle Tractor Towing a Three-Axle Trailer
- Three-Axle Tractor Towing Two Three-Axle Trailers

Simulation 3D Blocks: Visualize motorcycles and trailer dollies in the Unreal Engine 3D environment

R2021b introduces blocks to visualize motorcycles and trailer dollies in the Unreal Engine 3D environment.

Block	Description	Example
Simulation 3D Motorcycle	Visualize motorcycles in the 3D environment when you develop control algorithms for ride and handling studies and chassis controls development.	Longitudinal Motorcycle Braking Test
Simulation 3D Dolly	Visualize trailer dollies in the 3D environment when you develop control algorithms for tractor trailer towing maneuvers.	Three-Axle Tractor Towing Two Three-Axle Trailers

For the minimum hardware required to use the blocks, see 3D Visualization Engine Requirements and Limitations.

Simulation 3D Environment Upgrade: Run 3D simulations using Unreal Engine, Version 4.25

The 3D visualization engine that comes installed with Vehicle Dynamics Blockset has been updated to Unreal Engine, Version 4.25. Previously, the toolbox used Unreal Engine, Version 4.23.

For information about using Unreal Engine to create custom scenes, see Customize 3D Scenes for Vehicle Dynamics Simulations.

Compatibility Considerations

If your Simulink model uses an Unreal Engine executable or project developed using a prior release of the Vehicle Dynamics Blockset Interface for Unreal Engine 4 Projects support package, the simulation may error. To migrate the project so that it is compatible with the R2021b version of the support package, see Migrate Projects Developed Using Prior Support Packages.

Simulation 3D Environment Performance Improvements: Run 3D simulations faster than real-time

Simulink co-simulations with Unreal Engine can run faster than real-time. Previously, the Unreal Engine frames-per-second (FPS) was limited by the inverse of the *simulation* sample rate. If you want to slow down a 3D simulation to investigate the system behavior, you can still use simulation pacing.

To control simulation time, use the Simulation 3D Scene Configuration block parameter **Sample time**. For example, if **Sample time** is `1/30`, then the visualization engine solver tries to achieve a *minimum* frame rate of 30 fps. However, the real-time graphics frame rate is often lower due to factors such as graphics card performance and model complexity. With reasonable graphics card performance and less model complexity, the fps could be greater than 30, not limited to 30 as it was in previous releases.

Simulink Model Data: Use functions that read and write block parameter data for Excel

If you manage model data in external files, you can use scripts to pass the data between the data file and a Simulink model. The Read and Write Block Parameters to Excel example shows you how to read block parameter data from and write parameter data to an Excel® data file.

Reference Applications: Generate and start maneuvers from non-zero steady-state velocities

These maneuver reference applications include options to start simulations from a non-zero steady-state velocity:

- Double-Lane Change Maneuver
- Slowly Increasing Steering Maneuver
- Swept-Sine Steering Maneuver

Previously, the process to determine the steady-state condition at a specific operating point was iterative, time-consuming, and required multiple simulations.

To use the steady-state feature, open the reference application, then specify parameters on the maneuver Reference Generator blocks. For an example, see Start Double-Lane Change Maneuver at Target Velocity.

Japanese Language Localization Support: Block masks, parameters, variables, and error messages are available in Japanese

Vehicle Dynamics Blockset is available in Japanese for Japanese localized systems. For more information, see Internationalization.

5

R2021a

Braking Test: Use reference application to characterize braking tests, including split-mu

Use this reference application to represent a full vehicle dynamics model undergoing a braking test maneuver, including a split-mu test. You can create your own versions, establishing a framework to test that your vehicle meets the design requirements under normal and extreme driving conditions. Use this reference application in ride and handling studies and chassis controls development to characterize braking systems, including anti-lock braking system (ABS) control algorithms.

1. To create and open a working copy of the braking test reference application, enter:

   ```
   vdynblksBrakingStart
   ```

   By default, the reference application is configured with an ABS controller.
2. Open the Straight Maneuver Reference Generator block. On the **Maneuver Parameters** tab, specify the braking test parameters.
3. Optionally, on the Straight Maneuver Reference Generator block **Tracking Parameters** tab, select **Enable fault tracking before braking**. Use the parameters to specify fault conditions before braking during a split-mu test. If the vehicle speed, steering angle, or yaw rate is not within the allowable range before braking, the block sets a fault condition. The default values represent compliance with ISO 14512.
4. Optionally, in the Environment subsystem, open the Road Track Friction block. Use **Type of surface** to specify the friction coefficient scaling factor:

   - `Constant friction coefficient scaling factor` — Constant surface friction during the maneuver
   - `Split friction coefficient scaling factor` — Two friction coefficients

     Select this option to specify the friction scaling coefficients for a split-mu braking test. Use the enabled parameters to set the ground friction and rectangular surface friction coefficient scaling factors.

For more information about the maneuver, see Braking Test.

ABS Control: Configure vehicle test maneuvers to use five-state anti-lock braking system control

You can configure the reference applications to use a five-state anti-lock braking system (ABS) controller when you simulate the maneuver. Previously, the reference applications implemented only bang-bang ABS control. The five-state ABS controller uses logic-switching based on wheel deceleration and vehicle acceleration to control the braking pressure at each wheel.

Use this option to test ABS control algorithms during standard vehicle test maneuvers, including the double-lane change, split-mu braking, swept-sine, and increasing steering. To configure the reference applications to use the ABS controller:

1. Open a reference application. For example, open the double-lane change reference application.

   ```
   vdynblksDblLaneChangeStart
   ```

   The braking test maneuver reference application uses an ABS controller by default.

2. Navigate to the Controllers subsystem. Right-click the Brake Pressure Control subsystem and select **Variant** > **Label Mode Active Choice** > **Five-State ABS**.

For more information, see the maneuver reference applications:

- Braking Test
- Constant Radius Maneuver
- Double-Lane Change Maneuver
- Slowly Increasing Steering Maneuver
- Swept-Sine Steering Maneuver

Combined Slip Wheel 2DOF Block: Implement fitted tire data sets provided by GCAPS

The Combined Slip Wheel 2DOF block implements the longitudinal and lateral behavior of a wheel characterized by fitted tire data sets provided by the Global Center for Automotive Performance Simulation (GCAPS).

To update the block parameters with the GCAPS fitted tire data:

1. Set **Tire type** to the tire that you want to implement. Options include:
   - `Light passenger car 205/60R15`
   - `Mid-size passenger car 235/45R18`
   - `Performance car 225/40R19`
   - `SUV 265/50R20`
   - `Light truck 275/65R18`
   - `Commercial truck 295/75R22.5`
2. Select **Update applicable Tire Parameters with tire type values**. On the **Tire Parameters** tab, the block updates the applicable parameters, including **Wheel width**, **Rim radius**, and **Wheel mass**.
3. Select **Apply**.

Acknowledgements

Rear Hitches: Implement rear hitch forces, moments, and lateral offsets in trailer blocks

On the **Input signals** pane parameter of the trailer blocks, you can select **Hitch rear forces** and **Hitch rear moments** to create input ports for the rear hitch forces and moments. Additionally, you can specify a lateral rear hitch offset. Previously, you could only specify front hitch forces, moments, and offsets.

This table summarizes the Trailer Body 3DOF and Trailer Body 6DOF block implementation of the front and rear hitch force and moment input.

Parameter	Input Port	Description
Front hitch forces *Renamed from* ***Hitch forces***	`FhF` *Renamed from Fh*	Hitch force applied to the body at the front hitch location, FhF_x, FhF_y, and FhF_z, in the vehicle-fixed frame
Front hitch moments *Renamed from* ***Hitch moments***	`MhF` *Renamed from Mh*	Hitch moment at the front hitch location, MhF_x, MhF_y, and MhF_z, about the vehicle-fixed frame
Rear hitch forces	`FhR`	Hitch force applied to the body at the rear hitch location, FhR_x, FhR_y, and FhR_z, in the vehicle-fixed frame
Rear hitch moments	`MhR`	Hitch moment at the rear hitch location, MhR_x, MhR_y, and MhR_z, about the vehicle-fixed frame

Compatibility Considerations

R2021a renames the Trailer Body 3DOF block and Trailer Body 6DOF block front hitch force and moments from `Fh` to `FhF` and `Mh` to `MhF`, respectively. Models that use the blocks might have compatibility issues, including disconnected line errors. Consider replacing the blocks with the R2021a versions.

