755 Phoenix Drive, Ann Arbor MI, 48108, USA
Phone: 734 668-2930 • Fax: 734 668-2877 • Email: info@carsim.com

CarSim 2020.0 New Features

This document lists notable new features in CarSim version 2020.0.

VS Solver: Architecture

New VS Command

A new VS Command `INIT_CURRENT_PATH_SWEEP` has been added. This command is designed to be used in conjunction with an assignment to `PATH_ID_DM`. The solver will search the entire path from end to end and assign the `Station` a value that is as close as possible to the vehicle's current X and Y global coordinates.

In the Run Control library, a new category { * CS 2020.0 – Lane Change } includes two datasets demonstrating the new VS Command in the context of a path change maneuver.

> **Note** When the `INIT_CURRENT_PATH_SWEEP` command is used, the user is unable to manually assign the `SV_STATION` state variable. This command is mutually exclusive with `INIT_CURRENT_PATH_SWEEP`.

Group ID for Events in Log File

The reporting of Events in the log file now includes the group ID, making it easier to identify which Events were triggered.

VS Solver: Models

Electric and Hybrid-Electric Powertrain Enhancement

The electric and hybrid-electric powertrain model now supports multiple electric motors. A new option is added to specify an electric motor on each drive axle. For example, a four-wheel (2-axle) passenger car can have two electric motors, and a 5-axle truck is able to have five electric motors.

The new option enables support of an electrified all-wheel-drive system without any mechanical connection between the front and rear axles (such a system is sometimes called *e-AWD*.) The torque distribution between the front and rear electric motors can be set by a user parameter or external command through Simulink, VS Commands, etc.

All electric motors are involved in the battery state-of-charge calculation.

Closed-Loop Path Follower

Three parameters have been added to allow the reference point for the path following driver model to be placed at an arbitrary location. The parameters `XREF_DM_F` and `XREF_DM_R` allow the reference point to be shifted longitudinally from the default locations of the front suspension (when driving forward) or rear suspension (when driving backward). The parameter `YREF_DM` specifies the Y coordinate of the reference point. Advanced users can employ these parameters to adjust the point of interest in steering the vehicle. For example, steer to have a wheel follow a path; or steer to move a corner of the vehicle in parking.

The parameters may be changed dynamically with VS Commands to adjust the reference point in advanced scenarios, as demonstrated in an example dataset.

Brake System Transport Delay

The brake system transport delay is now supported for all brake systems, regardless of control method. Previously, the only supported brake systems were those whose control was by master cylinder pressure with an open-loop table. The delay otherwise works as it did before, using a parameter `TLAG_BK` for each wheel to establish a pure time delay between the master cylinder and the line pressure of each wheel. Output variables for the line pressure of each wheel — e.g., `PbkL_L1` — have been added.

This improvement is particularly significant for vehicles with air brakes and/or trailers that have significant transport delays.

More details are available in the brake systems documentation.

VS Terrain for RT Systems

VS Terrain is now supported on all real-time hardware-in-the-loop (RT HIL) platforms with the exception of dSPACE DS1006/DS1005/DS1103.

VS Browser: Graphic User Interface (GUI)

New Multiple Vehicle Screen

A new library screen has been added: **Multiple Vehicles**, available from the **Tools** menu and the **Libraries** menu (Batch submenu). This screen facilitates running simulations from Simulink or SCALEXIO FMI that have multiple vehicles with full dynamics. It also has buttons for viewing the plot and animation results for each vehicle in the simulation.

Vehicle Assembly Screen

An option has been added to the pulldown control for selecting the rear suspension type on the **Vehicle: Assembly** screen, to choose “Solid axle (with wheel end compliance)”. By default, choosing a solid axle with the steering system screen set to “No rear steering” installs an axle module with no degrees of freedom for steer motion. As a result, wheel-end kinematics have no effect. The new option installs the more detailed module with steer capability, so full wheel-end compliance is supported.

Tire (External) Screen

Two additions were made to the **Tire (External)** screen.

1. An entitlement file field was added to specify the entitlement file path for Siemens MF-Tyre/MF-Swift version 7 and newer versions.
2. A diametral (yaw/roll) moment of inertia field was added. With the MF-Tyre/MF-Swift tire model option, there is a selection either specifying the diametral inertia on the screen or feeding the value from the tire property file (TIR).

Batch Run and Batch Matrix Screens

Both the **Batch Runs** and **Batch Matrix** screens have a new field for **Alternate Results Output Folder**. When this field has a valid folder, each executing either the Batch Run or the Batch Matrix will save the simulation results to this alternate directory.

