

# Powertrain Blockset™ Release Notes



# MATLAB® & SIMULINK®



## How to Contact MathWorks



Latest news: [www.mathworks.com](http://www.mathworks.com)  
Sales and services: [www.mathworks.com/sales\\_and\\_services](http://www.mathworks.com/sales_and_services)  
User community: [www.mathworks.com/matlabcentral](http://www.mathworks.com/matlabcentral)  
Technical support: [www.mathworks.com/support/contact\\_us](http://www.mathworks.com/support/contact_us)



Phone: 508-647-7000



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

### *Powertrain Blockset™ Release Notes*

© COPYRIGHT 2016–2023 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

### **Trademarks**

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [www.mathworks.com/trademarks](http://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

### **Patents**

MathWorks products are protected by one or more U.S. patents. Please see [www.mathworks.com/patents](http://www.mathworks.com/patents) for more information.

## R2023a

<b>SI Engine: Use deep learning to generate engine model from transient data</b> .....	<b>1-2</b>
<b>Electric Vehicle Charging System: Charge an EV battery using standard protocol</b> .....	<b>1-2</b>
<b>Virtual Vehicle Composer: Configure vehicles with EV 2, 3, and 4 powertrain architectures</b> .....	<b>1-3</b>
<b>Virtual Vehicle Composer: Autogenerate static calibrations</b> .....	<b>1-3</b>
<b>Virtual Vehicle Composer: Use FMUs to integrate engines with vehicle models</b> .....	<b>1-4</b>
<b>MDF-Files: Direct access without an mdf object</b> .....	<b>1-6</b>

## R2022b

<b>Virtual Vehicle Composer App: Configure vehicles with Simscape subsystems</b> .....	<b>2-2</b>
<b>Virtual Vehicle Composer App: Resize engines and mapped motors</b> .....	<b>2-2</b>
<b>SI Engine Dynamometer: Generate a deep learning engine model for HIL testing of ECM functions</b> .....	<b>2-3</b>
<b>Engine Fault Diagnosis: Use AUTOSAR Blockset to implement on-board diagnostics and detect engine misfires</b> .....	<b>2-3</b>
<b>Functionality being removed or changed</b> .....	<b>2-4</b>
Conventional vehicle reference application performance mode .....	<b>2-4</b>

## R2022a

<b>Virtual Vehicle Composer App: Configure, build, and analyze a virtual vehicle</b> .....	3-2
<b>Motor Dynamometer Reference Application: Resize and evaluate electric motors and controllers</b> .....	3-2
<b>HEV, EV, and FCEV Reference Applications: Assess powertrain efficiency with expanded capabilities</b> .....	3-3
<b>Engine Controller Blocks: New Engine Stop-Start function</b> .....	3-3

## R2021b

<b>Fuel Cell Electric Vehicle Reference Application: Design an energy system for a hydrogen-based vehicle</b> .....	4-2
<b>Real Driving Emissions: Generate drive cycles to meet emissions standard</b> .....	4-2
<b>Electric Vehicle Reference Application: Utilize the Battery Management System and Battery Cooling System</b> .....	4-2
<b>Transfer Case Block: Implement transfer cases in drivetrain models</b> ...	4-2
<b>Simple Engine Block: Implement simple engines using basic specifications</b> .....	4-2
<b>Motorcycle Blocks: Implement in-plane longitudinal motorcycles, including chain dynamics</b> .....	4-3
<b>Simulink Model Data: Use functions that read and write block parameter data for Excel</b> .....	4-3

## R2021a

<b>Engine Fault Diagnosis: Use on-board diagnostics to detect misfire</b> ....	5-2
<b>Simscape Drivetrain Variant: Model the drivetrain using Simscape</b> ....	5-2
<b>Charge-Neutral Drive Cycles: Calibrate supervisory controller</b> .....	5-2

## R2020b

<b>New ECMS Block: Model equivalent consumption minimization strategy for hybrid electric vehicle control</b> .....	<b>6-2</b>
<b>SI Core Engine Block: Calculate cylinder pressure and brake torque as a function of crank angle</b> .....	<b>6-2</b>
<b>HEV Reference Applications: Targeted options for engine models</b> .....	<b>6-2</b>
<b>SI Reference Applications: Configure your engine model architecture, air path, and EGR</b> .....	<b>6-2</b>
<b>SI Engine Dynamometer: Resize engine using maximum torque and engine speed</b> .....	<b>6-3</b>

## R2020a

<b>Transmission Control Module: Optimize shift schedules for algorithm design and performance, fuel economy, and emissions analysis</b> .....	<b>7-2</b>
<b>Drive Cycle Fault Tracking: Identify drive cycle faults specified by standardized tests</b> .....	<b>7-2</b>
<b>Longitudinal Driver Block: Configure inputs to control acceleration and deceleration commands</b> .....	<b>7-2</b>

## R2019b

<b>SI Engine Dynamometer: Generate a deep learning engine model for algorithm design and performance, fuel economy, and emissions analysis</b> .....	<b>8-2</b>
<b>HEV P0, P1, P3, and P4 Reference Applications: Use fully assembled models for HIL testing, tradeoff analysis, and control parameter optimization of hybrid electric vehicles</b> .....	<b>8-2</b>
<b>HEV Reference Applications: Equivalent consumption minimization strategy for hybrid control</b> .....	<b>8-3</b>
<b>SI Controller Block: Use closed-loop air-fuel-ratio control to assess three-way catalyst performance, fuel economy, and emissions</b> .....	<b>8-3</b>
<b>Engine Controller Blocks: Prevent over-revving engine with rev-limiter control</b> .....	<b>8-4</b>

<b>Powertrain Efficiency: Evaluate and report energy and power losses at component and system level</b> .....	<b>9-2</b>
<b>Virtual Calibration: Use Model-Based Calibration Toolbox to calibrate Mapped Motor and Three-Phase Voltage Source Inverter block efficiency maps with measured data</b> .....	<b>9-3</b>
<b>New Inverter Block: Model a three-phase voltage source inverter</b> .....	<b>9-4</b>
<b>Motor Blocks: Configure blocks for discrete sample time to generate code for single-precision targets</b> .....	<b>9-4</b>
<b>Mapped Engine Blocks: Account for engine temperature effects on fuel economy, performance, and emissions</b> .....	<b>9-4</b>
<b>Open-Loop Driver: Configure reference application driver commands with constant or signal-based inputs</b> .....	<b>9-5</b>
<b>Longitudinal Wheel Block: Configure block for ISO 28580 rolling resistance calculation</b> .....	<b>9-6</b>
<b>Vehicle Body Blocks: Configuration options for external inputs, including forces, moments, and air temperature</b> .....	<b>9-6</b>
Compatibility Considerations .....	<b>9-6</b>
<b>Differential Blocks: Account for efficiency as a function of torque, temperature, and driveshaft speed</b> .....	<b>9-7</b>
Compatibility Considerations .....	<b>9-7</b>

<b>Virtual Calibration: Use Model-Based Calibration Toolbox to calibrate spark-ignition and compression-ignition engine model blocks</b> .....	<b>10-2</b>
<b>HEV P2 Reference Application: Use fully assembled model for HIL testing, tradeoff analysis, and control parameter optimization of a P2 hybrid electric vehicle</b> .....	<b>10-2</b>
<b>Compression-Ignition Engine: Model engine torque and exhaust temperature as a function of multiple fuel injections, fuel pressure, and air system states</b> .....	<b>10-2</b>
Compatibility Considerations .....	<b>10-3</b>
<b>Spark-Ignition Engine Startup: Use SI Controller block to account for fuel consumption due to rich operation during engine startup</b> .....	<b>10-3</b>

<b>Longitudinal Driver Block: Implement predictive driver control and specify gear schedule to track longitudinal speed, including reverse</b> .....	<b>10-3</b>
<b>CI and SI Engine Dynamometer Reference Applications: Resize engines and recalibrate controllers based on desired engine displacement and number of cylinders</b> .....	<b>10-5</b>
<b>Electric Vehicle Reference Application: Implements dynamic supervisory control to determine torque and regenerative braking commands</b> ..	<b>10-5</b>

## R2018a

<b>Engine Data Import: Use engine test data to calibrate mapped engines in CI and SI engine dynamometer reference applications</b> .....	<b>11-2</b>
<b>Low-Pressure EGR: Default SI engine includes low-pressure exhaust gas recirculation</b> .....	<b>11-2</b>
Compatibility Considerations .....	<b>11-2</b>
<b>Gear Shift Schedule: Use Drive Cycle block to specify gear shift schedule</b> .....	<b>11-2</b>
<b>Transmission Temperature Dependency: Include effects of oil temperature when calculating efficiency loss in transmission blocks</b> .....	<b>11-3</b>
<b>Engine Torque Table: Mapped CI Engine block now uses either engine torque or injected fuel mass lookup table</b> .....	<b>11-3</b>

## R2017b

<b>HEV Input Power-Split Reference Application: Use fully assembled model for HIL testing, tradeoff analysis, and control parameter optimization of a power-split hybrid like the Toyota Prius</b> .....	<b>12-2</b>
<b>CI and SI Engine Dynamometer Reference Applications: Resize engines and recalibrate controllers based on desired power and number of cylinders</b> .....	<b>12-2</b>
<b>New Drivetrain and Propulsion Blocks: Model drivetrain coupling, bidirectional DC-to-DC energy conversion, and flux-based PMSM motors</b> .....	<b>12-2</b>
<b>Electric Motor Controllers: Calculate inverter electrical losses</b> .....	<b>12-3</b>

<b>Flux-Based Motor Parameterization: Generate parameters for Flux-Based PMSM and Flux-Based PM Controller blocks</b> .....	12-3
<b>Longitudinal Wheel Block: Model tires using Magic Formula longitudinal slip, vertical motion, and external friction input</b> .....	12-4
<b>CI Core Engine Block: Customize fuel injection with fuel mass input</b> .....	12-4
Compatibility Considerations .....	12-5
<b>Combustion Engine Reference Applications: Use control actuator IO during HIL testing</b> .....	12-5
<b>Battery Blocks: Input rated capacity at nominal temperature</b> .....	12-6
Compatibility Considerations .....	12-6
<b>Mapped Engine Blocks: Model turbocharger lag</b> .....	12-6
<b>Conventional and Hybrid Electric Vehicle Reference Applications: Idle the engine until catalyst light-off</b> .....	12-7
<b>Combustion Engine Torque: Independent friction and pumping loss calculation</b> .....	12-7
Compatibility Considerations .....	12-8
<b>Ports and Signals: Consistent names and units across library blocks</b> ..	12-8
Compatibility Considerations .....	12-8

## R2017a+

### Bug Fixes

## R2017a

<b>Wide Open Throttle (WOT) Drive Cycle: Generate WOT drive cycle signals for performance testing</b> .....	14-2
<b>Engine-Out (EO) Emissions: Track EO emissions using combustion engine blocks</b> .....	14-2
Core Engine Blocks .....	14-2
Two-way Connector Ports .....	14-2
<b>Hybrid Electric Vehicle Reference Application: Minimize drive cycle fuel consumption using a dynamic embedded optimizer</b> .....	14-3
<b>Drivetrain Blocks: Model drivetrain coupling with Disc Clutch, Gearbox, and Planetary Gear blocks</b> .....	14-3



<b>Battery Parameterization: Generate parameters for Datasheet Battery and Equivalent Circuit Battery blocks</b> .....	<b>14-3</b>
<b>CI Core Engine Block: Track exhaust gas recirculation via two-way connector port</b> .....	<b>14-3</b>
Compatibility Considerations .....	<b>14-3</b>

## **R2016b+**

<b>Introducing Powertrain Blockset</b> .....	<b>15-2</b>
<b>Fully assembled models for gasoline, diesel, hybrid, and electric powertrains</b> .....	<b>15-2</b>
<b>Libraries for engine, transmission, traction motor, and battery components</b> .....	<b>15-2</b>
<b>Basic controllers for powertrain subsystems</b> .....	<b>15-3</b>
<b>Standard drive cycle data, including FTP-75, NEDC, and JC08</b> .....	<b>15-3</b>
<b>Engine dynamometer models for virtual calibration and testing</b> .....	<b>15-3</b>
<b>MDF file support for calibration data import</b> .....	<b>15-4</b>



# R2023a

---

**Version: 2.0**

**New Features**

**Bug Fixes**

## SI Engine: Use deep learning to generate engine model from transient data

You can use Deep Learning Toolbox™ and Statistics and Machine Learning Toolbox™ to generate a dynamic (transient) deep learning spark-ignition (SI) engine model to use for hardware-in-the-loop (HIL) testing, powertrain control design, on-board diagnostic calibration, and estimator algorithm design. For example, from physical hardware or high-fidelity model data, you can fit a deep learning model to measured engine performance data and use the model for aftertreatment management and closed-loop air/fuel ratio control algorithm development.