Unreal Engine Vehicle Enhancements: Import custom meshes and control vehicle lights

You can configure the Simulation 3D Vehicle and Simulation 3D Vehicle with Ground Following blocks to import custom meshes and control vehicle lights.

To	Action
Import custom meshes	**1** Install the Vehicle Dynamics Blockset Interface for Unreal Engine 4 Projects support package. See Customize 3D Scenes for Vehicle Dynamics Simulations. **2** On the block **Parameters** tab, set **Type** to `Custom`. **3** In the **Path to custom mesh** field, enter the path to the vehicle mesh in the Unreal Engine project. For example, enter `/MathWorksSimulation/Vehicles/Muscle/Meshes/SK_MuscleCar.SK_MuscleCar`. To create a custom vehicle mesh, see Prepare Custom Vehicle Mesh for the Unreal Editor. **4** Use the vehicle dimensions in the custom mesh to enter the dimensions in the corresponding block parameter fields.

To	Action
Control vehicle lights	1 Install the Vehicle Dynamics Blockset Interface for Unreal Engine 4 Projects support package. See Customize 3D Scenes for Vehicle Dynamics Simulations. 2 On the block **Light Controls** tab, select **Enable light controls**. 3 Use the enabled parameters to specify the vehicle lights for: • Headlights • Brake lights • Reverse lights • Turn signal lights 4 Connect Boolean light control signals to the `Signal lights` input port. For a vehicle light control example, see Scene Interrogation in 3D Environment.

Unreal Engine Scene Environment: Control weather and sun position

Use the Simulation 3D Scene Configuration block to control scene weather and sun position. Options allow you to create realistic environments when you run maneuvers and test control algorithms in the Unreal Engine 3D simulation environment. The Simulation 3D Camera Get block receives the image from the 3D simulation environment.

To control scene weather and sun position, on the Simulation 3D Scene Configuration block **Weather** tab, select **Override scene weather**. Use the enabled parameters to change the sun position, clouds, fog, and rain during the simulation.

6

R2020b

Vehicles and Trailers: Implement 6DOF trailers and vehicles with three axles

The Vehicle Body block library now contains blocks that implement a six degree-of-freedom (DOF) trailer and a six DOF vehicle with three axles. To analyze the vehicle dynamics under different loading conditions, you can use the block **Inertial Loads** parameters.

- Vehicle Body 6DOF Three Axles — Models a three-axle vehicle body.
- Trailer Body 6DOF — Models a two- or three-axle trailer body.

Simulation 3D Blocks: Visualize tractors and trailers in the Unreal Engine 3D environment

Use the blocks to visualize tractors and trailers in the Unreal Engine 3D environment when you develop control algorithms for standard maneuvers like the split-friction test. This table provides the block descriptions.

Block	Description
Simulation 3D Tractor	Implements a three-axle conventional or cab-over tractor in the 3D simulation environment.
Simulation 3D Trailer	Implements a two- or three- axle trailer in the 3D visualization environment.

For an example that uses the blocks, see Three-Axle Tractor Towing a Three-Axle Trailer.

For the minimum hardware required to use the blocks, see 3D Visualization Engine Requirements and Limitations.

Camera Views: Usability improvements, including controls for visualizing vehicle acceleration, pitch, and roll

The camera views in the Unreal Engine 3D visualization environment include these usability improvements.

Smooth Transition Between Views

Press the keyboard keys 0-9 to transition smoothly between vehicle camera views.

Cycle Through Vehicles in Scene

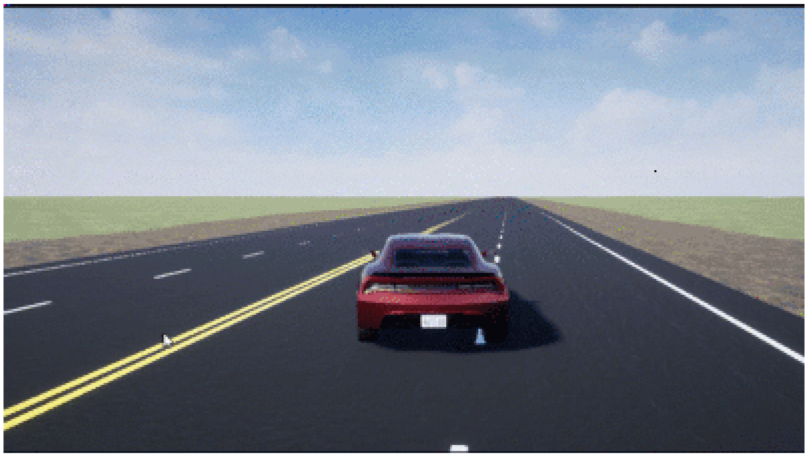
Press the **Tab** key to cycle the view between all vehicles in the scene.

Vehicle Acceleration and Rotation

Press the **L** key to toggle a camera lag effect on or off. When you enable the lag effect, the camera view includes:

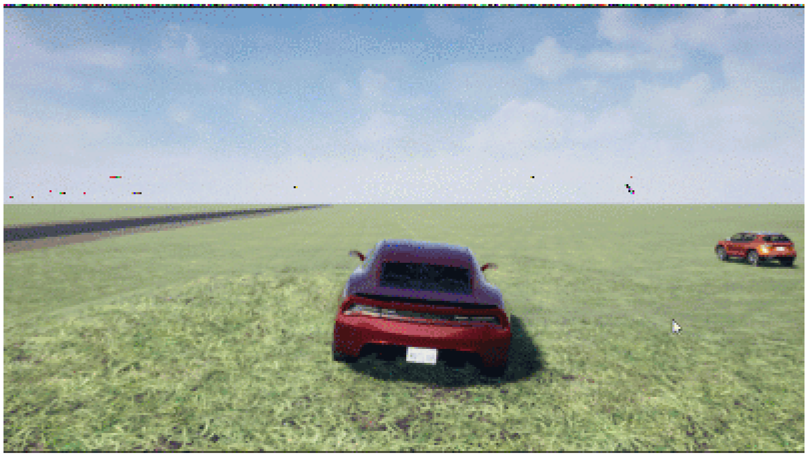
- Position lag, based on the vehicle translational acceleration
- Rotation lag, based on the vehicle rotational velocity

This lag enables improved visualization of overall vehicle acceleration and rotation.



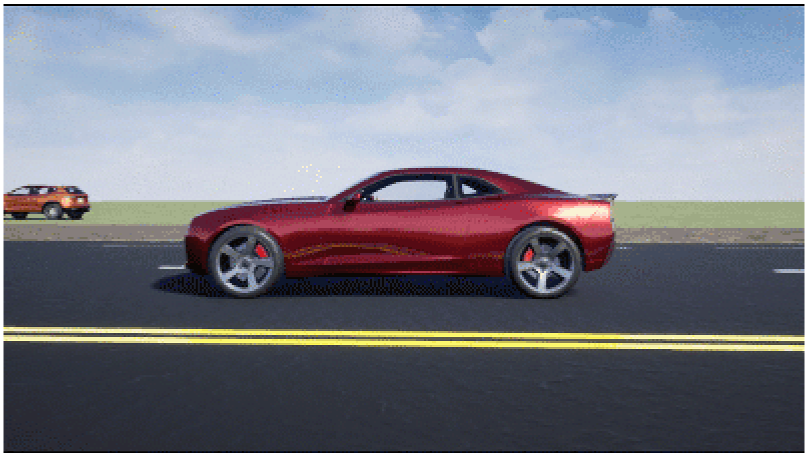
Vehicle Pitch and Roll

The views lock the camera pitch and roll to the horizon, providing improved visualization of the vehicle pitch and roll.



