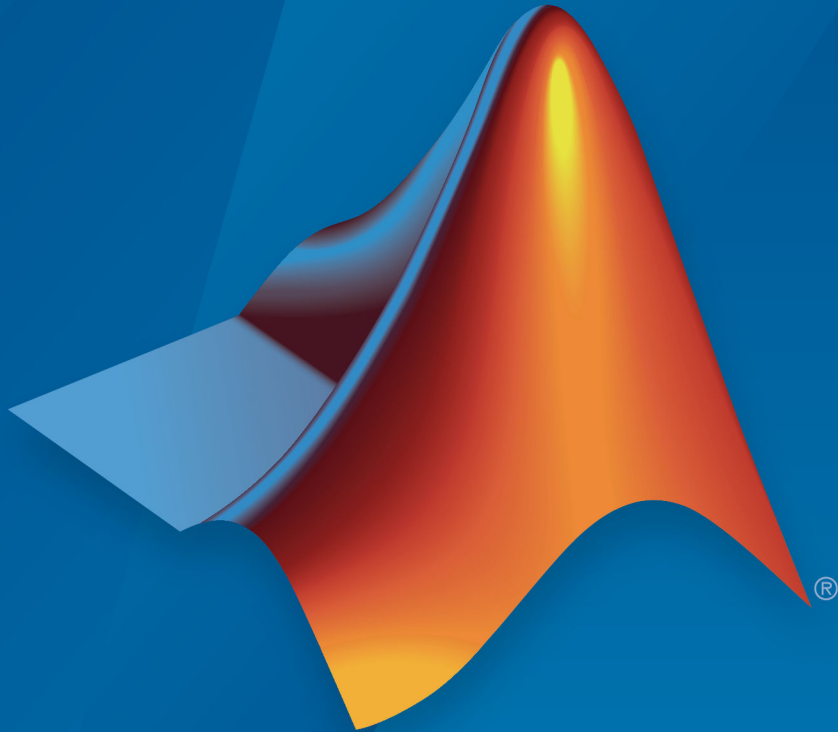


Powertrain Blockset™ Release Notes



MATLAB® & SIMULINK®

How to Contact MathWorks



Latest news: www.mathworks.com
Sales and services: www.mathworks.com/sales_and_services
User community: www.mathworks.com/matlabcentral
Technical support: www.mathworks.com/support/contact_us



Phone: 508-647-7000



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

Powertrain Blockset™ Release Notes

© COPYRIGHT 2016–2018 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 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 for more information.

| | |
|---|------------|
| Virtual Calibration: Use Model-Based Calibration Toolbox to calibrate spark-ignition and compression-ignition engine model blocks | 1-2 |
| HEV P2 Reference Application: Use fully assembled model for HIL testing, tradeoff analysis, and control parameter optimization of a P2 hybrid electric vehicle | 1-2 |
| Compression-Ignition Engine: Model engine torque and exhaust temperature as a function of multiple fuel injections, fuel pressure, and air system states | 1-2 |
| Compatibility Considerations | 1-3 |
| Spark-Ignition Engine Startup: Use SI Controller block to account for fuel consumption due to rich operation during engine startup | 1-4 |
| Longitudinal Driver Block: Implement predictive driver control and specify gear schedule to track longitudinal speed, including reverse | 1-4 |
| CI and SI Engine Dynamometer Reference Applications: Resize engines and recalibrate controllers based on desired engine displacement and number of cylinders | 1-6 |
| Electric Vehicle Reference Application: Implements dynamic supervisory control to determine torque and regenerative braking commands | 1-7 |

| | |
|--|-----|
| Engine Data Import: Use engine test data to calibrate mapped engines in CI and SI engine dynamometer reference applications | 2-2 |
| Low-Pressure EGR: Default SI engine includes low-pressure exhaust gas recirculation | 2-2 |
| Compatibility Considerations | 2-2 |
| Gear Shift Schedule: Use Drive Cycle block to specify gear shift schedule | 2-3 |
| Transmission Temperature Dependency: Include effects of oil temperature when calculating efficiency loss in transmission blocks | 2-3 |
| Engine Torque Table: Mapped CI Engine block now uses either engine torque or injected fuel mass lookup table | 2-4 |