Management of Deprecated Screens and Libraries

Over the years, as the models have been extended to include more options, new screens have been added to the Browser. To maintain backward compatibility, older screens have been kept for multiple releases but are eventually removed.

Starting with 2020.0, there is a formal handling of old screens. If Mechanical Simulation plans to remove a screen in a future release, it is formally *deprecated*. A deprecated screen and library will still exist and continue to function, but is identified in several ways to encourage moving the data to a newer library:

1. The screen has the word “Deprecated” in the title.
2. A popup message appears when visiting the screen in the browser indicating it is deprecated.
3. The screen appears in a summary of deprecated libraries if a database is rewritten.

Several screens have been deprecated (marked for removal). Each deprecated screen can be replaced by another screen with additional capabilities. In all cases, the replacement screens have been available for years.

Documents are provided in the **Help** menu with instructions for migrating data from deprecated screens to their replacements as well as showing the timetable for removal.

Screens deprecated in 2020.0

The following screens have been deprecated for CarSim 2020.0: they will still exist in 2020.1 (June 2020) but will be removed for 2021.0 (December 2020).

- Control: Speed (Closed Loop) vs. Station (Deprecated)
- Path: X-Y Coordinates (Deprecated)
- Powertrain: Lockup Schedule (Deprecated)
- Powertrain: Unlock Schedule (Deprecated)
- Transmission (7 gears, Deprecated)

Find Deprecated Datasets

The **Tools** menu has a new utility function: **Find All Deprecated Datasets in Database**. This command generates a report showing any dataset that currently uses a library that has been marked as deprecated. The report also includes links to documentation that will aid in migrating your data away from deprecated screens.

Activate Outputs for Writing

Output variables activated for post processing have become more robust. Previously, if **Plot: Setup** datasets were linked at a location in the run hierarchy where a variable was not yet defined, the variable would be silently omitted. Now the `WRT_` commands can bind to variables defined anywhere in the run.

Other Tools: Stacks

VS Scene Builder

The following improvements have been made to the VS Scene Builder tool:

1. Improvements to progress bars and current status.
2. New options in right-click context menu.
    a. Set scene origin and shift entire scene contents around new origin.
    b. Delete option.
3. New windowpanes for sections of the Scene Builder. Section panes can be moved out of the Scene Builder window or be rearranged.
4. Add name of current scene to the top of the Scene Builder window.
5. Scene Import screen in VehicleSim has new option in the table to loop or unloop paths.
6. Improved tile snapping and highlighting.
7. Right and Left-handed traffic tile filter in tile pane.

VS S-Function for Multiple Vehicles

The capability for running multiple CarSim vehicle models under Simulink that interact with each other has been available for many years. The operation has been simplified in version 2020.0 with the introduction of the **Multiple Vehicles** screen, mentioned earlier. When running with Simulink, a new Multiple Solver S-Function further simplifies the setup.

As with all the other CarSim S-Functions, communication between the new Multiple Vehicles S-Function and the VS Solver is facilitated by import and export variables and set up using the I/O Channels: Import and I/O Channels: Export screens.

When the Run button is clicked in the Simulink model, the Multiple Solver S-Functions check the working directory to ensure that an independent DLL file exists for a VS Solver. If they do not exist, each S-Function copies the installed DLL (e.g., `carsim_64.dll` from the `CarSim_Prog\Programs\Solvers` folder) and renames the copy appropriately. Each S-Function then uses a copy of the original VS Solver DLL with a distinct name associated with the block in the S-Function.

VS Terrain Library

There is now a library included in the VS SDK that allows users to create VS Terrain files from within their applications. These can then be used with a VS Solver running under Windows or Linux environments. The package includes full documentation and examples on how to start integrating and populating with surface mesh data.

VS Terrain Optimizations

The VS Terrain implementation introduced in the 2019.0 product stack has seen significant upgrades. VS Terrain file sizes should generally be smaller, load faster, and provide more rapid queries at run-time.

The build settings are now configurable through the API as well as the VS Terrain Utility. The VS Terrain Utility now includes command-line support, application settings for build parameters, a rebuild command, and a basic performance tester.

VS Connect Python Example

There is a new VS Connect Python API and example in the VS SDK. This example was built with the Unreal Engine CarSim example project. The example project contains python files for loading the VS Connect API, setting up a client VS Connect node, and a vehicle braking controller for following a lead vehicle.

Licensing for High Performance Computing (HPC)

CarSim solvers feature a new lightweight licensing mechanism that is well-suited for HPC installations, especially on Cloud or Virtualized hardware with ephemeral lifetimes. When using HPC Licensing, CarSim math model solvers can be run on fresh machine installations with minimal license setup. Licenses are managed by a central networked server.