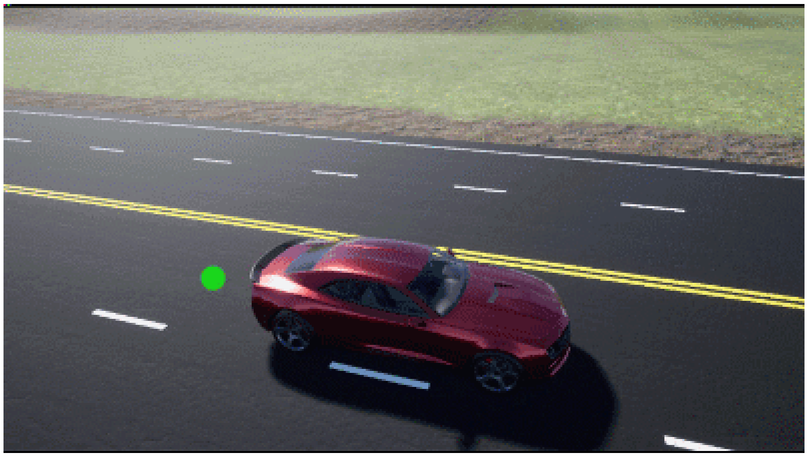
Camera Distance

To control the camera distance from the vehicle, use the mouse scroll-wheel.



Free-Camera Views

Press the **F** key to toggle the free camera mode or off. When you enable the free camera mode, you can use the mouse to change the pitch and yaw of the camera. This mode enables you to orbit the camera around the vehicle.



Driver Blocks: Implement Stanley method for lateral control

The Lateral Driver and Predictive Driver blocks include an option to implement the Stanley method for lateral control. This method uses a control law that minimizes the difference between the vehicle position and angle and a reference vehicle position and angle.

To configure the blocks to use the Stanley method, set **Lateral control type, controlTypeLat** to `Stanley`. On the **Reference Control** pane, use the **Include dynamics** parameter to specify the type of model for the controller to use.

Setting	Implementation
`off`	Controller uses a kinematic bicycle model that is suitable for path following in low-speed environments such as parking lots, where inertial effects are minimal.
`on`	Controller uses a dynamic bicycle model that is suitable for path following in high-speed environments such as highways, where inertial effects are more pronounced.

Hitches: Implement hitch forces, moments, and lateral offsets in all vehicle and trailer blocks

On the **Input signals** pane parameter of the vehicle body and trailer blocks, you can select **Hitch forces** and **Hitch moments** to create input ports for the hitch forces and moments. Additionally, you can specify a lateral hitch offset.

Input Signals Pane Parameter	Input Port	Description
Hitch forces	`Fh`	Hitch force applied to the body at the hitch location, Fh_x, Fh_y, and Fh_z, in the vehicle-fixed frame
Hitch moments	`Mh`	Hitch moment at the hitch location, Mh_x, Mh_y, and Mh_z, about the vehicle-fixed frame

This table summarizes the vehicle and trailer block changes to implement the hitch force and moment input.

Block	Hitch Force and Moment Input Change
Vehicle Body 3DOF	*New — Option available in R2020b*
Vehicle Body 6DOF	
Vehicle Body 3DOF Three Axles	*None — Option available when blocks were introduced in R2020a*
Trailer Body 3DOF	
Vehicle Body 6DOF Three Axles	*New — Option available for new blocks introduced in R2020b*
Trailer Body 6DOF	

Scene Customization Workflows: Place cameras and animate actors in Unreal Editor

R2020b includes these workflows for customizing scenes and communicating with the Unreal Editor.

Workflow	Description
Install Support Package and Configure Environment	To customize the scenes, start by following this workflow to install the Vehicle Dynamics Blockset Interface for Unreal Engine 4 Projects support package.
Get Started Communicating with the Unreal Engine Visualization Environment	Send and receive data from the Unreal Editor by using the Simulation 3D Message Get and Simulation 3D Message Set blocks.
Animate Custom Actors in the Unreal Editor	Animate a bicycle actor in the Unreal Editor by using the Simulation 3D Actor Transform Set block.
Place Cameras on Actors in the Unreal Editor	Place the Simulation 3D Camera Get block on static and custom actors in the Unreal Editor.

7

R2020a

Vehicle and Trailer Blocks: Implement 3DOF trailers and vehicles with three axles

The Vehicle Body block library now contains blocks that implement a three degree-of-freedom (DOF) trailer and a three DOF vehicle with three axles.

- Vehicle Body 3DOF Three Axles — Models a three-axle vehicle body with longitudinal, lateral, and yaw motion. Configure for a single or dual track.
- Trailer Body 3DOF — Models a trailer body with longitudinal, lateral, and yaw motion. Configure a two-axle or three-axle trailer with a single or dual track.

For an example that uses the blocks, see Three-Axle Tractor Towing a Trailer.

Drive Cycle Fault Tracking: Identify drive cycle faults specified by standardized tests

Use the Drive Cycle Source block to identify drive cycle faults during component, sizing, fuel economy, and emissions analysis. You can set parameters that identify the faults within tolerances specified by standardized tests, including:

- EPA dynamometer driving schedules
- Worldwide Harmonised Light Vehicle Test Procedure laboratory tests

On the Drive Cycle Source block **Fault Tracking** tab, use the parameters to specify the fault tolerances. If the vehicle speed or time tolerance is not within the allowable range, the block sets a fault condition.

Driver Blocks: Configure inputs to control acceleration, deceleration, and steering commands

The Lateral Driver, Longitudinal Driver, and Predictive Driver blocks include input options that allow you to configure simulations of standard test maneuvers, including maximum forward speed (MFS). Use the **External action** parameters to create input ports for external signals that control the accelerator, decelerator, and steering commands.

Block	External Actions
Longitudinal Driver	• Disable the acceleration or deceleration commands. • Hold the acceleration or deceleration commands at the current values. • Override the acceleration or deceleration commands with input commands.
Lateral Driver	• Disable the steering command. • Hold the steering command at the current value. • Override the steering command with an input command.
Predictive Driver	• Disable the acceleration, deceleration, or steering commands. • Hold the acceleration, deceleration, or steering commands at the current values. • Override the acceleration, deceleration, or steering commands with input commands.

Compatibility Considerations

R2020a renames the Longitudinal Driver block and Lateral Driver block gear command output ports from `Gear` to `GearCmd`. Models that use the blocks might have compatibility issues, including disconnected line errors. Consider replacing the blocks with the R2020a version.

Headless Mode: Run 3D simulations without opening the Unreal Engine 3D visualization display

You can run simulations in the 3D environment without visualizing the results, that is, in headless mode. To run 3D simulations in headless mode, open the Simulation 3D Scene Configuration block in a model. Clear **Display 3D simulation**. By default, the block is not configured for headless mode.

Consider running in headless mode if you want to:

- Run multiple 3D simulations in parallel to test models in different Unreal Engine scenarios.
- Optimize model parameters without visualizing the results in the Unreal Engine environment. For example, consider using headless mode if you want to tune vehicle suspension parameters over a terrain scenario defined in Unreal Engine.

Inertial Measurement Unit Block: Implement a three-axis inertial block to measure accelerations and angular rates

The Sensor block library now includes a Three-axis Inertial Measurement Unit block that contains a three-axis accelerometer and gyroscope. Use this block to measure vehicle accelerations from the accelerometer and vehicle angular rates from the gyroscope.

3D Simulation Version Upgrade: Run 3D simulations using Unreal Engine, Version 4.23

The 3D visualization engine that comes installed with Vehicle Dynamics Blockset has been updated to Unreal Engine, Version 4.23. Previously, the toolbox used Unreal Engine, Version 4.19.

For information about using Unreal Engine to create custom scenes, see Support Package for Customizing Scenes.

Compatibility Considerations

If your Simulink model uses an Unreal Engine executable or project developed using a prior release of the *Vehicle Dynamics Blockset Interface for Unreal Engine 4 Projects* support package, the simulation will error. To migrate the project so that it is compatible with the R2020a version of the support package, see Migrate Projects Developed Using Prior Support Packages.

Simulation 3D Scene Configuration Block: Improved parameter options for using customized scenes

The Simulation 3D Scene Configuration block now provides improved parameter options for simulating driving scenarios within your own customized scenes. The customized scenes must be created using the Unreal Editor and must be compatible with Version 4.23.

This table summarizes the Simulation 3D Scene Configuration block parameter updates in R2020a.