You can generate a deep learning SI engine model in two different ways:

- 1 Open the “SI Engine Dynamometer Reference Application”. Click **Generate Deep Learning Engine Model**.

For more information, see “Generate Deep Learning SI Engine Model”.

- 2 Start the **Virtual Vehicle Composer** app.

On the **Setup** tab, select any hybrid or conventional vehicle architecture, and click **Configure**.

On the **Data and Calibration** tab, select **Powertrain > Engine**.

From the **Engine** list, select **SI Deep Learning Engine**.

On the **Calibrate from Data** tab, select a MATLAB® binary file containing your measured engine performance data and click **Calibrate**.

Generating the deep learning model can take several hours.

In R2023a, the deep learning engine model no longer requires support from a physical throttle model, so it only requires input and output data to be trained. The application uses seven inputs to train the model. Throttle position (%) is now an input. Intake manifold gas pressure, formerly an input, is now an output. Throttle inlet pressure, formerly an output, is no longer required for training. This table shows the required training data.

Input Data	Output Data
<ul style="list-style-type: none"> <li>• Throttle position (%)</li> <li>• Wastegate area (%)</li> <li>• Engine speed (RPM)</li> <li>• Intake cam advance from park (deg)</li> <li>• Exhaust cam retard from park (deg)</li> <li>• Spark retard from nominal (deg)</li> <li>• Lambda — Air/Fuel ratio, normalized to stoichiometric Air/Fuel ratio</li> </ul>	<ul style="list-style-type: none"> <li>• Airflow — Intake air mass flow rate (kg/s)</li> <li>• Torque (N·m)</li> <li>• Intake manifold gas pressure (Pa)</li> <li>• Exhaust gas temperature (K)</li> </ul>

## Electric Vehicle Charging System: Charge an EV battery using standard protocol

Starting in R2023a, you can use the EV charging reference application to model DC charging of an electric vehicle (EV) battery from an electric vehicle supply equipment (EVSE) unit. The application

uses SAE J1772 protocol for basic charging and ISO15118 protocol for digital communication and incorporates a mapped battery block. Use the EV charging reference application for control and diagnostic algorithm design, and testing of digital communication protocols. To create and open a working copy of the reference application project, enter this command:

```
autoblkEvChargingStart
```

For more information, see “Charge an Electric Vehicle”.

## Virtual Vehicle Composer: Configure vehicles with EV 2, 3, and 4 powertrain architectures

**Virtual Vehicle Composer** app has four additional electric vehicle (EV) configurations. On the **Setup** tab, the **Powertrain architecture** list provides these EV options.

Powertrain Architecture	Description
Electric Vehicle 1EM	Single motor driving front axle, rear axle, or both
Electric Vehicle 2EM	Single motor driving front axle, and single motor driving rear axle
Electric Vehicle 3EM Dual-Front	Two independent motors in front, single motor in rear
Electric Vehicle 3EM Dual-Rear	Single motor in front, two independent motors in rear
Electric Vehicle 4EM	Four independent motors.

Previously, you could only select the Electric Vehicle 1EM architecture.

## 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
- DC — DC converter

Follow these step to calibrate your block:

Block to Calibrate	Options
CI Mapped Engine	<ul style="list-style-type: none"> <li>• On the <b>Setup</b> tab, select Conventional Vehicle.</li> <li>• On the <b>Data and Calibration</b> tab, select <b>Powertrain &gt; Engine</b>.</li> <li>• In the <b>Engine</b> list, select CI Mapped Engine.</li> <li>• Select Calibrate from Data.</li> </ul> <p>For more information, see “Calibrate Mapped CI Engine Using Data”.</p>

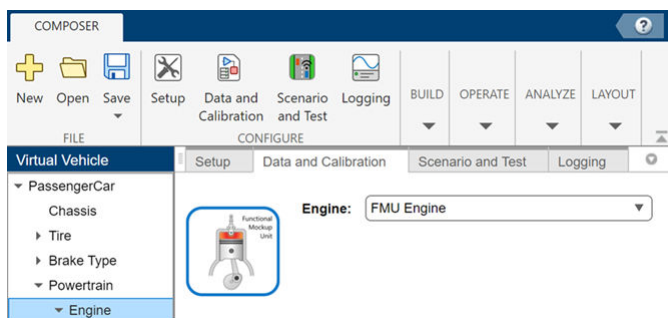
Block to Calibrate	Options
SI Mapped Engine	<ul style="list-style-type: none"> <li>On the <b>Setup</b> tab, select Hybrid Electric Vehicle P2.</li> <li>On the <b>Data and Calibration</b> tab, select <b>Powertrain &gt; Engine</b>.</li> <li>In the <b>Engine</b> list, select SI Mapped Engine.</li> <li>Select Calibrate from Data.</li> </ul> <p>For more information, see “Calibrate Mapped SI Engine Using Data”</p>
Mapped Electric Motor  In building an electric or hybrid electric vehicle variant, you can calibrate each motor.  This example is for the second motor of a three-motor electric vehicle.	<ul style="list-style-type: none"> <li>On the <b>Setup</b> tab, select Electric Vehicle 3EM Dual Front.</li> <li>On the <b>Data and Calibration</b> tab, select <b>Powertrain &gt; Electrical System &gt; Electric Machine 2</b>.</li> <li>In the <b>Electric Machine 2</b> list, select Electric Vehicle 3EM Dual Front.</li> <li>Select Calibrate from Data.</li> </ul> <p>For more information, see “Calibrate Mapped Electric Motors Using Data”.</p>
DC – DC Converter  This example is for the DC – DC converter of a three-motor electric vehicle.	<ul style="list-style-type: none"> <li>On the <b>Setup</b> tab, select Electric Vehicle 3EM Dual Front.</li> <li>On the <b>Data and Calibration</b> tab, select <b>Powertrain &gt; Electrical System &gt; DC – DC Converter</b>.</li> <li>In the <b>DC – DC Converter</b> list, select DC – DC Converter.</li> <li>Select Calibrate from Data.</li> </ul> <p>For more information, see “Calibrate Mapped DC-to-DC Converters Using Data”.</p>

Static calibrations require the Model-Based Calibration Toolbox™.

## Virtual Vehicle Composer: Use FMUs to integrate engines with vehicle models

You can use the **Virtual Vehicle Composer** app to configure vehicles with engine models imported as a functional mockup unit (FMU). The FMU engine option integrates your external engine model with the virtual vehicle model.

When you configure your virtual vehicle, on the **Data and Calibration** tab, select **Powertrain > Engine**. Set **Engine** to FMU Engine.



The FMU engine implements an FMU block with these engine inputs and outputs.

Inputs	Outputs
Torque command	Brake torque
Engine RPM	Fuel flow
	Air flow
	Exhaust gas temperature
	Exhaust gas temperature
	Air fuel ratio
	Brake-specific fuel consumption (BSFC)
	Crank angle

To implement the FMU engine model, follow these steps:

- 1 Set **Engine** to FMU Engine.
- 2 Use **Browse** to select the FMU file.
- 3 Select **Read** to verify the FMU inputs and outputs.
  - If verification passes, the number of FMU inputs and outputs matches the signals in the FMU Import subsystem.
  - If verification warns, the number of FMU inputs and outputs does not match the signals in the FMU Import subsystem. However, you can still import the FMU file and manually connect the signals.
- 4 Select **Import** to integrate the FMU in the virtual vehicle FMU Import subsystem.

Verification Status	FMU Import Subsystem
Pass	
Warn	

## **MDF-Files: Direct access without an mdf object**

The following functions are available to provide direct access to MDF-files without creating an `mdf` object in the workspace.

- `mdfInfo` — Access or create MDF-file metadata
- `mdfChannelGroupInfo` — Get channel group metadata from MDF-file
- `mdfChannelInfo` — Get channel metadata from MDF-file
- `mdfRead` — Read channel data from MDF-file
- `mdfCreate` — Create MDF-file with metadata
- `mdfWrite` — Write timetable data to MDF-file
- `mdfAddChannelGroupMetadata` — Add channel group and channel metadata to timetable
- `mdfSort` — Sort MDF-file by ASAM standards
- `mdfFinalize` — Finalize MDF-file by ASAM standards
- `mdfSaveAttachment` — Save embedded attachment files from MDF-file
- `mdfAddAttachment` — Attach file to MDF-file
- `mdfRemoveAttachment` — Remove attachment from MDF-file



# R2022b

---

**Version: 1.12**

**New Features**

**Bug Fixes**

**Compatibility Considerations**

## 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.

The screenshot shows the 'Setup' tab of the Virtual Vehicle Composer app. The configuration fields are as follows:

- Project name:** VVProj
- Project folder:** C:\work\VirtualVehicle
- Model name:** VirtualVehicleConfModel
- Vehicle class:** Two icons representing different vehicle types are shown.
- Powertrain archite...:** Conventional Vehicle (dropdown menu)
- Model template:** A dropdown menu with three options: Simscape (selected), Simulink, and Simscape. A mouse cursor is pointing at the bottom 'Simscape' option.

Below the 'Powertrain archite...' dropdown, there is a small schematic diagram of a powertrain system, showing components like an engine, transmission, and motor connected to wheels.

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

## Virtual Vehicle Composer App: Resize engines and mapped motors

Use the **Virtual Vehicle Composer** app to:

- Resize an engine based on desired maximum engine power or engine displacement.
- Resize a motor based on high-level specifications.

For more information, see [Resize Engines and Mapped Motors](#)

## SI Engine Dynamometer: Generate a deep learning engine model for HIL testing of ECM functions

If you have Deep Learning Toolbox and Statistics and Machine Learning Toolbox, you can generate a dynamic deep learning spark-ignition (SI) engine model for hardware-in-the-loop (HIL) testing of engine control module (ECM) functions. The deep learning SI engine models the dynamic engine behavior from measured laboratory data or a high-fidelity engine model. Additionally, you can use data measured in engine dynamometer laboratory transient testing, avoiding development of a physical simulation model of your engine behavior.

Previously, the software used two engine plant model inputs to train a deep learning engine plant model. In R2022b, the software uses seven inputs to train the new deep learning engine plant model, making it more suitable for HIL tests.