Docker

CarSim 2020.0 is now capable of being run from docker. Docker is a technology that enables automation of processes within isolated runtime environments, including massively parallel swarms. For more information on the underlying technology, visit www.docker.com. To make the task of configuring this easier, a Dockerfile can be used to automate the initialization. Dockerfiles can be found in the download section of www.carsim.com. These files can be used to download, install, deploy and run a VehicleSim product with only a few simple commands.

Documentation

The following documents have been added to the **Help** menu:

1. HPC Licensing (Technical Memos)
2. Path X-Y Coordinates Screens (Deprecated Screens)
3. Powertrain Transmission Screens (Deprecated Screens)
4. Running with Multiple Vehicles (Tools)
5. Speed (Closed Loop vs Station) Screen (Deprecated Screens)

The following Reference Manuals have been updated:

6. System Parameters in VS Solvers
7. VS API
8. VS Browser
9. VS Commands
10. VS Solver Programs

The following Screen documents have been updated:

11. ADAS Sensors and Target Objects
12. Animator Shapes and Groups

13. Batch Matrix
14. Batch Runs
15. Brake System
16. Driver Controls
17. Powertrain System
18. Powertrain for Electric and Hybrid Electric Vehicles
19. Procedures and Events
20. Road Surface Visualization
21. Run Control
22. Suspension Systems
23. Tire Models
24. Vehicle Screens and Outputs
25. VS Scene Builder
26. VS Scene Tile Creation
27. VS Terrain

The following tech memos have been updated:

28. Automating Runs with the VS API
29. VS Solver Wrapper
30. 3DS MAX: OBJECT and OPEN SCENE GRAPH Setup and Export

Database

New Examples

The following subsection titles correspond to categories for new examples that all share the prefix "* CS 2020," e.g., *** CS 2020.0 – Animated Trace Lines**.

Animated Trace Lines

Two examples show how to mimic long-exposure photographs sometimes used to show motions visually. This is done by enabling the "ghost" rendering option in VS Visualizer for a few Reference Frames used to locate spheres that identify points of interest.

Dampers with 2D Tables

An example vehicle includes user-defined 2D tables to calculate shock absorber force based on both compression and compression rate.

EV with Axle Motors

Four example runs are included with AWD vehicles, one with electric power on both axles, and the other with hybrid front and electric rear. One example procedure is based on the EPA urban cycle; the other is a simple cycle with acceleration followed by regenerative braking.

Lane Change

As mentioned earlier, a new VS Command `INIT_CURRENT_PATH_SWEEP` was added to help deal with complicated path switching in which a road's global X-Y coordinates might have multiple valid Station (S) values along its path. One example might be the intersection of a highway and an on-ramp.

For CarSim 2020.0, an example scenario is demonstrated with two roads: the first road is defined as a straight path with a single segment while the second road consists of two segments: a constant radius circle followed by a straight segment parallel to the first road. In this scenario, the vehicle starts on the single segment straight road, then switches to the adjacent, two-segment road triggered via an Event.

In the example "Changing Paths: Similar Station", the vehicle starts on the long, straight road and the Event is triggered when Station is greater than 50 meters. When this happens, the Closed Loop Driver Model is switched to the second road and path. The assumption is that the vehicle's X-Y position on the new road will be similar to that on the first road, at a Station position of approximately 50 meters. Although both roads do indeed share this Station position relative to their respective origin locations, the global X-Y Coordinates of the two Station positions are not close to one another since the roads have different geometries. As a result, when the Events are triggered to switch paths, the Closed Loop Driver Model's preview point is set to a location on the new road that is sufficiently far from the vehicle's position on the original road (i.e., similar Station position, but sufficiently different global X-Y coordinates). The calculated steering wheel angle needed to steer the vehicle to this new location is very large and causes the vehicle to drive off the path.

In the example "Changing Paths: Closest Point", the new VS Command is used to search the second path and identify a Station position whose global X-Y Coordinates are significantly closer to the vehicle's position on the first road. This, in conjunction with the Events used to switch paths, results in a lane change maneuver that is easily handled by the Closed Loop Driver Model.

Low Speed Behavior

Recent changes in the closed-loop speed controller and path follower support scenarios involving driving backwards, and providing direct control of acceleration. The examples cover a range of conditions:

1. **3-Point Turn Accel Control** shows the vehicle making a three-point turn: it turns right, slows to a stop, reverses while turning to the left, stops (now facing about 180° from the original direction), and finally accelerates in the new direction. All control is closed-loop (other than the changing of transmission mode between forward and reverse), using path preview and acceleration modes for the speed controller, and path following for the steering.
2. **Alley Dock (Extra License)** shows how to drive in reverse while maneuvering a trailer. A pickup truck towing a trailer drives in reverse, making a 90° turn and backing the trailer

into a docking area marked by cones. In this case, the controller involves VS Commands that try to control the hitch articulation angle as needed to follow a preview point.