To	*New* Parameter Settings (R2020a)	*Old* Parameter Settings (R2019b)
Simulate in one of the default, prebuilt scenes provided with Vehicle Dynamics Blockset.	**Scene source** — `Default Scenes` **Scene name** — *`Name of prebuilt scene`*	On the **Co-Simulation** tab, **Co-Simulation format** — `Executable` On the **Simulation Configuration** tab, **Scene description** — *`Name of prebuilt scene`*
Simulate in a scene that is part of an Unreal Engine executable file.	**Scene source** — `Unreal Executable` **File name** — `VehicleSimulation.exe` (default) or *`Path to the executable file`* **Scene** — *`Path to the scene within the executable file`*	On the **Co-Simulation** tab, **Co-Simulation format** — `Executable` On the **Simulation Configuration** tab: • **Scene description** — `Custom` • **Project name** — *`Path to the executable file, followed by the path to the scene within the file`*
Simulate in a scene that is part of an Unreal Engine project (`.uproject`) file and is open in the Unreal Editor.	**Scene source** — `Unreal Editor` **Project** — *`Path to project file`*	On the **Co-Simulation** tab, **Co-Simulation format** — `Editor`

To use customized scenes, you must install the Vehicle Dynamics Blockset Interface for Unreal Engine 4 Projects support package. For details, see Support Package for Customizing Scenes.

New Vehicle Type: Implement a box truck vehicle in 3D simulations

You can configure the Simulation 3D Vehicle and Simulation 3D Vehicle with Ground Following blocks to implement a box truck in 3D simulations. To configure the vehicle blocks, set **Type** to `Box truck`. For the vehicle dimensions see, Box Truck.

8

R2019b

Simulation 3D Blocks: Visualize simulations and communicate with the Unreal Engine 3D visualization environment

The Vehicle Scenarios > Sim3D > Sim3D Core block library contains new blocks and updates to existing blocks. Use the blocks to visualize simulations and communicate scene information with the Unreal Engine 3D visualization environment. This table summarizes the updates.

Block	Update
Simulation 3D Message Get	*New* The block retrieves data from the Unreal Engine 3D visualization environment. To use the block: • Install the Support Package for Customizing Scenes. • In the Unreal Engine environment, set up blueprint or C++ actor classes that can send the data to Simulink.
Simulation 3D Message Set	*New* The block sends data to the Unreal Engine 3D visualization environment. To use the block: • Install the Support Package for Customizing Scenes. • In the Unreal Engine environment, set up blueprint or C++ actor classes that can receive the Simulink data.
Simulation 3D Scene Configuration	Block renamed from Simulation 3D Config. In R2019b, the block does not include these output ports: • `State` — Visualization engine state • `Ts` — Visualization engine sample time
Simulation 3D Actor Transform Get	Parameters that allow you to get the position, rotation, and scale for scene actors, including vehicles. Previously, you had to specify a vehicle or camera actor type. In R2019b, use the Simulation 3D Camera Get block to configure a scene camera and get the camera image.
Simulation 3D Actor Transform Set	Parameters that allow you to set the position, rotation, and scale for scene actors, including vehicles. Previously, you had to specify a vehicle or camera actor type. In R2019b, use the: • Simulation 3D Camera Get block to configure a scene camera and get the camera image. • Simulation 3D Vehicle with Ground Following or Simulation 3D Vehicle block to configure a vehicle.
Simulation 3D Camera Get	Mounting parameters that allow you to specify the location of the camera in the 3D visualization environment.

For the minimum hardware required to use the blocks, see 3D Visualization Engine Requirements.

Compatibility Considerations

Models that use these simulation 3D blocks might have compatibility issues, including disconnected line errors. Consider replacing the blocks with the R2019b version.

Vehicle Blocks: Implement vehicles in the Unreal Engine 3D visualization environment

The Vehicle Scenarios > Sim3D > Sim3D Vehicle block library includes blocks that implement vehicles in the Unreal Engine 3D visualization environment. This table summarizes the new vehicle blocks.

Block	Description
Simulation 3D Vehicle with Ground Following	*New* Implements a vehicle with four wheels that follows the ground in the 3D visualization environment. The block uses the vehicle position to adjust the vehicle elevation, roll, and pitch so that the vehicle follows the ground terrain. It determines the vehicle velocity and heading and adjusts the steering angle and rotation for each wheel. You can select the type of vehicle, color, and initial position and rotation.
Simulation 3D Vehicle	*New* Implements a vehicle in the 3D visualization environment. The block uses the vehicle translation and rotation to place a vehicle with four wheels in the visualization environment. You can select the type of vehicle, color, and initial position.

For the minimum hardware required to use the blocks, see 3D Visualization Engine Requirements.

Scene Interrogation Reference Application: Interactive vehicle control and enhanced dynamics

The scene interrogation with camera and ray tracing reference application includes:

- Dials and gauges to control the vehicle steering, braking, and throttle in the 3D visualization environment.
- Powertrain, driveline, and steering models.
- Vehicle dynamics model for a 3 degree-of-freedom (DOF) vehicle.

Use the reference application to integrate a 3-DOF vehicle in the 3D visualization environment. The 3D visualization environment uses the Unreal Engine by Epic Games®.

To create and open a working copy of the camera and ray tracing reference application project, enter

```
vdynblksSceneCameraRayStart
```

For the minimum hardware required to run the reference application, see 3D Visualization Engine Requirements.

For more information about the reference application, see Scene Interrogation in 3D Environment.

9

R2019a

Kinematics and Compliance Virtual Test Laboratory: Generate mapped suspension calibration parameters from spreadsheet data

In the kinematics and compliance virtual test laboratory, select **Generate Mapped Suspension from Spreadsheet Data** to generate mapped suspension calibration parameters from measured vertical force and suspension geometry data. The virtual test lab uses Model-Based Calibration Toolbox to fit camber angle, toe angle, and vertical forces to generate calibrated suspension parameters for the mapped suspension blocks.

For more information, see Kinematics and Compliance Virtual Test Laboratory.

Active Damping: Implement damping as a function of duty cycle and suspension velocity in suspension blocks

During open-loop and closed-loop suspension feedback control algorithm development, you can configure the Independent Suspension - Double Wishbone and Independent Suspension - MacPherson suspension blocks to implement active damping. Use the **Enable active damping** parameter.

Enable Active Damping Setting	Damping Implementation
`off`	Constant
`on`	Lookup table as a function of active damper duty cycle and actuator velocity

New Fiala Tire Block: Implement a simplified tire with lateral and longitudinal slip capability

Use the Fiala Wheel 2DOF block to implement a simplified tire with lateral and longitudinal slip capability. The block uses a translational friction model to calculate the forces and moments during combined longitudinal and lateral slip, requiring fewer parameters than the Combined Slip Wheel 2DOF block. If you do not have the tire coefficients needed by the Magic Formula, consider using this block for studies that do not involve extensive nonlinear combined lateral slip or lateral dynamics. If your study does require nonlinear combined slip or lateral dynamics, consider using the Combined Slip Wheel 2DOF block.

Constant Radius Test: Use maneuver reference application to characterize steering and lateral vehicle dynamics

This reference application represents a full vehicle dynamics model undergoing a constant radius test maneuver. You can create your own versions, establishing a framework to test that your vehicle meets the design requirements under normal and extreme driving conditions. Use this reference application in ride and handling studies and chassis controls development to characterize the steering and lateral vehicle dynamics. For information about the similar maneuvers, see standards SAE J266_199601 and ISO 4138:2012.

To create and open a working copy of the constant radius reference application, enter

```
vdynblksConstRadiusStart
```

For more information about the maneuver, see Constant Radius Maneuver.

Fiala and Longitudinal Wheel Blocks: Configure blocks for ISO 28580 rolling resistance calculation

To help your models comply with ISO 28580:2018, *Passenger car, truck and bus tyre rolling resistance measurement method — Single point test and correlation of measurement results*, you can configure the Fiala Wheel 2DOF and Longitudinal Wheel blocks to use the ISO 28580:2018 rolling resistance calculation.

To configure the blocks for the ISO calculation, set the **Rolling Resistance** parameter to `ISO 28580`.

Vehicle Body Blocks: Configuration options for external inputs, including forces, moments, and air temperature

The vehicle body blocks have new options for configuring the block with external inputs. The options provide more flexibility when you incorporate your blocks into full vehicle models. The table summarizes the settings.