Input Data	Output Data
<ul style="list-style-type: none"> <li>Engine speed</li> </ul> <p><i>New in R2022b:</i></p> <ul style="list-style-type: none"> <li>Intake manifold gas pressure (<i>moved from output data</i>)</li> <li>Wastegate area percent</li> <li>Intake cam phaser angle</li> <li>Exhaust cam phaser angle</li> <li>Spark retard from nominal</li> <li>Lambda</li> </ul> <p><i>Removed in R2022b:</i></p> <ul style="list-style-type: none"> <li>Commanded torque</li> </ul>	<ul style="list-style-type: none"> <li>Torque</li> <li>Airflow - Intake air mass flow</li> <li>Exhaust gas temperature</li> </ul> <p><i>New in R2022b:</i></p> <ul style="list-style-type: none"> <li>Throttle inlet pressure</li> </ul> <p><i>Removed in R2022b:</i></p> <ul style="list-style-type: none"> <li>Intake manifold gas pressure (<i>moved to input data</i>)</li> <li>Intake manifold gas temperature</li> <li>Fuel flow (derived from airflow and lambda)</li> <li>Turbocharger speed</li> <li>Engine out (EO) hydrocarbon (HC) emission mass flow</li> <li>EO carbon monoxide (CO) emission mass flow</li> <li>EO nitric oxide and nitrogen dioxide emissions (NOx) emission mass flow</li> <li>EO carbon dioxide (CO2) emission mass flow</li> </ul>

To generate the deep learning SI engine model, open the SI Engine Dynamometer Reference Application. Click **Generate Deep Learning Engine Model**. Generating the deep learning model can take some time.

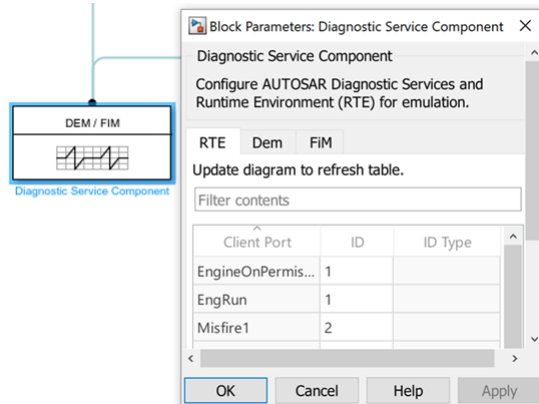
After you generate the deep learning SI model, you can use it as an engine plant model variant in the conventional vehicle and hybrid electric vehicle (HEV) reference applications.

For more information, see [Generate a Deep Learning SI Engine Model](#).

## Engine Fault Diagnosis: Use AUTOSAR Blockset to implement on-board diagnostics and detect engine misfires

Starting from R2022b, the Detect Misfires Using On-Board Diagnostics example includes the AUTOSAR Blockset Diagnostic Service Component block. The Diagnostic Service Component

(AUTOSAR Blockset) block manages the on-board diagnostics (OBD), simplifying the detection and reporting of engine misfires. Previously, the example did not include the Diagnostic Services Component block, requiring subsystems to implement the OBD.



## Functionality being removed or changed

### Conventional vehicle reference application performance mode

#### *Behavior change*

Starting from R2022b, the conventional vehicle reference application no longer has the performance mode transmission controller variant for maximum power, `PowertrainMaxPowerController`. For maximum power, use the `PowertrainPRNDLController` variant. For more information, see [Build a Conventional Vehicle Model](#).

# R2022a

---

**Version: 1.11**

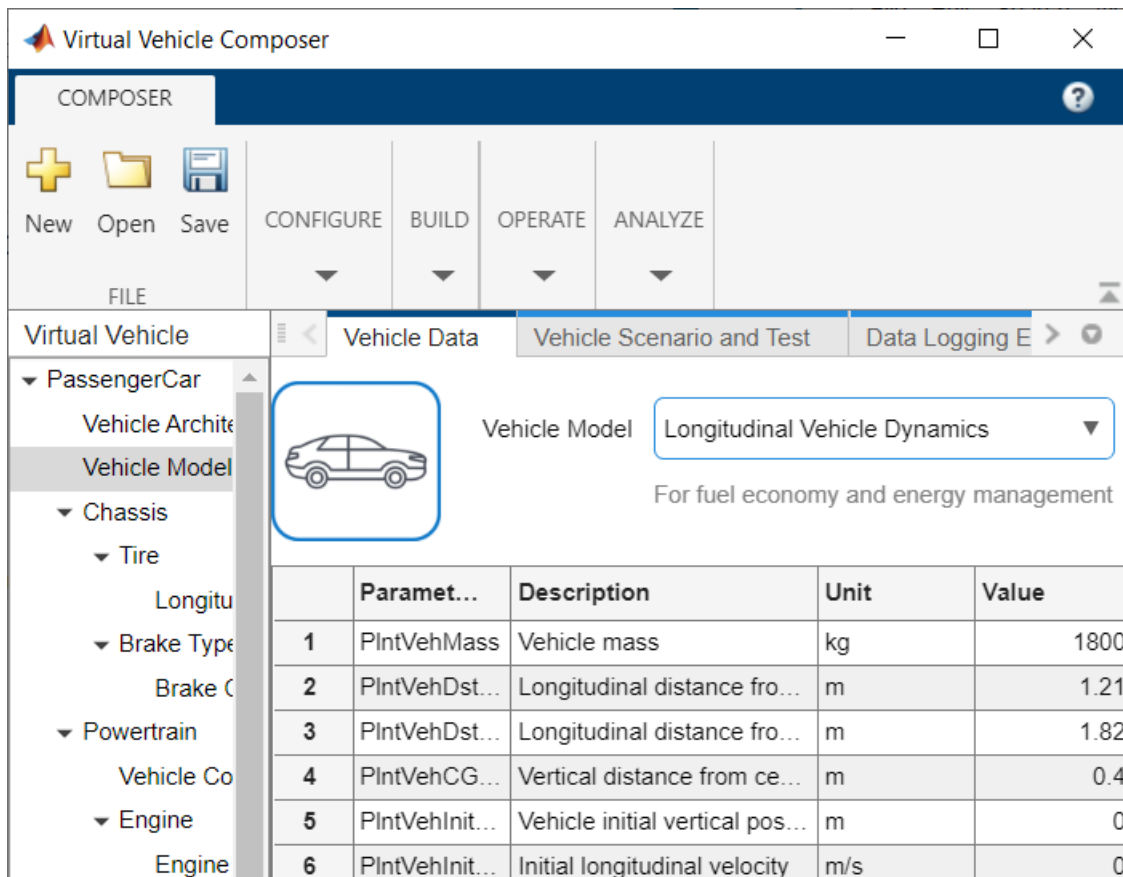
**New Features**

**Bug Fixes**

**Compatibility Considerations**

## 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 component sizing, fuel economy, and drive cycle tracking. To get started, see [Get Started with the Virtual Vehicle Composer](#).



## Motor Dynamometer Reference Application: Resize and evaluate electric motors and controllers

Use the Motor Dynamometer Reference Application to design and test traction electric motors (e-motors) and controllers for your electrified powertrain vehicle. Three designs are available:

- Mapped e-motor
- Dynamic e-motor and controller
- Flux-based e-motor and controller

You can Resize the Motor to automatically scale a combined e-motor and controller system for use in vehicle applications. Use these models for design tradeoff analysis and component sizing, control parameter optimization, and hardware-in-the-loop (HIL) testing.

---

## **HEV, EV, and FCEV Reference Applications: Assess powertrain efficiency with expanded capabilities**

Use the **Analyze Power and Energy** function to evaluate and report power and energy for all variants of the Hybrid Electric Vehicle P0-P4 Reference Applications, the Electric Vehicle Reference Application, and the Fuel Cell Electric Vehicle Reference Application.

For more information, see Hybrid and Electric Vehicle Reference Application Projects.

## **Engine Controller Blocks: New Engine Stop-Start function**

The CI Controller and SI Controller blocks implement a function that models an engine stop-start feature. To enable this function, on the **Stop-Start** tab of the block mask, select **Enable Engine Stop-Start**. You can optionally activate the function using an input signal.

## **Compatibility Considerations**

Starting in R2022a, the Start Stop Logic block has been removed from the CI Engine Dynamometer Reference Application and the SI Engine Dynamometer Reference Application. This block is no longer needed since the function is integrated into the engine controllers. If you have existing projects based on these reference applications, consider upgrading to this version.





# R2021b

---

**Version: 1.10**

**New Features**

**Bug Fixes**

## **Fuel Cell Electric Vehicle Reference Application: Design an energy system for a hydrogen-based vehicle**

To design an energy system for a hydrogen-based vehicle, use the fuel cell reference application project with a high-fidelity fuel cell model in Simscape. You can switch between a detailed and a mapped fuel cell. Use these models for design tradeoff analysis and component sizing, control parameter optimization, and hardware-in-the-loop (HIL) testing. For more information, see [Explore the Fuel Cell Electric Vehicle Reference Application](#).

Within this project, you can use the Model-Based Calibration Toolbox to create a mapped fuel cell model from measured fuel cell performance data stored in a spreadsheet. For more information, see [Generate Mapped Fuel Cell from a Spreadsheet](#).

## **Real Driving Emissions: Generate drive cycles to meet emissions standard**

To generate a Real Driving Emissions (RDE)-compliant trip, use the example [Generate Drive Cycles for Real Driving Emissions](#). RDE is an engine emissions standard which was first introduced in the European Union. It requires that pollutant emissions are measured while driving the car on a real road.

## **Electric Vehicle Reference Application: Utilize the Battery Management System and Battery Cooling System**

Utilize the new Battery Management System and a detailed battery plant model, along with a new Battery Cooling System. The vehicle parameters have been updated to represent design specifications from the latest generation of electric vehicles.

For more information, see [Explore the Electric Vehicle Reference Application](#).

## **Transfer Case Block: Implement transfer cases in drivetrain models**

The Drivetrain library now contains a Transfer Case block that 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 Block: Implement simple engines using basic specifications**

The Propulsion library now contains a Simple Engine block that implements a simplified engine model using a maximum torque vs engine speed table, two scalar fuel mass properties, and one scalar engine efficiency parameter to estimate engine torque and fuel flow.

Use the Simple Engine block 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.

---

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

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

Block	Description
Motorcycle Body Longitudinal In-Plane	<p>Implements a longitudinal in-plane motorcycle body model to calculate longitudinal, vertical, and pitch motion. The block accounts for:</p> <ul style="list-style-type: none"><li>• Mass of the frame, rear arm, front upper fork, front lower fork, front wheel, and rear wheel</li><li>• 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</li><li>• External forces, external moments, and aerodynamic drag</li><li>• Road incline</li><li>• Weight distribution between the axles due to acceleration</li></ul>
Motorcycle Chain	<p>Implements dynamic effects of motorcycle chain on Motorcycle Body Longitudinal In-Plane block, including dynamic tension and moment drive coupling.</p>

## 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.



# R2021a

---

**Version: 1.9**

**New Features**

**Bug Fixes**

## **Engine Fault Diagnosis: Use on-board diagnostics to detect misfire**

The Use On-Board Diagnostics to Detect Misfire example allows you to introduce, detect, mature, and report misfire events from the SI Core Engine. The model demonstrates misfire on-board diagnostics (OBD) warnings. Customize this example by modifying these parameters:

- OBD threshold calibrations for detection and maturation
- Misfire fault injection timing

## **Simscape Drivetrain Variant: Model the drivetrain using Simscape**

New variants are available to demonstrate how subsystems based in Simscape can be incorporated into a Powertrain Blockset reference application. Use the button in the top level of the HEV reference application to toggle between Simscape and Powertrain Blockset variants of the drivetrain subsystem.