| | |
|--|-----|
| 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 | 3-2 |
| CI and SI Engine Dynamometer Reference Applications: Resize engines and recalibrate controllers based on desired power and number of cylinders | 3-2 |
| New Drivetrain and Propulsion Blocks: Model drivetrain coupling, bidirectional DC-to-DC energy conversion, and flux-based PMSM motors | 3-3 |
| Electric Motor Controllers: Calculate inverter electrical losses | 3-3 |

| | |
|--|-------------|
| Flux-Based Motor Parameterization: Generate parameters for Flux-Based PMSM and Flux-Based PM Controller blocks . . | 3-4 |
| Longitudinal Wheel Block: Model tires using Magic Formula longitudinal slip, vertical motion, and external friction input | 3-4 |
| CI Core Engine Block: Customize fuel injection with fuel mass input | 3-6 |
| Compatibility Considerations | 3-7 |
| Combustion Engine Reference Applications: Use control actuator IO during HIL testing | 3-7 |
| Battery Blocks: Input rated capacity at nominal temperature | 3-8 |
| Compatibility Considerations | 3-8 |
| Mapped Engine Blocks: Model turbocharger lag | 3-9 |
| Conventional and Hybrid Electric Vehicle Reference Applications: Idle the engine until catalyst light-off | 3-9 |
| Combustion Engine Torque: Independent friction and pumping loss calculation | 3-10 |
| Compatibility Considerations | 3-10 |
| Ports and Signals: Consistent names and units across library blocks | 3-10 |
| Compatibility Considerations | 3-11 |

R2017a+

Bug Fixes

| | |
|--|-----|
| Wide Open Throttle (WOT) Drive Cycle: Generate WOT drive cycle signals for performance testing | 5-2 |
| Engine-Out (EO) Emissions: Track EO emissions using combustion engine blocks | 5-2 |
| Core Engine Blocks | 5-2 |
| Two-way Connector Ports | 5-3 |
| Hybrid Electric Vehicle Reference Application: Minimize drive cycle fuel consumption using a dynamic embedded optimizer | 5-3 |
| Drivetrain Blocks: Model drivetrain coupling with Disc Clutch, Gearbox, and Planetary Gear blocks | 5-4 |
| Battery Parameterization: Generate parameters for Datasheet Battery and Equivalent Circuit Battery blocks | 5-4 |
| CI Core Engine Block: Track exhaust gas recirculation via two-way connector port | 5-4 |
| Compatibility Considerations | 5-4 |

| | |
|--|-----|
| Introducing Powertrain Blockset | 6-2 |
| Fully assembled models for gasoline, diesel, hybrid, and electric powertrains | 6-2 |
| Libraries for engine, transmission, traction motor, and battery components | 6-2 |
| Basic controllers for powertrain subsystems | 6-3 |

| | |
|--|------------|
| Standard drive cycle data, including FTP-75, NEDC, and JC08 | 6-3 |
| Engine dynamometer models for virtual calibration and testing | 6-4 |
| MDF file support for calibration data import | 6-4 |

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 import 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 (λ) with an exponentially decaying delta λ . To initialize the delta λ , the block uses the engine coolant temperature at startup. The delta λ 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"> • Represents the dynamics as a linear single track (bicycle) vehicle • Minimizes the previewed error signal at a single point T^* seconds ahead in time • Accounts for the driver lag deriving from perceptual and neuromuscular mechanisms |

The blocks include these shift type options.

| Shift Type Options | Block Implementation |
|---------------------------|--|
| None | <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 Stateflow® chart 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> |

| Shift Type Options | Block Implementation |
|--------------------|--|
| 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"> • Initial gear • Upshift and downshift accelerator pedal positions • Upshift and downshift velocity • Timing for shifting and engaging forward and reverse from neutral <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> |
| 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. |

| Setting | Block Implementation |
|--|--|
| 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">• Gear• Input torque• Input speed• Oil temperature |