Block	***New*** **External Input Ports**	**Block Option Setting**	**Description**
Vehicle Body 1DOF Longitudinal	`FExt`	**External forces**	External force applied to vehicle CG in vehicle-fixed frame.
	`MExt`	**External moments**	External moment about vehicle CG in vehicle-fixed frame.
	`AirTemp`	**Air temperature**	Ambient air temperature. Consider this option if you want to vary the temperature during run-time.
	`WindXYZ`	**Wind X,Y,Z**	Wind speed along earth-fixed *X*-, *Y*-, and *Z*-axes. If you do not select this option, the block implements input port `WindX` — Longitudinal wind speed along the earth-fixed *X*-axis.
Vehicle Body 3DOF Longitudinal	`FExt`	**External forces**	External force applied to vehicle CG in vehicle-fixed frame.
	`MExt`	**External moments**	External moment about vehicle CG in vehicle-fixed frame.
	`AirTemp`	**Air temperature**	Ambient air temperature. Consider this option if you want to vary the temperature during run-time.
Vehicle Body 3DOF	`X_o`	**Input signals** pane **Initial longitudinal position**	Initial vehicle CG displacement along earth-fixed *X*-axis, in m

Block	*New* External Input Ports	Block Option Setting		Description
R2018a included block options to input the external forces, moments, friction, steering angles, and wind speed	`Y_o`		**Initial lateral position**	Initial vehicle CG displacement along earth-fixed *Y*-axis, in m
	`xdot_o`		**Initial longitudinal velocity**	Initial vehicle CG velocity along vehicle-fixed *x*-axis, in m/s
	`ydot_o`		**Initial lateral velocity**	Initial vehicle CG velocity along vehicle-fixed *y*-axis, in m/s
	`psi_o`		**Initial yaw angle**	Initial rotation of vehicle-fixed frame about earth-fixed *Z*-axis (yaw), in rad
	`r_o`		**Initial yaw rate**	Initial vehicle angular velocity about the vehicle-fixed *z*-axis (yaw rate), in rad/s
	`AirTemp`		**Air temperature**	Ambient air temperature. Consider this option if you want to vary the temperature during run-time.
Vehicle Body 6DOF	`AirTemp`	**Environment** pane	**Air temperature**	Ambient air temperature. Consider this option if you want to vary the temperature during run-time.

Compatibility Considerations

Models that use the vehicle body blocks might have disconnected line errors. Consider replacing the blocks with the R2019a version.

Differential Blocks: Account for efficiency as a function of torque, temperature, and driveshaft speed

To account for Open Differential and Limited Slip Differential block power and energy losses, you can use the **Efficiency factors** parameter. This table summarizes the block implementation for each setting.

Setting	Implementation
`Constant`	Constant efficiency that you can set with the **Constant efficiency factor, eta** parameter.

Setting	Implementation
`Driveshaft torque, temperature and speed`	Efficiency as a function of base gear input torque, air temperature, and driveshaft speed. Use these parameters to specify the lookup table and breakpoints: • **Efficiency lookup table, eta_tbl** • **Efficiency torque breakpoints, Trq_bpts** • **Efficiency speed breakpoints, omega_bpts** • **Efficiency temperature breakpoints, Temp_bpts** For the air temperature, you can either: • Select **Input temperature** to create an input port. • Set a **Ambient temperature, Tamb** parameter value.

Maneuver Reference Applications: Position vehicle in scene-specific recommended locations

When you run the maneuver reference applications in the 3D visualization environment, you can initially position the vehicle in locations that are recommended for the scene. After you open the reference applications for these maneuvers:

- Double-Lane Change Maneuver
- Slowly Increasing Steering Maneuver
- Swept-Sine Steering Maneuver

1. In the Visualization subsystem, open the 3D Engine block. Set these parameters.
   - **3D Engine** to **Enabled**.
   - **Scene** to one of the scenes.
2. To position the vehicle in the scene:
   a. Select the position initialization method:
      - **Recommended for scene** — Set the initial vehicle position to values recommended for the scene
      - **User-specified** — Set your own initial vehicle position
   b. Select **Apply** to modify the initial vehicle position parameters.
   c. Click **Update the model workspaces with the initial values** to overwrite the initial vehicle position in the model workspace with the applied values.

10

R2018b

Active Differential Block: Model a spur or dual planetary active differential gear for applications such as torque vectoring

R2018b includes an Active Differential block that can model the active differential as an open differential coupled to either a spur or planetary differential gear set. The block uses external pressure signals to regulate the clutch pressure to either speed up or slow down each axle rotation.

Use the block in hardware-in-the-loop (HIL) and optimization workflows to dynamically couple the driveshaft to the wheel axles when you want to direct the transmission torque to a specific axle. For detailed front-wheel driving studies, use the block to couple the driveshaft to universal joints. The block is suitable to use in system-level closed-loop control studies, for example, yaw stability and torque vectoring.

Vehicle Body 6DOF Block: Include additional loads on the vehicle body such as passengers, powertrain components, and cargo

To analyze the vehicle dynamics under different loading conditions, use the Vehicle Body 6DOF block **Inertial Loads** parameters. Specifically, you can specify these loads:

- Powertrain components
- Front and rear row passengers
- Overhead cargo
- Rear cargo

Kinematics and Compliance Virtual Test Laboratory: Use a reference application to calibrate suspension parameters

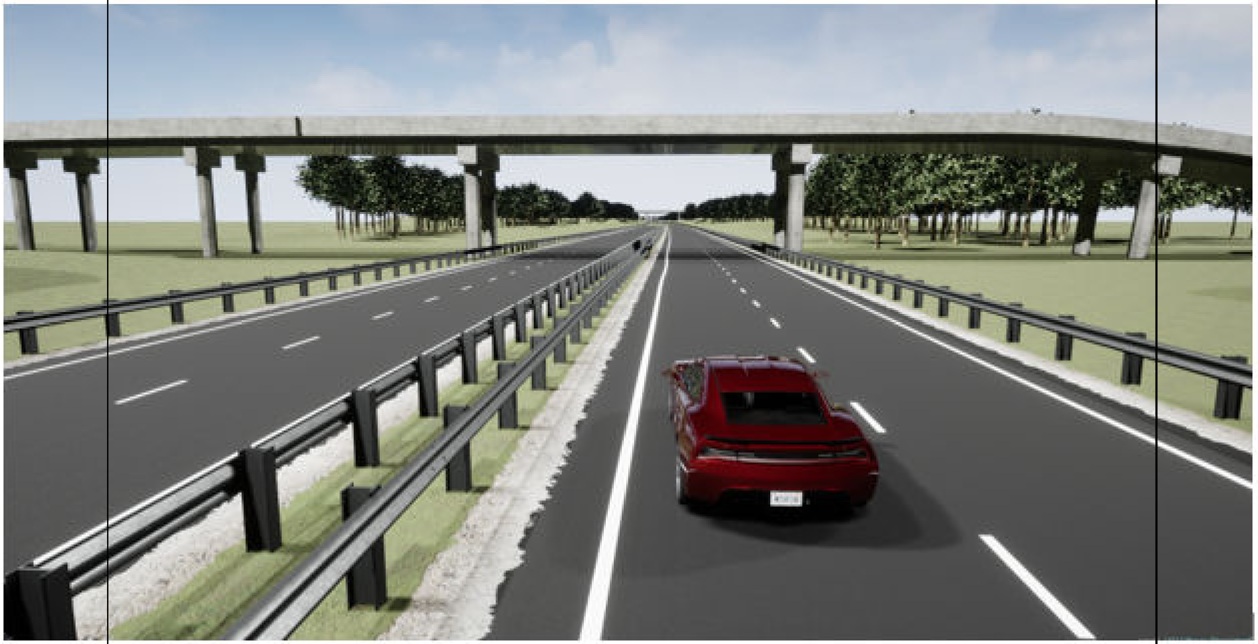
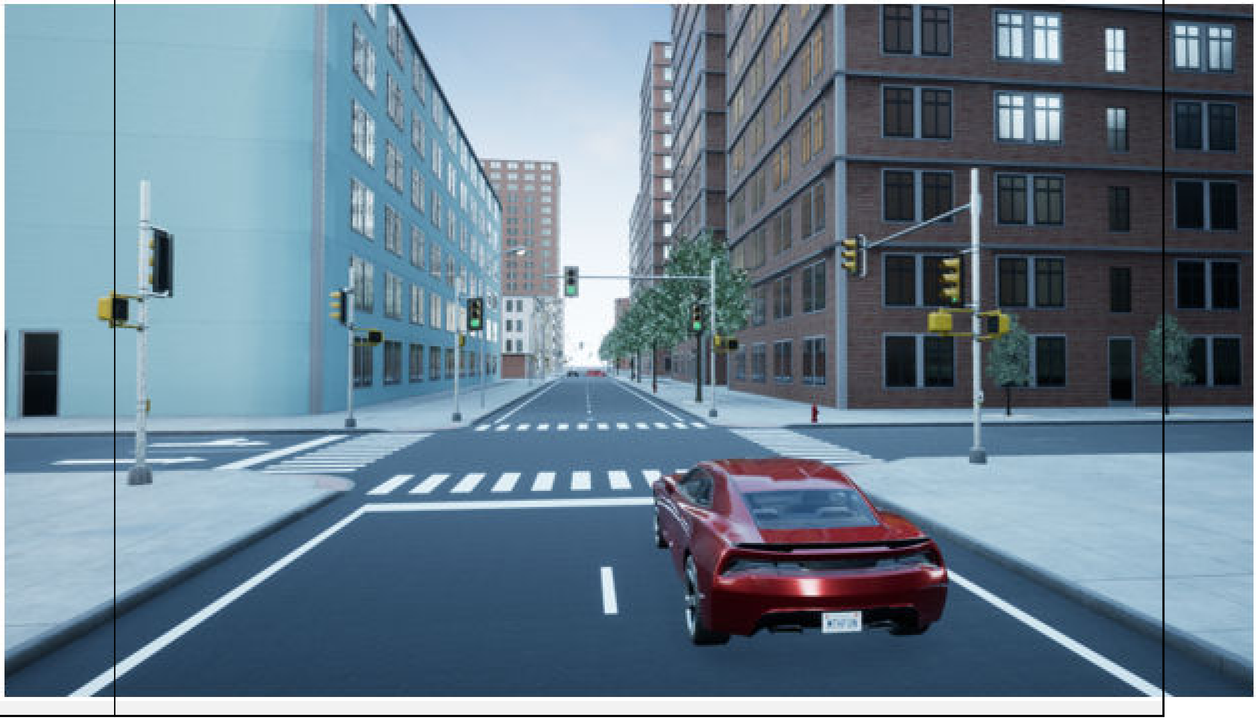
If you have Model-Based Calibration Toolbox and Simscape Multibody, you can use the kinematics and compliance virtual test laboratory to generate optimized suspension parameters for the Independent Suspension - Mapped and Solid Axle Suspension - Mapped blocks.