For more information, see:

- Explore the Hybrid Electric Vehicle P0 Reference Application
- Explore the Hybrid Electric Vehicle P1 Reference Application
- Explore the Hybrid Electric Vehicle P2 Reference Application
- Explore the Hybrid Electric Vehicle P3 Reference Application
- Explore the Hybrid Electric Vehicle P4 Reference Application

## **Charge-Neutral Drive Cycles: Calibrate supervisory controller**

Follow the steps in the Calibrate ECMS Block example to match starting and ending battery state of charge (SOC) over a drive cycle. You can modify the script to calibrate the supervisory controller in other HEV reference applications.

# R2020b

---

**Version: 1.8**

**New Features**

**Bug Fixes**

## **New ECMS Block: Model equivalent consumption minimization strategy for hybrid electric vehicle control**

The hybrid electric vehicle (HEV) P0, P1, P2, P3, and P4 reference applications include the Equivalent Consumption Minimization Strategy (ECMS) block for hybrid control. The ECMS block optimizes the torque split between the engine and motor to minimize energy consumption while maintaining the battery state of charge (SOC). For more information, see:

- Equivalent Consumption Minimization Strategy
- Hybrid and Electric Vehicle Reference Application Projects

## **SI Core Engine Block: Calculate cylinder pressure and brake torque as a function of crank angle**

You can use the enhanced SI Core Engine block to:

- Simulate advanced closed-loop engine controls in desktop simulations and on HIL bench, based on cylinder pressure recorded from a model or laboratory test as a function of crank angle.
- Simulate driveline vibrations downstream of the engine due to high-frequency crankshaft torsionals.
- Simulate engine misfires due to lean operation or spark plug fouling by using the injector pulse width input.
- Simulate cylinder deactivation effect (closed intake and exhaust valves, no injected fuel) on individual cylinder pressures, mean-value airflow, mean-value torque, and crank-angle-based torque.
- Simulate the fuel-cut effect on individual cylinder pressure, mean-value torque, and crank-angle-based torque.

To activate these calculations for the SI Core Engine, select **Crank angle pressure and torque** on the block **Torque** tab.

## **HEV Reference Applications: Targeted options for engine models**

The engine variants available for the hybrid electric vehicle (HEV) P0, P1, P2, P3, and P4 reference applications have been reduced to the following SI engine selections to match real-world applications:

- Naturally aspirated SI Engine
- Turbocharged SI Engine
- Mapped SI Engine (default)

## **SI Reference Applications: Configure your engine model architecture, air path, and EGR**

You can use the enhanced SI Engine Dynamometer, Conventional Vehicle, and HEV Multimode reference applications to choose:

- Engine architecture (Inline or V)



- 
- Air path configuration (turbocharged or naturally aspirated)
  - Presence or absence of cooled exhaust gas re-circulation (EGR)

To use the new variants, open one of the reference applications listed above. Navigate to the SiEngine. Right-click the Dynamic SI Engine, hover over **Variant**, then hover over **Label Mode Active Choice**. Click the desired engine variant from the list.

## **SI Engine Dynamometer: Resize engine using maximum torque and engine speed**

You can configure the architecture, air path configuration, and presence or absence of cooled exhaust gas re-circulation (EGR) of your engine model.

When resizing the engine, you can define the maximum torque and the engine speed at which maximum torque occurs. As in previous versions, you can define the displacement and number of cylinders.

To get started, open the SI Engine Dynamometer Reference Application. Click **Resize Engine and Recalibrate Controller**.



# R2020a

---

**Version: 1.7**

**New Features**

**Bug Fixes**

**Compatibility Considerations**

## Transmission Control Module: Optimize shift schedules for algorithm design and performance, fuel economy, and emissions analysis

You can use the conventional vehicle reference application to optimize the transmission control module (TCM) shift schedules. Use the optimized shift schedules to:

- Design control algorithms.
- Assess the impact of powertrain changes, such as an engine or gear ratio, on performance, fuel economy, and emissions.

TCM shift schedule optimization requires Simulink Design Optimization™, the Global Optimization Toolbox, and Stateflow®. To increase the performance of the optimization, consider also using the Parallel Computing Toolbox™.

To run the TCM shift schedule optimization, open a version of the conventional vehicle reference application that includes the option to optimize transmission shift maps by using this command:

```
autoblkConVehShftOptStart
```

Click **Optimize Transmission Shift Maps**. Optimizing the shift schedules can take time to run.

For more information, see [Optimize Transmission Control Module Shift Schedules](#).

## 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 (WLTP) 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. For more information, see [Track Drive Cycle Errors](#).

## Longitudinal Driver Block: Configure inputs to control acceleration and deceleration commands

The Longitudinal Driver block allows you to configure inputs that control acceleration and deceleration commands. Specifically, use the **External action** parameters to create input ports that:

- 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.

---

## **Compatibility Considerations**

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



# R2019b

---

**Version: 1.6**

**New Features**

**Bug Fixes**

**Compatibility Considerations**

## SI Engine Dynamometer: Generate a deep learning engine model for algorithm design and performance, fuel economy, and emissions analysis

If you have the Deep Learning Toolbox and Statistics and Machine Learning Toolbox licenses, you can generate a dynamic deep learning spark-ignition (SI) engine model to use for powertrain control, diagnostic, and estimator algorithm design. For example, use the deep learning model for aftertreatment control and diagnostic algorithm development. The deep learning SI engine models the dynamic engine behavior from measured laboratory data or a high-fidelity engine model.

To generate the deep learning SI engine model, open the SI Engine Dynamometer Reference Application. Click **Generate Deep Learning Engine Model**. Generating the deep learning model can take time to run.

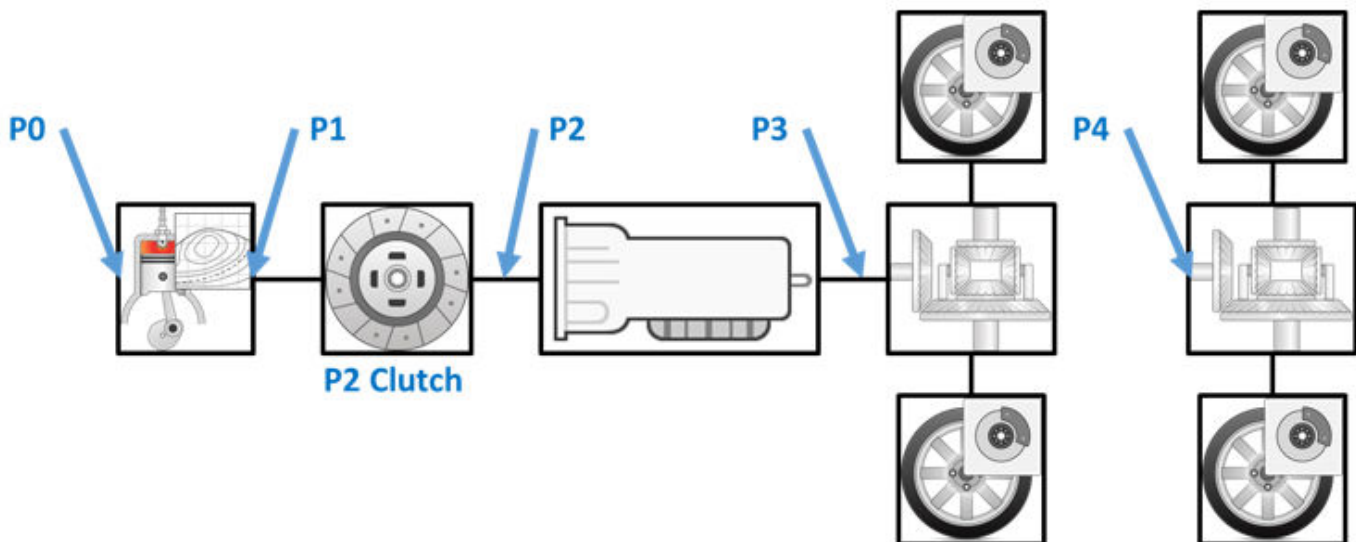
After you generate the deep learning SI model, you can use it as an engine plant model variant in the conventional vehicle and hybrid electric vehicle (HEV) reference applications.

For more information, see [Generate a Deep Learning SI Engine Model](#).

## HEV P0, P1, P3, and P4 Reference Applications: Use fully assembled models for HIL testing, tradeoff analysis, and control parameter optimization of hybrid electric vehicles

Powertrain Blockset includes hybrid electric vehicle (HEV) P0, P1, P3, and P4 reference applications that you can use for design tradeoff analysis and component sizing, control parameter optimization, and hardware-in-the-loop (HIL) testing. Previously, the Powertrain Blockset included only an HEV P2 reference application.

This diagram shows the powertrain motor locations for the HEV reference applications.



This table provides links to open a working copy of the reference application project and access more information.



Reference Application	MATLAB Command	More Information
HEV P0	autoblkHevP0Start	Explore the Hybrid Electric Vehicle P0 Reference Application
HEV P1	autoblkHevP1Start	Explore the Hybrid Electric Vehicle P1 Reference Application
HEV P3	autoblkHevP3Start	Explore the Hybrid Electric Vehicle P3 Reference Application
HEV P4	autoblkHevP4Start	Explore the Hybrid Electric Vehicle P4 Reference Application

In addition to the new HEV reference applications, the Powertrain Blockset includes these HEV and electric vehicle (EV) reference applications:

- Explore the Electric Vehicle Reference Application
- Explore the Hybrid Electric Vehicle P2 Reference Application
- Explore the Hybrid Electric Vehicle Multimode Reference Application
- Explore the Hybrid Electric Vehicle Input Power-Split Reference Application

## HEV Reference Applications: Equivalent consumption minimization strategy for hybrid control

The hybrid electric vehicle (HEV) P0, P1, P2, P3, and P4 reference applications include an equivalent consumption minimization strategy (ECMS) for hybrid control. Previously, the HEV P2 reference application included only rule-based hybrid control. By default, the HEV P2 reference application uses the ECMS variant.

ECMS optimizes the torque split between the engine and motor to minimize energy consumption while maintaining the battery state of charge (SOC). ECMS is an instantaneous minimization method that Powertrain Blockset solves at every controller time step. It can provide near optimal control for a known drive cycle.

For more information, see:

- Explore the Hybrid Electric Vehicle P0 Reference Application
- Explore the Hybrid Electric Vehicle P1 Reference Application
- Explore the Hybrid Electric Vehicle P2 Reference Application
- Explore the Hybrid Electric Vehicle P3 Reference Application
- Explore the Hybrid Electric Vehicle P4 Reference Application

## SI Controller Block: Use closed-loop air-fuel-ratio control to assess three-way catalyst performance, fuel economy, and emissions

The SI Controller block includes closed-loop and open-loop air-fuel-ratio (AFR) control, allowing you to assess the powertrain:

- Three-way-catalyst (TWC) conversion efficiency, including the impact of AFR control waveform characteristics
- Emission control algorithms
- Fuel economy

Previously, the block implemented only open-loop control.

Goal	Option	Controls > Fuel > Closed-loop feedback Parameter Setting
<ul style="list-style-type: none"> <li>• Assess the dynamic and steady-state accuracy of the controller airflow estimation and fuel delivery.</li> </ul>	(default) Open-loop control	off
<ul style="list-style-type: none"> <li>• <i>New in R2019b</i> Hold the average AFR close to stoichiometric AFR to maintain a high TWC conversion efficiency.</li> </ul>	Closed-loop control	on

## Compatibility Considerations

Models that use the SI Controller block might have disconnected line errors. Consider replacing the blocks with the R2019a version.

## Engine Controller Blocks: Prevent over-revving engine with rev-limiter control

To prevent over-revving the engine, the CI Controller and SI Controller blocks implement an engine speed limit controller that limits the engine speed to the value specified by the **Rev-limiter speed threshold** parameter on the **Controls > Idle Speed** tab.