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 Longitudinal Force to Magic Formula pure longitudinal slip. |

| Enhancement | Implementation |
|--|--|
| Vertical motion that depends on wheel mass stiffness, damping, and pressure. | Set Vertical Motion to Mapped stiffness and damping. To specify the ground displacement, on the Vertical pane, do either of the following: <ul style="list-style-type: none"> • Select Input ground displacement to create input port Gnd. • Specify a Ground displacement, Gndz parameter. |
| Input tire pressure for Magic Formula, vertical motion, and rolling resistance calculations. | To create the TirePrs port: <ul style="list-style-type: none"> • Set one of these parameters: <ul style="list-style-type: none"> • Longitudinal Force to Magic Formula pure longitudinal slip. • Rolling Resistance to Pressure and velocity or Magic Formula. • Vertical Motion to Mapped stiffness and damping. • On the Wheel Dynamics pane, select Input tire pressure. |
| Input scaling factor for longitudinal friction calculation. | To create the lam_mux port, select Input friction scale factor . |

References

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

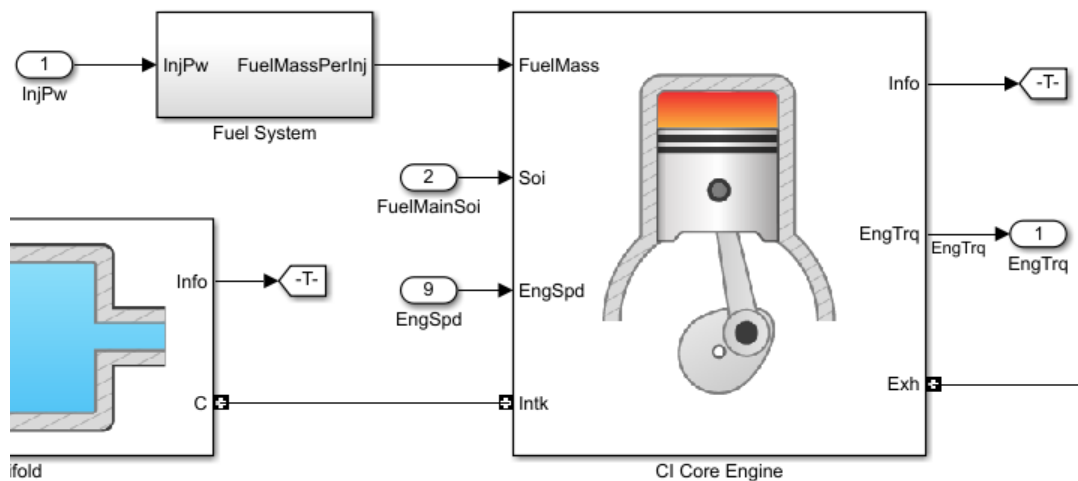
CI Core Engine Block: Customize fuel injection with fuel mass input

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.

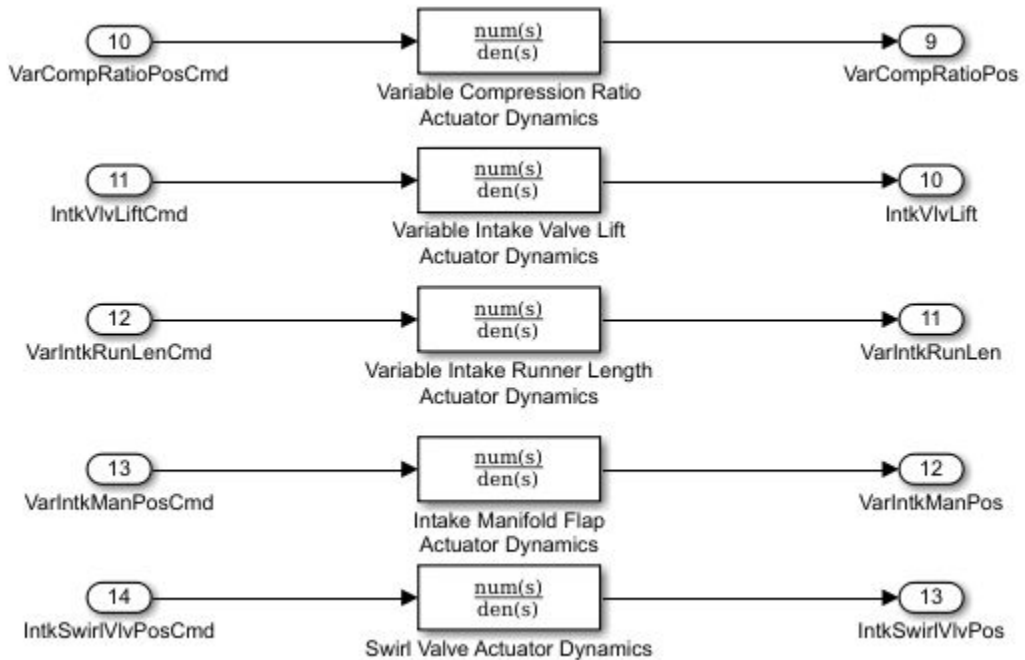
Combustion Engine Reference Applications: Use control actuator IO during HIL testing