Using Model-Based Calibration Toolbox, the reference application performs a Sobol sequence design of experiments (DoE) on the suspension height, handwheel angle, chirp minimum and maximum frequency, chirp amplitude, and simulation time. At each operating point, the reference application stimulates the Simscape Multibody suspension system. The reference application then uses the data to fit the suspension force and camber angle to a Gaussian process model (GPM). Finally, the reference application uses the GPM to generate suspension parameters for the suspension blocks.

For more information, see Kinematics and Compliance Virtual Test Laboratory.

Unreal Engine 4 Interface: Use support package to customize and install additional 3D scenes

To customize and install additional scenes, use the Vehicle Dynamics Blockset Interface for Unreal Engine 4 Projects support package. The support package includes these scenes:

Scene	
US H	
US Ci	

For information about installing the support package, see Supporting Scene, Drive Cycle, and Maneuver Data.

Driver Blocks: Implement predictive driver control to generate commands that track lateral reference and longitudinal speed, including reverse

You can configure the Longitudinal Driver and Predictive Driver blocks to generate acceleration and braking commands that track lateral reference and longitudinal speed, including reverse. The blocks include these control type options.

Control Type Setting	Block Implementation
`PI`	Proportional-integral (PI) control with tracking windup and feed-forward gains.
`Scheduled PI`	PI control with tracking windup and feed-forward gains that are a function of vehicle velocity.
`Predictive`	Optimal single-point preview (look ahead) control model. The model represents driver steering control behavior during path-following and obstacle avoidance maneuvers. Drivers preview (look ahead) to follow a predefined path. To implement the MacAdam model, the block: • Represents the dynamics as a linear single track (bicycle) vehicle • Minimizes the previewed error signal at a single point T* seconds ahead in time • Accounts for the driver lag deriving from perceptual and neuromuscular mechanisms

The blocks include these shift type options.

Shift Type Options	Block Implementation
`None`	No transmission. Block outputs a constant gear of 1. Use this setting to minimize the number of parameters you need to generate acceleration and braking commands to track forward vehicle motion. This setting does not allow reverse vehicle motion.
`Reverse, Neutral, Drive`	Block uses a Stateflow® chart to model reverse, neutral, and drive gear shift scheduling. Use this setting to generate acceleration and braking commands to track forward and reverse vehicle motion using simple reverse, neutral, and drive gear shift scheduling. Depending on the vehicle state and vehicle velocity feedback, the block uses the initial gear and time required to shift the vehicle up into drive or down into reverse or neutral. For neutral gears, the block uses braking commands to control the vehicle speed. For reverse gears, the block uses an acceleration command to generate torque and a brake command to reduce vehicle speed.

Shift Type Options	Block Implementation
`Scheduled`	Block uses a Stateflowchart to model reverse, neutral, park, and N-speed gear shift scheduling. Use this setting to generate acceleration and braking commands to track forward and reverse vehicle motion using reverse, neutral, park, and N-speed gear shift scheduling. Depending on the vehicle state and vehicle velocity feedback, the block uses these parameters to determine the gear: • Initial gear • Upshift and downshift accelerator pedal positions • Upshift and downshift velocity • Timing for shifting and engaging forward and reverse from neutral For neutral gears, the block uses braking commands to control the vehicle speed. For reverse gears, the block uses an acceleration command to generate torque and a brake command to reduce vehicle speed.
`External`	Block uses the input gear, vehicle state, and velocity feedback to generate acceleration and braking commands to track forward and reverse vehicle motion. For neutral gears, the block uses braking commands to control the vehicle speed. For reverse gears, the block uses an acceleration command to generate torque and a brake command to reduce vehicle speed.

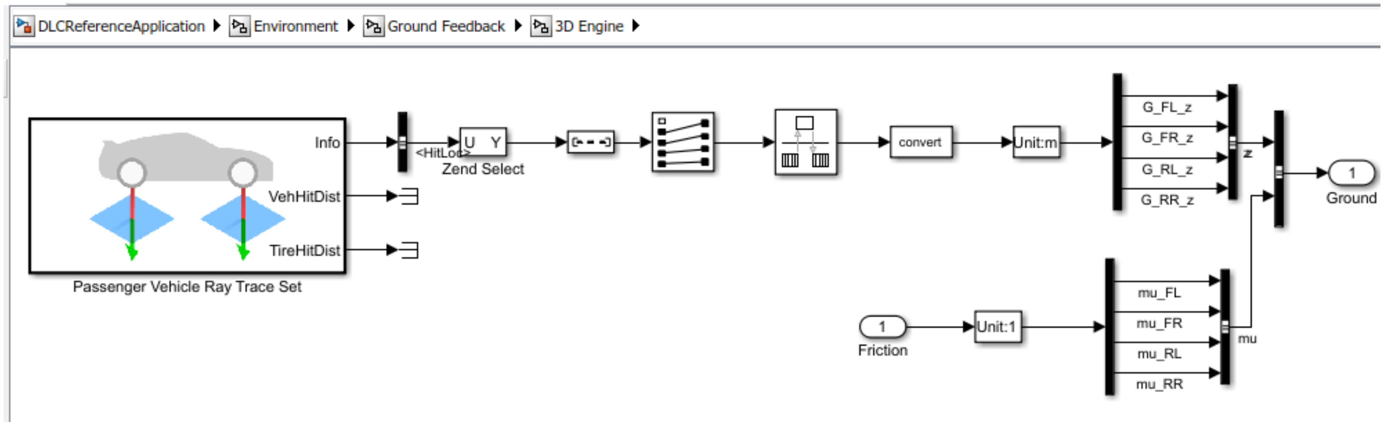
Maneuver Reference Applications: Use 3D environment ray tracing to determine ground location under tires during vehicle maneuver

To detect the terrain below the tires in the 3D visualization environment, you can configure the maneuver reference applications to use the Vehicle Terrain Sensor block. The reference application uses the terrain feedback to determine the friction forces acting on the vehicle during the maneuver.