If the engine speed,  $N$ , exceeds the engine speed limit,  $N_{lim}$ , the blocks set the commanded engine torque to 0.

# R2019a

---

**Version: 1.5**

**New Features**

**Bug Fixes**

**Compatibility Considerations**

## Powertrain Efficiency: Evaluate and report energy and power losses at component and system level

R2019a introduces power and energy calculations for component-level blocks and system-level reference applications. You can use the calculations to assess the powertrain efficiency.

Update	Description
Power calculations for blocks	<p>Plant model blocks include calculations for transferred, stored, and not transferred power. The blocks use the Power Accounting Bus Creator block to create the PwrInfo bus. The blocks log the PwrInfo signals to the Info output port.</p> <p>The block documentation includes a <b>Power Accounting</b> section with details about the power calculations.</p>
New block — Power Accounting Bus Creator	Block that logs subsystem transferred, stored, and not transferred power signals to the PwrInfo bus. Use the signals to generate a report that summarizes the energy losses at the component and system levels.
<p>Live scripts for these reference applications that generate energy reports.</p> <ul style="list-style-type: none"> <li>• Explore the Conventional Vehicle Reference Application</li> <li>• Explore the Hybrid Electric Vehicle Multimode Reference Application</li> <li>• Explore the Hybrid Electric Vehicle Input Power-Split Reference Application</li> <li>• Explore the Hybrid Electric Vehicle P2 Reference Application</li> <li>• Explore the Electric Vehicle Reference Application</li> </ul>	<p>The live scripts included with the reference applications analyze the energy consumption at the component- and subsystem-level. After you open one of the listed reference applications, click <b>Analyze Power and Energy</b> to open the live script. To generate the energy report, select <b>Run</b>.</p> <p>The script provides:</p> <ul style="list-style-type: none"> <li>• An overall energy summary that you can export to an Excel spreadsheet.</li> <li>• Engine, electric, and drivetrain efficiencies, including an engine histogram of time spent at the different engine efficiencies.</li> <li>• Data logging so that you can use the Simulation Data Inspector to analyze the powertrain efficiency and energy transfer signals.</li> </ul> <p>For more information, see Analyze Power and Energy.</p>

## Compatibility Considerations

Starting in R2019a, these blocks include the Info output port bus signal. Previously, the blocks did not include the signal. Models that include these blocks that are saved in previous releases might have disconnected line errors. Reconnect the signals or consider replacing the blocks with this version.

- Rotational Inertia
- Split Torsional Compliance
- Torsional Compliance

## Virtual Calibration: Use Model-Based Calibration Toolbox to calibrate Mapped Motor and Three-Phase Voltage Source Inverter block efficiency maps with measured data

If you have the Model-Based Calibration Toolbox, you can use measured data to calibrate efficiency maps for these blocks.

Block	Description																		
Mapped Motor	<p>Calibrate motor efficiency as a function of motor torque, motor speed, and operating temperature.</p> <p>To specify the electrical losses, on the <b>Electrical Losses</b> tab, for <b>Parameterize losses by</b>, select one of these options.</p>																		
	<table border="1"> <thead> <tr> <th>Setting</th> <th>Block Implementation</th> <th>R2019a Update</th> </tr> </thead> <tbody> <tr> <td>Single efficiency measurement</td> <td>Sum of fixed, torque-dependent, and speed-dependent electrical losses.</td> <td><i>None</i></td> </tr> <tr> <td>Tabulated loss data</td> <td> <p>Power loss lookup table that is a function of motor speeds and load torques.</p> <p>If you have Model-Based Calibration Toolbox, click <b>Calibrate Maps</b> to virtually calibrate the 2D lookup tables using measured data.</p> </td> <td><i>Virtual calibration</i></td> </tr> <tr> <td>Tabulated loss data with temperature</td> <td> <p>Loss lookup table that is a function of motor speeds, load torques, and operating temperature.</p> <p>If you have Model-Based Calibration Toolbox, click <b>Calibrate Maps</b> to virtually calibrate the 3D lookup tables using measured data.</p> </td> <td><i>New setting</i></td> </tr> <tr> <td>Tabulated efficiency data</td> <td>2D efficiency lookup table that is a function of motor speeds and load torques.</td> <td><i>None</i></td> </tr> <tr> <td>Tabulated efficiency data with temperature</td> <td>3D efficiency lookup table that is a function of motor speeds, load torques, and operating temperature.</td> <td><i>New setting</i></td> </tr> </tbody> </table>	Setting	Block Implementation	R2019a Update	Single efficiency measurement	Sum of fixed, torque-dependent, and speed-dependent electrical losses.	<i>None</i>	Tabulated loss data	<p>Power loss lookup table that is a function of motor speeds and load torques.</p> <p>If you have Model-Based Calibration Toolbox, click <b>Calibrate Maps</b> to virtually calibrate the 2D lookup tables using measured data.</p>	<i>Virtual calibration</i>	Tabulated loss data with temperature	<p>Loss lookup table that is a function of motor speeds, load torques, and operating temperature.</p> <p>If you have Model-Based Calibration Toolbox, click <b>Calibrate Maps</b> to virtually calibrate the 3D lookup tables using measured data.</p>	<i>New setting</i>	Tabulated efficiency data	2D efficiency lookup table that is a function of motor speeds and load torques.	<i>None</i>	Tabulated efficiency data with temperature	3D efficiency lookup table that is a function of motor speeds, load torques, and operating temperature.	<i>New setting</i>
	Setting	Block Implementation	R2019a Update																
	Single efficiency measurement	Sum of fixed, torque-dependent, and speed-dependent electrical losses.	<i>None</i>																
	Tabulated loss data	<p>Power loss lookup table that is a function of motor speeds and load torques.</p> <p>If you have Model-Based Calibration Toolbox, click <b>Calibrate Maps</b> to virtually calibrate the 2D lookup tables using measured data.</p>	<i>Virtual calibration</i>																
	Tabulated loss data with temperature	<p>Loss lookup table that is a function of motor speeds, load torques, and operating temperature.</p> <p>If you have Model-Based Calibration Toolbox, click <b>Calibrate Maps</b> to virtually calibrate the 3D lookup tables using measured data.</p>	<i>New setting</i>																
	Tabulated efficiency data	2D efficiency lookup table that is a function of motor speeds and load torques.	<i>None</i>																
Tabulated efficiency data with temperature	3D efficiency lookup table that is a function of motor speeds, load torques, and operating temperature.	<i>New setting</i>																	
Three-Phase Voltage Source Inverter	<p>Click <b>Calibrate Maps</b> to virtually calibrate inverter power loss lookup tables as a function of motor torque and motor speed. If you select <b>Input engine temperature</b>, the tables are also a function of the inverter temperature.</p>																		

## New Inverter Block: Model a three-phase voltage source inverter

Use the Three-Phase Voltage Source Inverter block in a closed-loop model to simulate a power inverter. The block implements a three-phase voltage source inverter that generates neutral voltage commands for a balanced three-phase load. You can configure the voltage switching function for continuous vector modulation or inverter switch input signals.

To enable power loss calculations suitable for code generation targets that limit memory, select **Enable memory optimized 2D LUT**.

If you have the Model-Based Calibration Toolbox, click **Calibrate Maps** to virtually calibrate inverter power loss lookup tables as a function of motor torque and motor speed. If you select **Input engine temperature**, the tables are also a function of the inverter temperature.

## Motor Blocks: Configure blocks for discrete sample time to generate code for single-precision targets

To configure these motor blocks for a discrete sample time during simulation, set **Simulation Type** to **Discrete**. Then specify the **Sample Time, Ts** parameter.

- Flux-Based PMSM
- Induction Motor
- Interior PMSM
- Surface Mount PMSM

Consider using the discrete sample time if you want to generate code for fixed-step double- and single-precision targets. Previously, the blocks supported only continuous sample time.

## Mapped Engine Blocks: Account for engine temperature effects on fuel economy, performance, and emissions

To account for engine temperature effects on fuel economy, performance, and emissions, you can implement core engine block tables that are functions of temperature.

This table summarizes the block implementation.

Block	Implementation
Mapped Core Engine	Lookup tables that are functions of engine load, $L$ , and engine speed $N$ . If you select <b>Input engine temperature</b> , the tables are also a function of engine temperature, $T$ .

Block	Implementation													
Mapped CI Engine	Lookup tables that are functions of injected fuel mass, $F$ , engine torque, $T$ , engine speed, $N$ , and engine temperature, $Temp_{Eng}$ .													
	<table border="1"> <thead> <tr> <th>Input Command Setting</th> <th>Input Engine Temperature Parameter Setting</th> <th>Lookup Tables</th> </tr> </thead> <tbody> <tr> <td rowspan="2">Fuel mass</td> <td>off</td> <td><math>f(F,N)</math></td> </tr> <tr> <td>on</td> <td><math>f(F,N,Temp_{Eng})</math></td> </tr> <tr> <td rowspan="2">Torque</td> <td>off</td> <td><math>f(T,N)</math></td> </tr> <tr> <td>on</td> <td><math>f(T,N,Temp_{Eng})</math></td> </tr> </tbody> </table>	Input Command Setting	Input Engine Temperature Parameter Setting	Lookup Tables	Fuel mass	off	$f(F,N)$	on	$f(F,N,Temp_{Eng})$	Torque	off	$f(T,N)$	on	$f(T,N,Temp_{Eng})$
	Input Command Setting	Input Engine Temperature Parameter Setting	Lookup Tables											
	Fuel mass	off	$f(F,N)$											
		on	$f(F,N,Temp_{Eng})$											
Torque	off	$f(T,N)$												
	on	$f(T,N,Temp_{Eng})$												
Mapped SI Engine	Lookup tables that are functions of commanded torque, $T_{cmd}$ , brake torque, $T_{brake}$ , and engine speed, $N$ . If you select <b>Input engine temperature</b> , the tables are also a function of engine temperature, $Temp_{Eng}$ .													
	<table border="1"> <thead> <tr> <th>Input Engine Temperature Parameter Setting</th> <th>Lookup Tables</th> </tr> </thead> <tbody> <tr> <td rowspan="2">off</td> <td><math>f(T_{cmd},N)</math></td> </tr> <tr> <td><math>f(T_{brake},N)</math></td> </tr> <tr> <td rowspan="2">on</td> <td><math>f(T_{cmd},N,Temp_{Eng})</math></td> </tr> <tr> <td><math>f(T_{brake},N,Temp_{Eng})</math></td> </tr> </tbody> </table>	Input Engine Temperature Parameter Setting	Lookup Tables	off	$f(T_{cmd},N)$	$f(T_{brake},N)$	on	$f(T_{cmd},N,Temp_{Eng})$	$f(T_{brake},N,Temp_{Eng})$					
	Input Engine Temperature Parameter Setting	Lookup Tables												
	off	$f(T_{cmd},N)$												
$f(T_{brake},N)$														
on	$f(T_{cmd},N,Temp_{Eng})$													
	$f(T_{brake},N,Temp_{Eng})$													

## Open-Loop Driver: Configure reference application driver commands with constant or signal-based inputs

To implement an open-loop system for driver control in the reference applications, use the Open Loop control system variant. In the subsystem, you can configure constant or signal-based acceleration, deceleration, gear, and clutch commands with constant or signal-based inputs. For more information, see:

- Explore the Conventional Vehicle Reference Application
- Explore the Electric Vehicle Reference Application
- Explore the Hybrid Electric Vehicle Multimode Reference Application
- Explore the Hybrid Electric Vehicle Input Power-Split Reference Application
- Explore the Hybrid Electric Vehicle P2 Reference Application

## Longitudinal Wheel Block: Configure block 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 Longitudinal Wheel block 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 1DOF Longitudinal and Vehicle Body 3DOF Longitudinal blocks include options for configuring the blocks 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	<b>External forces</b>	External force applied to vehicle CG in vehicle-fixed frame.
	MExt	<b>External moments</b>	External moment about vehicle CG in vehicle-fixed frame.
	AirTemp	<b>Air temperature</b>	Ambient air temperature. Consider this option if you want to vary the temperature during run-time.
	WindXYZ	<b>Wind X,Y,Z</b>	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	<b>External forces</b>	External force applied to vehicle CG in vehicle-fixed frame.
	MExt	<b>External moments</b>	External moment about vehicle CG in vehicle-fixed frame.
	AirTemp	<b>Air temperature</b>	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 power and energy losses for these blocks, you can use the **Efficiency factors** parameter.