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 signals 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"> • Engine coolant temperature • Injected fuel mass • Engine speed | Function of: <ul style="list-style-type: none"> • Injected fuel mass • Engine speed |
| SI | SI Engine Torque Structure Model | SI Controller SI Core Engine | Function of: <ul style="list-style-type: none"> • Engine coolant temperature • Engine load • Engine speed | Function of: <ul style="list-style-type: none"> • Engine load • Engine speed |

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. |

| Update | Description |
|-----------------------------------|--|
| 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 Display > Signals & Ports > Port Units . |

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 ₂ - 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> |

| Block | EO Exhaust Emission | | | | |
|------------------|---------------------|----------------------|---|----------------------------------|---------------------------|
| | HC - Hydrocarbon | CO - Carbon monoxide | NOx - Nitric oxide and nitrogen dioxide | CO ₂ - Carbon dioxide | PM - Particulate matter |
| 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

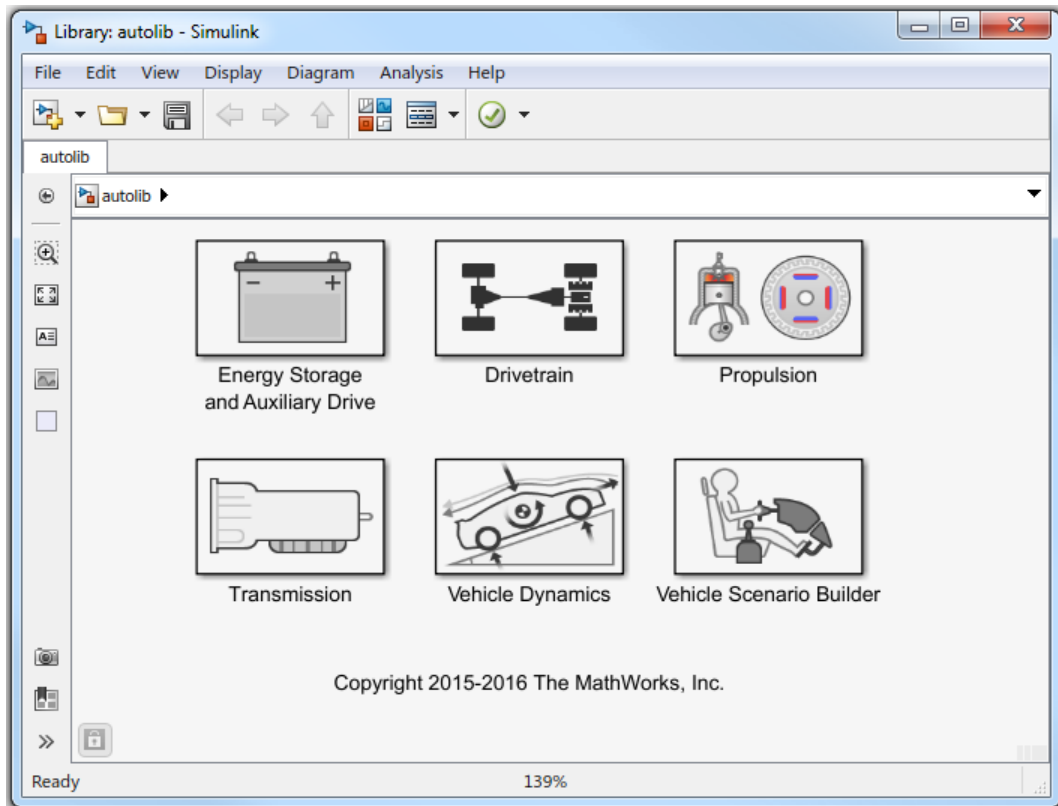
Powertrain Blockset has full vehicle reference application projects that you can use for design tradeoff analysis and component sizing, control parameter optimization, and hardware-in-the-loop (HIL) testing.

- 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