1. Open the maneuver reference application. For example, to create and open a working copy of the double-lane change maneuver reference application, enter

   ```
   vdynblksDblLaneChangeStart
   ```
2. In the Visualization subsystem, open the 3D Engine block. Set these parameters.

   - **3D Engine** to **Enabled**.
   - **Scene description** to one of the scenes, for example `Straight road`.
3. In the Environment subsystem, open the Ground Feedback subsystem. The 3D Engine variant uses the Vehicle Terrain Sensor block to determine the ground location under the tires during a vehicle maneuver.



For more information about the maneuver reference applications, see:

- Double-Lane Change Maneuver
- Slowly Increasing Steering Maneuver
- Swept-Sine Steering Maneuver

Virtual Calibration: Use Model-Based Calibration Toolbox to calibrate SI and CI mapped engine blocks

If you have the Model-Based Calibration Toolbox, you can use measured engine data to calibrate these parameters for the Mapped CI Engine and Mapped SI Engine blocks:

- Power
- Air mass flow
- Fuel flow
- Exhaust temperature
- Efficiency
- Emissions

Maneuver Reference Applications: Display ISO 15037-1:2006 standard measurement signals after you run a maneuver

You can configure the maneuver applications to display ISO 15037-1:2006 standard measurement signals in the Simulation Data Inspector, including steering wheel angle and torque, longitudinal and lateral velocity, and sideslip angle.

To configure the ISO signal display, in the reference application Visualization subsystem, open the ISO 15037-1:2006 block. Select **Enabled**. After you run the maneuver, the Simulation Data Inspector opens with ISO standard signal measurements. For more information, see ISO 15037-1:2006 Standard Measurement Signals.

11

R2018a

Version: 1.0

New Features

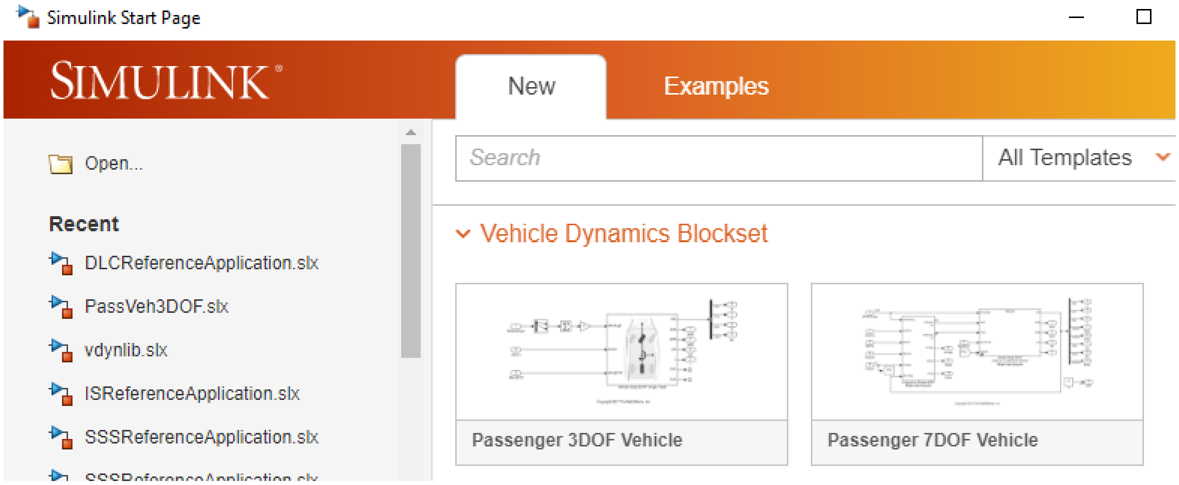
Introducing Vehicle Dynamics Blockset

Vehicle Dynamics Blockset provides fully assembled reference application models that simulate driving maneuvers in a 3D environment. You can use the prebuilt scenes to visualize roads, traffic signs, trees, buildings, and other objects around the vehicle. You can customize the reference models by using your own data or by replacing a subsystem with your own model. The blockset includes a library of components for modeling propulsion, steering, suspension, vehicle bodies, brakes, and tires.

Vehicle Dynamics Blockset provides a standard model architecture that can be used throughout the development process. It supports ride and handling analyses, chassis controls development, software integration testing, and hardware-in-the-loop testing. By integrating vehicle dynamics models with a 3D environment, you can test ADAS and automated driving perception, planning, and control software. These models let you test your vehicle with standard driving maneuvers such as a double lane change or with your own custom scenarios.

Preassembled vehicle dynamics models for passenger cars and trucks

Vehicle Dynamics Blockset provides preassembled vehicle dynamics models that you can use to analyze the dynamic system response to common ride and handling tests. Open project files that contain the vehicle models from the Simulink start page.



<table>
<tr><th>Vehicle Model</th><th>Description</th><th>Vehicle Body Degrees-of-Freedom (DOFs)</th><th>Wheel DOFs</th></tr>
<tr>
<td>Passenger 14DOF Vehicle</td>
<td><ul><li>Vehicle with four wheels</li><li>Available as model variant in the maneuver reference applications</li></ul></td>
<td>Six
<table>
<tr><th>Translational</th><th></th><th>Rotational</th><th></th></tr>
<tr><td>Longitudinal</td><td>✓</td><td>Pitch</td><td>✓</td></tr>
<tr><td>Lateral</td><td>✓</td><td>Yaw</td><td>✓</td></tr>
<tr><td>Vertical</td><td>✓</td><td>Roll</td><td>✓</td></tr>
</table></td>
<td>Two per wheel - eight total
<table>
<tr><th>Translational</th><th></th><th>Rotational</th><th></th></tr>
<tr><td>Vertical</td><td>✓</td><td>Rolling</td><td>✓</td></tr>
</table></td>
</tr>
</table>

<table>
<tr><th>Vehicle Model</th><th>Description</th><th>Vehicle Body Degrees-of-Freedom (DOFs)</th><th>Wheel DOFs</th></tr>
<tr><td>Passenger 7DOF Vehicle</td><td><ul><li>Vehicle with four wheels</li><li>Available as model variant in the maneuver reference applications</li></ul></td><td>Three<table><tr><th colspan="2">Translational</th><th colspan="2">Rotational</th></tr><tr><td>Longitudinal</td><td>✓</td><td>Pitch</td><td></td></tr><tr><td>Lateral</td><td>✓</td><td>Yaw</td><td>✓</td></tr><tr><td>Vertical</td><td></td><td>Roll</td><td></td></tr></table></td><td>One per wheel - four total<table><tr><th colspan="2">Rotational</th></tr><tr><td>Rolling</td><td>✓</td></tr></table></td></tr>
<tr><td>Passenger 3DOF Vehicle</td><td><ul><li>Vehicle with ideal tire</li></ul></td><td>Three<table><tr><th colspan="2">Translational</th><th colspan="2">Rotational</th></tr><tr><td>Longitudinal</td><td>✓</td><td>Pitch</td><td></td></tr><tr><td>Lateral</td><td>✓</td><td>Yaw</td><td>✓</td></tr><tr><td>Vertical</td><td></td><td>Roll</td><td></td></tr></table></td><td>None</td></tr>
</table>

Preassembled maneuvers for common ride and handling tests, including a double-lane change

Vehicle Dynamics Blockset provides fully assembled reference applications of vehicle models running driving maneuvers in a 3D environment that is integrated with Simulink. You can create your own versions of these maneuver applications, providing a framework to test your vehicle model and subsystems.