- Open Differential
- Gearbox
- Limited Slip Differential

This table summarizes the block implementation for each setting.

Setting	Implementation
Constant	Constant efficiency that you can set with the <b>Constant efficiency factor, eta</b> parameter.
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: <ul style="list-style-type: none"><li>• <b>Efficiency lookup table, eta_tbl</b></li><li>• <b>Efficiency torque breakpoints, Trq_bpts</b></li><li>• <b>Efficiency speed breakpoints, omega_bpts</b></li><li>• <b>Efficiency temperature breakpoints, Temp_bpts</b></li></ul> For the air temperature, you can: <ul style="list-style-type: none"><li>• Select <b>Input temperature</b> to create an input port.</li><li>• Set a <b>Ambient temperature, Tamb</b> parameter value.</li></ul>

### Compatibility Considerations

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



# R2018b

---

**Version: 1.4**

**New Features**

**Bug Fixes**

**Compatibility Considerations**

## Virtual Calibration: Use Model-Based Calibration Toolbox to calibrate spark-ignition and compression-ignition engine model blocks

If you have the Model-Based Calibration Toolbox, you can use measured engine data to calibrate parameters for these engine blocks.

Block	Description
Mapped CI Engine	Calibrate power, air mass flow, fuel flow, exhaust temperature, efficiency, and emission performance.
Mapped SI Engine	
Compressor	Calibrate mass flow rate and efficiency.
Turbine	

## HEV P2 Reference Application: Use fully assembled model for HIL testing, tradeoff analysis, and control parameter optimization of a P2 hybrid electric vehicle

Powertrain Blockset includes a hybrid electric vehicle (HEV) P2 reference application that you can use for design tradeoff analysis and component sizing, control parameter optimization, and hardware-in-the-loop (HIL) testing. For more information, see [Explore the Hybrid Electric Vehicle P2 Reference Application](#).

In addition to the HEV P2 reference application, the Powertrain Blockset includes these HEV and electric vehicle (EV) reference applications:

- [Explore the Hybrid Electric Vehicle Multimode Reference Application](#)
- [Explore the Hybrid Electric Vehicle Input Power-Split Reference Application](#)
- [Explore the Electric Vehicle Reference Application](#)

## Compression-Ignition Engine: Model engine torque and exhaust temperature as a function of multiple fuel injections, fuel pressure, and air system states

To determine the engine torque, the compression-ignition (CI) torque structure model uses an industry-standard approach that aligns with calibration processes. To improve the engine torque and exhaust temperature fit with data, the approach decouples these engine effects:

- Start of injection (SOI) timing
- Exhaust back-pressure
- Burned fuel mass
- Late injection fuel mass
- Pass-through fuel mass
- Intake manifold gas pressure, temperature, and oxygen percentage
- Fuel rail pressure

For more information about using the torque structure model, use these links.

About	See
Decoupling the CI engine effects on engine torque and engine temperature	CI Engine Torque Structure Model
Blocks that implement the model	CI Controller CI Core Engine
Reference applications that implement the model	Explore the Conventional Vehicle Reference Application Explore the CI Engine Dynamometer Reference Application Explore the Hybrid Electric Vehicle Multimode Reference Application Explore the Hybrid Electric Vehicle P2 Reference Application

### Compatibility Considerations

In previous releases, the CI engine model torque structure model did not use the fuel rail pressure to calculate the engine torque. The CI Core Engine block now includes an inport port, FuelPrs, for the calculation.

Models that use the CI Core Engine block might have disconnected line errors. Consider replacing the block with the R2018b version.

### Spark-Ignition Engine Startup: Use SI Controller block to account for fuel consumption due to rich operation during engine startup

The SI Controller block accounts for the extra fuel delivered to the spark-ignition (SI) engine during startup. If the engine speed is greater than the startup engine cranking speed, the SI Controller block enriches the optimal relative air-fuel ratio ( $\lambda$ ) with an exponentially decaying delta  $\lambda$ . To initialize the delta  $\lambda$ , the block uses the engine coolant temperature at startup. The delta  $\lambda$  exponentially decays to zero based on a time constant that is a function of the engine coolant temperature.

### Longitudinal Driver Block: Implement predictive driver control and specify gear schedule to track longitudinal speed, including reverse

You can configure the Longitudinal Driver block to implement predictive driver control and specify a gear schedule to track longitudinal speed, including reverse. The blocks include these control type options.

Control Type Options	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.

Control Type Options	Block Implementation
Predictive	<p>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:</p> <ul style="list-style-type: none"> <li>• Represents the dynamics as a linear single track (bicycle) vehicle</li> <li>• Minimizes the previewed error signal at a single point <math>T^*</math> seconds ahead in time</li> <li>• Accounts for the driver lag deriving from perceptual and neuromuscular mechanisms</li> </ul>

The blocks include these shift type options.

Shift Type Options	Block Implementation
None	<p>No transmission. Block outputs a constant gear of 1.</p> <p>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.</p>
Reverse, Neutral, Drive	<p>Block uses a Stateflowchart to model reverse, neutral, and drive gear shift scheduling.</p> <p>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.</p> <p>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.</p>
Scheduled	<p>Block uses a Stateflowchart to model reverse, neutral, park, and N-speed gear shift scheduling.</p> <p>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:</p> <ul style="list-style-type: none"> <li>• Initial gear</li> <li>• Upshift and downshift accelerator pedal positions</li> <li>• Upshift and downshift velocity</li> <li>• Timing for shifting and engaging forward and reverse from neutral</li> </ul> <p>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.</p>

Shift Type Options	Block Implementation
External	<p>Block uses the input gear, vehicle state, and velocity feedback to generate acceleration and braking commands to track forward and reverse vehicle motion.</p> <p>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.</p>

## CI and SI Engine Dynamometer Reference Applications: Resize engines and recalibrate controllers based on desired engine displacement and number of cylinders

Based on a desired maximum engine power or engine displacement, you can resize the compression-ignition (CI) engine and spark-ignition (SI) engine models and recalibrate the controllers. Previously, you could only resize the engine based on maximum engine power.

To resize the engines, use the dynamometer reference applications. After you open the reference application, click **Resize Engine and Recalibrate Controller**. In the dialog box, set **Power or displacement** to either:

- Power — Enter a **Desired maximum power**
- Displacement — Enter a **Desired displacement**

For either power or displacement, set the **Desired number of cylinders**.

After you apply the changes, the reference application:

- Resizes the dynamic engine and engine calibration parameters. The **Recalibrate Engine** dialog box provides the updated engine performance characteristics based on the resized calibration parameters.
- Recalibrates the controller and mapped engine model to match the resized dynamic engine.

You can use the variants in other applications, for example, in vehicle projects that require a larger engine model.

For resizing examples, see:

- Resize the CI Engine
- Resize the SI Engine

## Electric Vehicle Reference Application: Implements dynamic supervisory control to determine torque and regenerative braking commands

To determine the motor torque and brake pressure commands, the EV reference application implements a supervisory controller. Specifically, the controller subsystem includes a powertrain control module (PCM) with:

- Regenerative braking control

- Motor torque arbitration and power management

For more information, see [Explore the Electric Vehicle Reference Application](#).



# R2018a

---

**Version: 1.3**

**New Features**

**Bug Fixes**

**Compatibility Considerations**

## **Engine Data Import: Use engine test data to calibrate mapped engines in CI and SI engine dynamometer reference applications**

If you have the Model-Based Calibration Toolbox, you can use the engine dynamometer reference applications to generate lookup tables for the Mapped CI Engine and Mapped SI Engine blocks. The reference applications use engine data to calibrate the engines and generate the tables. For more information, see:

- Generate Mapped CI Engine from a Spreadsheet
- Generate Mapped SI Engine from a Spreadsheet

## **Low-Pressure EGR: Default SI engine includes low-pressure exhaust gas recirculation**

You can model low-pressure exhaust gas recirculation (EGR) in the spark-ignition (SI) engine. The Powertrain Blockset uses the SI engine model in:

- Blocks
  - SI Core Engine block
  - SI Controller block
- Templates and reference applications
  - SI Engine Project Template
  - Explore the Conventional Vehicle Reference Application
  - Explore the Hybrid Electric Vehicle Multimode Reference Application
  - Explore the Hybrid Electric Vehicle Input Power-Split Reference Application
  - Explore the SI Engine Dynamometer Reference Application

### **Compatibility Considerations**

In previous releases, the SI engine model did not calculate EGR. To calculate the EGR, the SI Controller block now includes these input ports:

- `Iat` — Intake air temperature
- `EgrVlvInTemp` — EGR valve inlet temperature
- `EgrVlvAreaPct` — EGR valve area percent
- `EgrVlvDeltaPrs` — EGR valve delta pressure

Models that use the SI Controller block might have disconnected line errors. Consider replacing the block with the R2018a version.

## **Gear Shift Schedule: Use Drive Cycle block to specify gear shift schedule**

Use the Drive Cycle Source block to set drive cycle gear shift schedules. To specify the gear shift schedules, you can use:

- A support package to install standard drive cycles that include the gear shift schedules, for example JC08 and CUEDC. By default, the block includes the FTP–75 drive cycle. To install additional drive cycles, see Install Drive Cycle Data.
- Workspace variables.
- MAT, XLS, XLSX, or TXT files.

## Transmission Temperature Dependency: Include effects of oil temperature when calculating efficiency loss in transmission blocks

You can configure these transmission blocks to calculate efficiency losses:

- Automated Manual Transmission
- Dual Clutch Transmission
- Ideal Fixed Gear Transmission

To specify the block efficiency calculation, for **Efficiency factors**, select either of these options.

Setting	Block Implementation
Gear only	Efficiency determined from a 1-D lookup table that is a function of the gear.
Gear, input torque, input speed, and temperature	Efficiency determined from a 4-D lookup table that is a function of: <ul style="list-style-type: none"> <li>• Gear</li> <li>• Input torque</li> <li>• Input speed</li> <li>• Oil temperature</li> </ul>

## Engine Torque Table: Mapped CI Engine block now uses either engine torque or injected fuel mass lookup table

To use tables that are functions of commanded engine torque and engine speed, on the Mapped CI Engine block parameter dialog box, select the **Input Command** parameter Torque. Previously, you could only use tables that were functions of injected fuel mass and engine speed.



# R2017b

---

**Version: 1.2**

**New Features**

**Bug Fixes**

**Compatibility Considerations**

## **HEV Input Power-Split Reference Application: Use fully assembled model for HIL testing, tradeoff analysis, and control parameter optimization of a power-split hybrid like the Toyota Prius**

Powertrain Blockset includes a hybrid electric vehicle (HEV) input power-split reference application that you can use for design tradeoff analysis and component sizing, control parameter optimization, and hardware-in-the-loop (HIL) testing for vehicles like the Toyota® Prius®. For more information, see [Explore the Hybrid Electric Vehicle Input Power-Split Reference Application](#).

In addition to the HEV input power-split reference application, the Powertrain Blockset includes an HEV multimode reference application. Prior to R2017b, the HEV multimode reference application was known as the HEV reference application. For more information, see [Explore the Hybrid Electric Vehicle Multimode Reference Application](#).

## **CI and SI Engine Dynamometer Reference Applications: Resize engines and recalibrate controllers based on desired power and number of cylinders**

Based on a desired maximum engine power and the number of cylinders, you can resize the compression-ignition (CI) engine and spark-ignition (SI) engine models and recalibrate the controllers.

To resize the engines, use the dynamometer reference applications. After you open the reference application, click **Resize Engine and Recalibrate Controller**. In the dialog box, enter values for:

- **Desired maximum power**
- **Desired number of cylinders**

After you apply the changes, the reference application:

- Resizes the dynamic engine and engine calibration parameters. The **Recalibrate Engine** dialog box provides the updated engine performance characteristics based on the resized calibration parameters.
- Recalibrates the controller and mapped engine model to match the resized dynamic engine.

You can use the variants in other applications, for example, in vehicle projects that require a larger engine model.

For resizing examples, see:

- [Resize the CI Engine](#)
- [Resize the SI Engine](#)

## **New Drivetrain and Propulsion Blocks: Model drivetrain coupling, bidirectional DC-to-DC energy conversion, and flux-based PMSM motors**

This table summarizes the blocks that are available with R2017b.

Block	Description
Split Torsional Compliance	Implements parallel spring-damper coupling between shafts. Use the block to model the mechanical power transfer between common drivetrain elements such as motors, planetary gears, and clutches.
Bidirectional DC-DC	Implements a DC-to-DC converter that supports bidirectional boost and buck (lower) operation. Depending on your battery system configuration, the voltage might not be at a potential that is required by electrical system components such as inverters and motors. You can use the block to boost or buck the voltage.
Flux-Based PMSM	Implements a flux-based, three-phase permanent magnet synchronous motor (PMSM) with a tabular-based electromotive force.
Flux-Based PM Controller	Implements a torque-based, field-oriented controller for a flux-based PMSM.

## Electric Motor Controllers: Calculate inverter electrical losses

You can configure these electric motor controller blocks to calculate electrical losses:

- Bidirectional DC-DC
- Flux-Based PM Controller
- IM Controller
- Interior PM Controller
- Surface Mount PM Controller

Previously, you could configure only the Mapped Motor block to calculate electrical loss.

To specify the electrical loss calculation, on the block **Electrical Losses** tab, for **Parameterize losses by**, select one of these options.

Setting	Block Implementation
Single efficiency measurement	Electrical loss calculated using a constant value for inverter efficiency
Tabulated loss data	Electrical loss calculated as a function of motor speeds and load torques
Tabulated efficiency data	Electrical loss calculated using inverter efficiency that is a function of motor speeds and load torques

## Flux-Based Motor Parameterization: Generate parameters for Flux-Based PMSM and Flux-Based PM Controller blocks

This version includes workflows that you can follow to generate parameters for the flux-based motor blocks. See Generate Parameters for Flux-Based Blocks.

## Longitudinal Wheel Block: Model tires using Magic Formula longitudinal slip, vertical motion, and external friction input

The Longitudinal Wheel block includes these enhancements.

Enhancement	Implementation
Longitudinal force as a function of wheel slip relative to the road surface. Implemented using coefficients fit from experimental data or derived using Magic Formula equations 4.E9 through 4.E18 in <i>Tire and Vehicle Dynamics</i> .	Set <b>Longitudinal Force</b> to Magic Formula pure longitudinal slip.
Vertical motion that depends on wheel mass stiffness, damping, and pressure.	Set <b>Vertical Motion</b> to Mapped stiffness and damping.  To specify the ground displacement, on the <b>Vertical</b> pane, do either of the following: <ul style="list-style-type: none"> <li>• Select <b>Input ground displacement</b> to create input port Gnd.</li> <li>• Specify a <b>Ground displacement, Gndz</b> parameter.</li> </ul>
Input tire pressure for Magic Formula, vertical motion, and rolling resistance calculations.	To create the TirePrs port: <ul style="list-style-type: none"> <li>• Set one of these parameters: <ul style="list-style-type: none"> <li>• <b>Longitudinal Force</b> to Magic Formula pure longitudinal slip.</li> <li>• <b>Rolling Resistance</b> to Pressure and velocity or Magic Formula.</li> <li>• <b>Vertical Motion</b> to Mapped stiffness and damping.</li> </ul> </li> <li>• On the <b>Wheel Dynamics</b> pane, select <b>Input tire pressure</b>.</li> </ul>
Input scaling factor for longitudinal friction calculation.	To create the lam_mux port, select <b>Input friction scale factor</b> .

## References

- [1] Pacejka, H. B. *Tire and Vehicle Dynamics*. 3rd ed. Oxford, United Kingdom: SAE and Butterworth-Heinemann, 2012.

## CI Core Engine Block: Customize fuel injection with fuel mass input

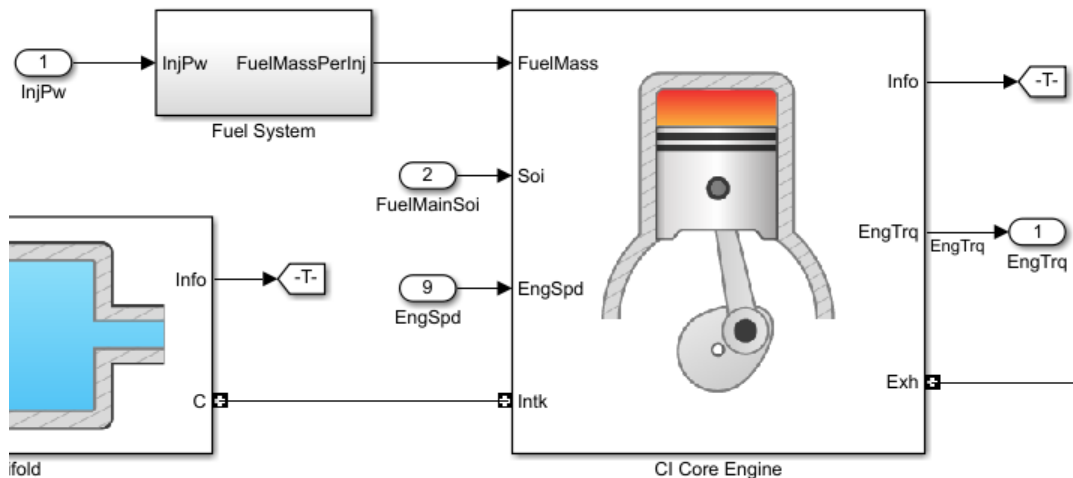
To accommodate multiple fuel injection events during hardware-in-the-loop (HIL) simulation, you can provide the CI Core Engine block with these fuel-related input vectors:

- FuelMass — Fuel mass per injection
- Soi — Fuel injection timing



The `CiEngineCore.slx` model includes the Fuel System subsystem and the updated CI Core Engine block. The Fuel System subsystem contains a fuel delivery command subsystem. These reference applications and templates use `CiEngineCore.slx`:

- Conventional Vehicle Reference Application
- Hybrid Electric Vehicle Reference Application
- CI Engine Dynamometer Reference Application
- CI Engine Project Template



### Compatibility Considerations

In previous releases, the CI Core Engine block calculated the fuel mass flow rate using fuel injector pulse-width and fuel injection main timing block input. In this release, the `InjPw` and `FuelMainSoi` scalar input ports are replaced by `FuelMass` and `Soi` vector input ports. Models that have `InjPw` or `FuelMainSoi` signals input to the CI Core Engine block might have disconnected line errors. Consider replacing the block with this version.

### Combustion Engine Reference Applications: Use control actuator IO during HIL testing

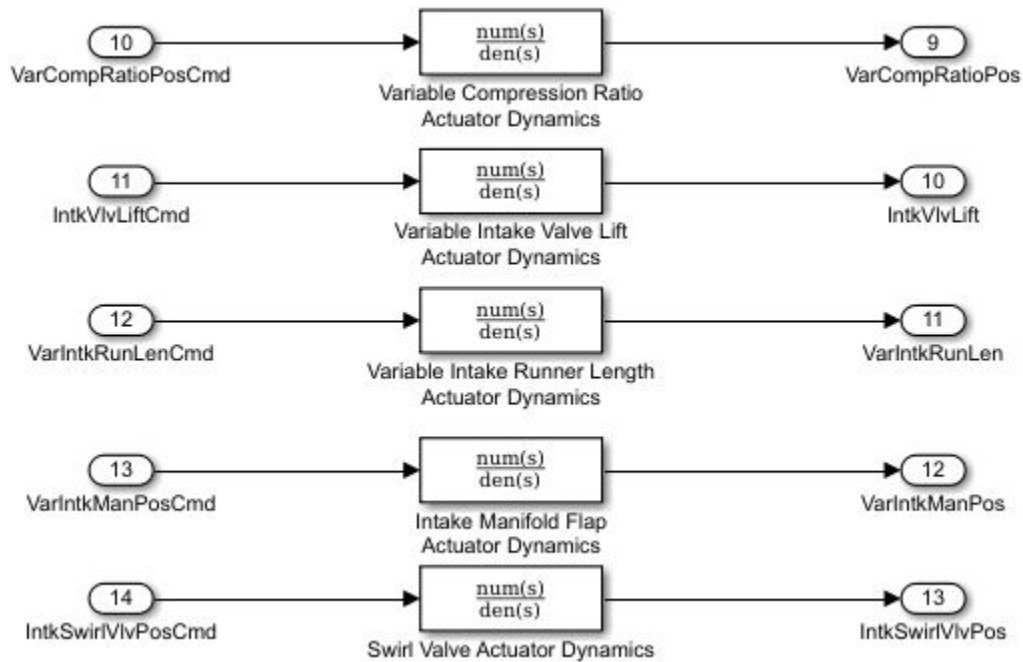
To facilitate hardware-in-the-loop (HIL) testing of actuator and sensor dynamics, you can use control actuator IO for these internal combustion engine reference applications:

- Conventional Vehicle Reference Application
- CI Engine Dynamometer Reference Application
- SI Engine Dynamometer Reference Application

Specifically, the compression-ignition (CI) and spark-ignition (SI) engines available with the reference applications use low-pass filters to model these control actuators:

- Variable compression ratio
- Variable intake valve lift

- Variable intake runner length
- Intake manifold flap
- Swirl valve



## Battery Blocks: Input rated capacity at nominal temperature

To simulate effects such as aging, you can configure these blocks to input rated capacity at nominal temperature:

- Datasheet Battery
- Equivalent Circuit Battery

### Compatibility Considerations

Models saved in previous releases might have disconnected line errors. Reconnect the s or consider replacing the blocks with this version.

## Mapped Engine Blocks: Model turbocharger lag

To simulate turbocharger lag with the mapped engine blocks, on the block parameter dialog box, select **Include turbocharger lag effect**. To model the lag, the blocks use a first-order system with a time constant.

- Mapped CI Engine — At low torque, boost is not required to provide sufficient air flow. When the requested fuel mass requires boost, the block uses a time constant to determine the maximum fuel mass per injection.

- Mapped SI Engine — During throttle control, the time constant models the manifold filling and emptying dynamics. When the torque request requires a turbocharger boost, the block uses a larger time constant to represent the turbocharger lag.