- Double-Lane Change Maneuver
- Slowly Increasing Steering Maneuver
- Swept-Sine Steering Maneuver

To analyze the vehicle ride and handling and develop chassis controls, use the maneuvers to test your vehicle model and subsystems. Specifically, use the maneuvers to perform vehicle studies, including:

- Friction, speed, and steering sweeps
- Yaw stability
- Understeer and oversteer
- Steering sensitivity
- Lateral acceleration limit
- Split-mu
- Driveline comparisons

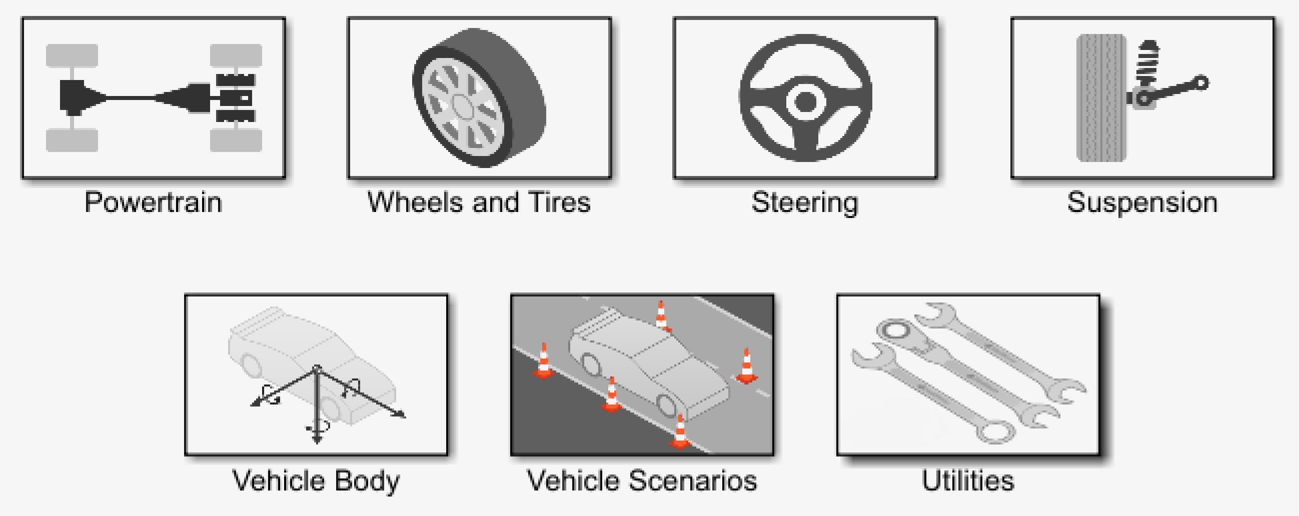
3D environment for visualizing simulations and communicating scene information to Simulink

Vehicle Dynamics Blockset integrates the 3D simulation environment with Simulink so that you can query the world around the vehicle for virtually testing perception, control, and planning algorithms. The Vehicle Dynamics Blockset visualization environment uses Unreal Engine by Epic Games.

Topic	See
How Simulink communicates with the 3D environment	Vehicle Dynamics Blockset Communication with 3D Visualization Software
3D visualization engine platform requirements, hardware recommendations, and limitations	3D Visualization Engine
Configuring the reference applications for the 3D environment	Scene Interrogation

Libraries of propulsion, steering, suspension, vehicle body, brake, and tire components

The Vehicle Dynamics Blockset provides blocks that you can use to simulate vehicles, scenarios, and scenes with vehicle body, powertrain, steering, suspension, tires, and scenario blocks. To open the block library, at the MATLAB command–line, type `vdynlib`.

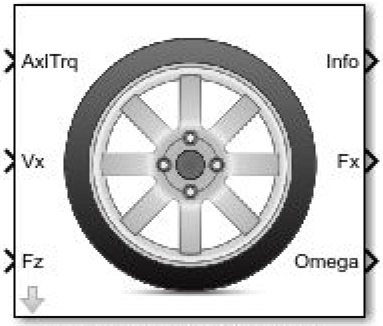


Combined longitudinal and lateral slip dynamic tire models

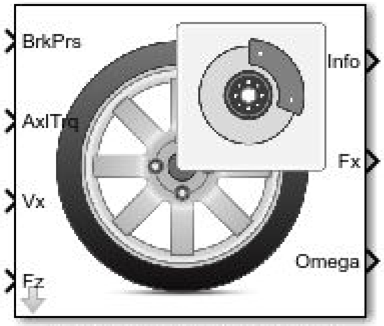
The Vehicle Dynamics Blockset has these tire blocks:

- Combined Slip Wheel 2DOF — Longitudinal and lateral tire dynamics. Use the **Brake Type** parameter to specify disc, drum, mapped, or no brakes.
- Longitudinal Wheel — Longitudinal tire dynamics with disc, drum, mapped, or no brakes.

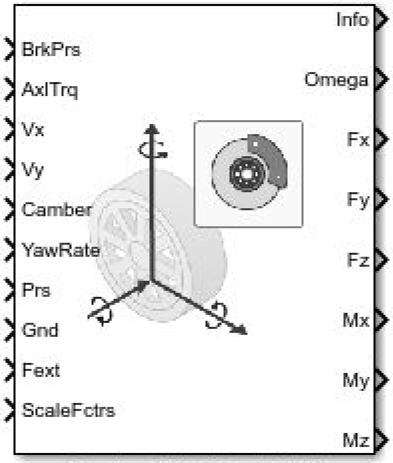
To open the block library, at the MATLAB command–line, type `vdynlib`. Open **Wheels and Tires**.



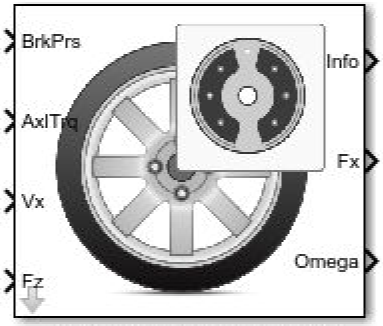
Longitudinal Wheel - No Brake



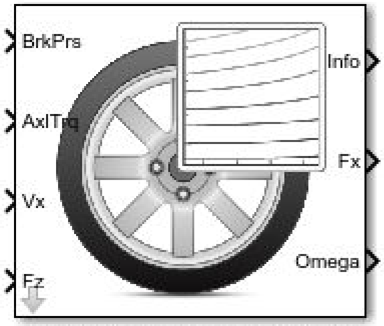
Longitudinal Wheel - Disc Brake



Combined Slip Wheel 2DOF



Longitudinal Wheel - Drum Brake



Longitudinal Wheel - Mapped Brake

Predictive driver model for generating steering commands that track a predefined path

The Predictive Driver block implements a controller that generates normalized steering, acceleration, and braking commands to track longitudinal velocity and a lateral reference displacement. Use the block to:

- Close the loop between a predefined path and actual vehicle motion.
- Generate steering commands that track predefined paths. You can connect the Predictive Driver block output to steering block inputs.
- Generate braking and acceleration commands that track a longitudinal drive cycle.

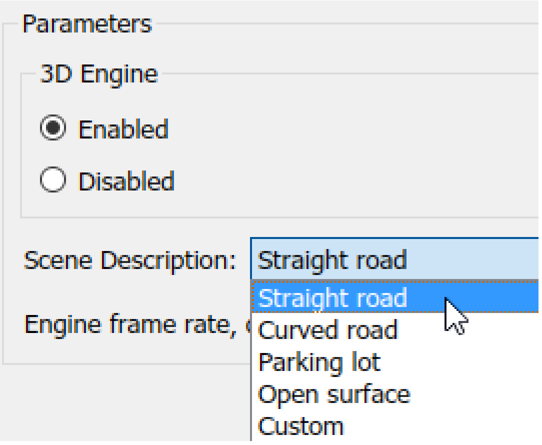
Prebuilt 3D scenes, including straight roads, curved roads, and parking lots

You can simulate driving maneuvers in prebuilt 3D scenes, including straight roads, curved roads, and parking lots. To configure the scene for a reference application driving maneuver:

1. Open a reference application. For example, open the double lane change reference application.

   ```
   vdynblksDblLaneChangeStart
   ```

2. In the Visualization subsystem, set the 3D Engine block parameter **Scene description** to one of the scenes, for example `Straight road`.



Parameters

3D Engine

● Enabled

○ Disabled

Scene Description: Straight road

Engine frame rate, (