## Conventional and Hybrid Electric Vehicle Reference Applications: Idle the engine until catalyst light-off

To simulate catalyst light-off before moving the vehicle with a pedal command, you can idle the conventional and hybrid electric vehicle engines at the beginning of a drive cycle. In the reference applications, the `Longitudinal Driver` subsystem includes an ignition switch signal profile, `IgSw`. The engine controller uses the ignition switch signal to start both the engine and a catalyst light-off timer.

The catalyst light-off timer overrides the engine stop-start (ESS) stop function control while the catalyst light-off timer is counting up. During the simulation, after the `IgSw` down-edge time reaches the catalyst light-off time `CatLightOffTime`, normal ESS operation resumes. If there is no torque command before the simulation reaches the `EngStopTime`, the ESS shuts down the engine.

To control ESS and catalyst light-off, use these engine controller calibration parameters:

- `EngStopStartEnable` — Enables ESS. To disable ESS, set the value to false.
- `CatLightOffTime` — Engine idle time from engine start to catalyst light-off.
- `EngStopTime` — ESS engine run time after driver model torque request cut-off.
- `IgSw` — Starts and idles the engine. Set ignition switch profile to 'on' inside driver model.

These parameters are in the engine controller model workspace.

## Combustion Engine Torque: Independent friction and pumping loss calculation

The compression-ignition (CI) and spark-ignition (SI) torque structure calculation independently accounts for pumping and friction losses. Previously, the torque calculation combined the losses.

Combustion Engine	Description	Impacted Blocks	Friction Loss	Pumping Loss
CI	CI Engine Torque Structure Model	CI Controller CI Core Engine	Function of: <ul style="list-style-type: none"> <li>• Engine coolant temperature</li> <li>• Injected fuel mass</li> <li>• Engine speed</li> </ul>	Function of: <ul style="list-style-type: none"> <li>• Injected fuel mass</li> <li>• Engine speed</li> </ul>
SI	SI Engine Torque Structure Model	SI Controller SI Core Engine	Function of: <ul style="list-style-type: none"> <li>• Engine coolant temperature</li> <li>• Engine load</li> <li>• Engine speed</li> </ul>	Function of: <ul style="list-style-type: none"> <li>• Engine load</li> <li>• Engine speed</li> </ul>

### Compatibility Considerations

Models saved in previous releases might have disconnected line errors. Reconnect the signals or consider replacing the blocks with this version.

### Ports and Signals: Consistent names and units across library blocks

For consistency and readability, the Powertrain Blockset includes these updates for the library blocks.

Update	Description
Input and Output port names	Consistent Input and Output port names across all blocks.
Output port Info bus signal names	Consistent Info bus signal names across all blocks.
Input and Output port units	SI units for block Input and Output ports. To display the signal units in your model, select <b>Display &gt; Signals &amp; Ports &gt; Port Units</b> .

The block documentation includes the names and units for the Input and Output ports and Info bus signals.

### Compatibility Considerations

Models saved in previous releases might have disconnected line errors. Reconnect the signals or consider replacing the blocks with this version.

# R2017a+

---

**Version: 1.1.1**

**Bug Fixes**



# R2017a

---

**Version: 1.1**

**New Features**

**Bug Fixes**

**Compatibility Considerations**

## Wide Open Throttle (WOT) Drive Cycle: Generate WOT drive cycle signals for performance testing

Use the Drive Cycle Source block to generate WOT drive cycle signals for performance testing. You can specify a drive cycle start time, nominal reference speed, and the time to start vehicle deceleration.

## Engine-Out (EO) Emissions: Track EO emissions using combustion engine blocks

### Core Engine Blocks

Use the combustion engine blocks to track these EO exhaust emission components.

Block	EO Exhaust Emission				
	HC - Hydrocarbon	CO - Carbon monoxide	NOx - Nitric oxide and nitrogen dioxide	CO <sub>2</sub> - Carbon dioxide	PM - Particulate matter
CI Core Engine	✓ <i>New in R2017a</i>	✓ <i>New in R2017a</i>	✓ <i>New in R2017a</i>	✓ <i>New in R2017a</i>	✓ <i>New in R2017a</i>
SI Core Engine	✓ <i>New in R2017a</i>	✓ <i>New in R2017a</i>	✓ <i>New in R2017a</i>	✓ <i>New in R2017a</i>	✓ <i>New in R2017a</i>
Mapped Core Engine	✓	✓	✓	✓	✓ <i>New in R2017a</i>
Mapped CI Engine	✓	✓	✓	✓	✓ <i>New in R2017a</i>
Mapped SI Engine	✓	✓	✓	✓	✓ <i>New in R2017a</i>

### Two-way Connector Ports

The two-way connector ports for these component blocks include the EO exhaust emission components.

- Compressor
- CI Core Engine
- Control Volume System
- Flow Boundary
- Flow Restriction
- Heat Exchanger
- SI Core Engine



- 
- Turbine

## **Hybrid Electric Vehicle Reference Application: Minimize drive cycle fuel consumption using a dynamic embedded optimizer**

The hybrid vehicle reference application implements a dynamic embedded controller that minimizes fuel consumption. The controller does not require pre-calculated optimal control signals. Instead, the controller calculates the control signals that minimize brake-specific fuel consumption (BSFC) and meet the engine power requirements. For more information, see Explore the Hybrid Electric Vehicle Reference Application.

## **Drivetrain Blocks: Model drivetrain coupling with Disc Clutch, Gearbox, and Planetary Gear blocks**

Use these blocks to model the mechanical power transfer between common driveline elements such as transmissions, engines, clutches, and differentials. You can use the blocks to customize the drivetrains that are available in the reference applications.

- Disc Clutch - Implements an ideal dry friction clutch.
- Gearbox - Implements an ideal rotational gearbox.
- Planetary Gear - Implements an ideal planetary gear consisting of a rigidly connected sun, ring, and carrier.

## **Battery Parameterization: Generate parameters for Datasheet Battery and Equivalent Circuit Battery blocks**

This version includes two workflows that you can follow to generate parameters for battery blocks:

- Generate Parameter Data for Datasheet Battery Block
- Generate Parameter Data for Equivalent Circuit Battery Block

## **CI Core Engine Block: Track exhaust gas recirculation via two-way connector port**

Track the exhaust gas recirculation (EGR) mass flow rate in the CI Core Engine block with the Intk two-way connector port.

### **Compatibility Considerations**

In the previous release, the EGR mass flow rate was input to the block via the EgrFlow input port. In this release, the EgrFlow input port is removed. Models that have EgrFlow signals input to the CI Core Engine block might have disconnected line errors. Consider replacing the block with this version.



# R2016b+

---

**Version: 1.0**

**New Features**

## Introducing Powertrain Blockset

Powertrain Blockset provides fully assembled reference applications of automotive powertrains, including gasoline, diesel, hybrid, and electric systems. You can use it for design tradeoff analysis and component sizing, control parameter optimization, and hardware-in-the-loop (HIL) testing. You can customize models by parameterizing components in a reference application with your own data or by replacing a subsystem with your own model. See [Getting Started with Powertrain Blockset](#).

## Fully assembled models for gasoline, diesel, hybrid, and electric powertrains

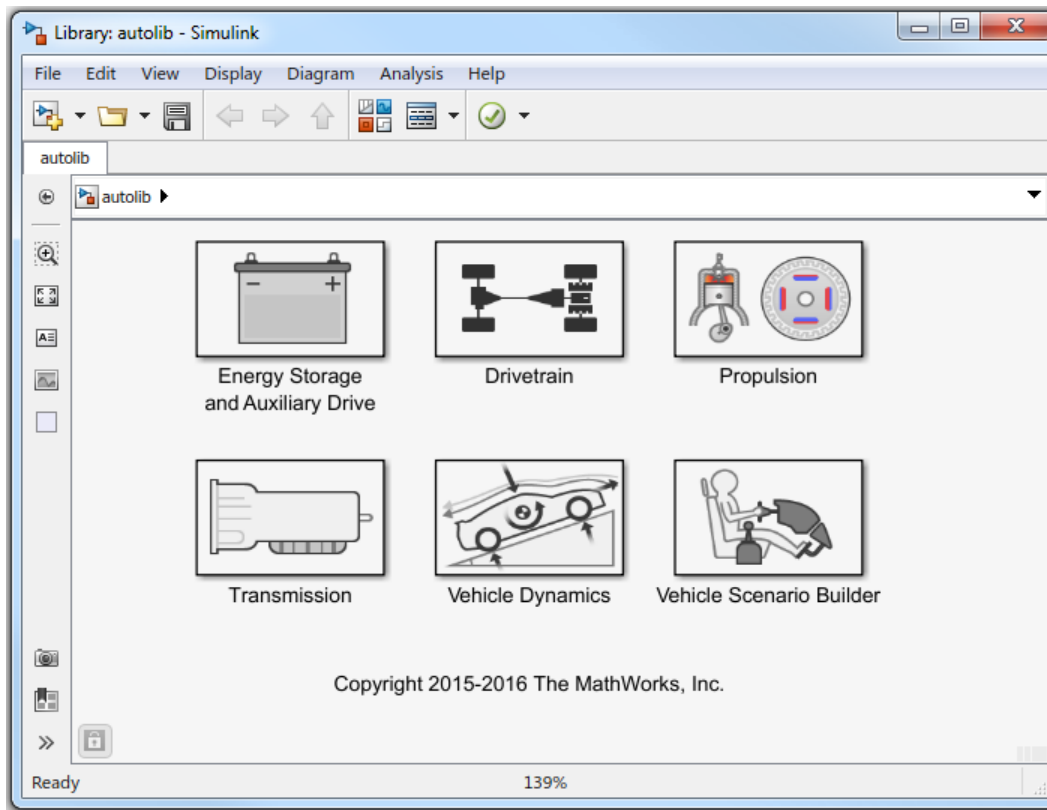
Powertrain Blockset has full vehicle reference application projects that you can use for design tradeoff analysis and component sizing, control parameter optimization, and hardware-in-the-loop (HIL) testing.

- Explore the Conventional Vehicle Reference Application
- Explore the Hybrid Vehicle Reference Application
- Explore the Electric Vehicle Reference Application

## Libraries for engine, transmission, traction motor, and battery components

The Powertrain Blockset library has blocks that you can use to simulate engine subsystems, transmission assemblies, traction motors, battery packs, and controller models.

To open the Powertrain Blockset library, at MATLAB command-line, type `autolib`.



## Basic controllers for powertrain subsystems

The Powertrain Blockset library contains electric motor, combustion engine, and transmission controller blocks that you can use in powertrain subsystem models.

## Standard drive cycle data, including FTP-75, NEDC, and JC08

Use the Drive Cycle Source block to select standard drive cycles in full vehicle model simulations. By default, the block includes the FTP-75 drive cycle. To install additional over 25 additional drive cycles via a support package, including NEDC and JC08, see [Install Drive Cycle Data](#).

## Engine dynamometer models for virtual calibration and testing

To represent an engine plant and controller connected to an AC dynamometer with a tailpipe emission analyzer, Powertrain Blockset provides reference application projects for compression-ignition (CI) and spark-ignition (SI) engines. Use the CI and SI dynamometer reference applications to calibrate, validate, and optimize engine controller and plant model parameters before integrating the engine with the vehicle model.

- Explore the CI Engine Dynamometer Reference Application
- Explore the SI Engine Dynamometer Reference Application

## **MDF file support for calibration data import**

These Powertrain Blockset functions allow you to read data and save attachments from Measurement Data Format (MDF) files.

- `mdf` — Access information contained in MDF files
- `read` — Read channel data from MDF files
- `saveAttachment` — Save attachment from MDF files