3. **Low speed Left Turn, Track Tire** shows a vehicle being controlled such that a point on the outside of the right-front wheel tracks a 90° marked turn. New parameters in the driver model are used to set the reference point at the tire, allowing the tracking to be within 20mm. (This example is inspired by Performance Based Standards used for heavy trucks.)
4. **Reverse DLC** shows a pickup truck following the same double lane change (DLC) that is used in the Quick Start Guide, but going in reverse.
5. **Reverse DLC: Trailer (Extra License)** shows the same pickup truck driving the DLC course in reverse with a trailer. As in the other trailer example, the controller involves VS Commands that try to control the hitch articulation angle as needed to follow a preview point.

Miscellaneous

This category has examples that are each one-of-a-kind.

1. **ADAS Sensor Collision Detection** shows how to setup custom HUD displays that make use of a built-in ADAS sensor whose distance detection is used to trigger conditions of imminent collision and contact. VS Visualizer is setup to show four alerts: a text-based collision counter, and three lights (detection, collision imminent, and contact).
2. **Brake System Transport Delay** demonstrates a pure transport delay for the brake line pressure going to trailer brakes, based on changes made to the VS Solver. In versions of CarSim prior to 2020.0, the brake system transport delay was only supported for open-loop braking control via the GUI table for Master Cylinder Pressure. For CarSim 2020.0, the Brake System Transport Delay is supported when the brake system command comes Open Loop Braking (as before), the Closed Loop Speed Controller, or via an import variable for the brake system (e.g., Master Cylinder Pressure).
3. **Generic EPAS, DLC** shows how CarSim can be setup to handle an EPAS power steering boost system, using Generic screens.

Multiple Autonomous Vehicles

Two sets of examples making use of the new Multiple Vehicle S-Function in Simulink — one with four vehicles and another with two vehicles — are demonstrated in a City Environment with ADAS Sensor Target detection. Each of the vehicles include rudimentary collision detection logic to avoid other vehicles and pedestrians, as well as wait at stop signs and respond to traffic lights.

As noted earlier, the new **Multiple Vehicles** library screen can be found from both the **Tools** menu and the **Libraries** menu under **Batch**.

Performing a simulation with more than two vehicle S-Functions requires an optional parallel solver licenses, but the simulation results for the four-vehicle example can still be viewed with a standard CarSim license.

Optimization

Two new examples have been added to demonstrate how CarSim can be used for optimization. Both examples use the same basic problem: Optimize two driver-model control gains to reduce the lap time of a vehicle around a racetrack. Due to the limited number of optimization variables, and the heuristic algorithms demonstrated, these are not considered to be true lap time optimization examples. Instead, they are intended to show how CarSim can be used to (1) solve a CarSim-based optimization problem externally using MATLAB and (2) solve a CarSim-based optimization problem internally using Events and VS Commands. The MATLAB example uses the MATLAB function `fminsearch`, which is based on the Nelder-Mead method. The Events/VS Command example uses an approach sometimes referred to as a compass search method.

Platooning

In addition to the new ADAS examples with multiple vehicles, the existing Five-Vehicle platooning example was updated to include numbered balloon assets that appear over each vehicle, to illustrate the order number assigned to each vehicle in the simulation. This updated example has the five vehicles in a mixed order to start, and as the simulation progresses, the vehicles align into sequential order, while each vehicle (excluding vehicle number one) follows the vehicle immediately in front of it.

The Two-Vehicle platooning example, and the Radar Active Cruise example remain unmodified, except that they too can be accessed through the Multiple Vehicle GUI.

VS Terrain

Two examples have been added to showcase the capability within CarSim 2020.0 to include VS Terrain files manually, by using the VS Command `VS_TERRAIN_FILE`. The first example involves one vehicle which traverses the main loop of MCity, using:

- A VS Terrain file that was manually imported,
- an .obj file illustrating the surface textures of the MCity terrain, and
- a manually imported x-y path.

The second example utilizes all elements from the first example, and includes a truck that traverses a curb from the MCity terrain, showing that the manual import of the VS Terrain file includes full 3D properties of the ground.

Updated Datasets

Updates to aerodynamics datasets

The following aerodynamics datasets have been updated:

- Aero: Fx > Spline: 0.30 at 0 deg.
- Aero: Fy > Spline: 0.5 at 90 deg., now Aero Fy > Spline: 2 at 90 deg.
- Aero: My > Spline: 0.15 at 0 deg.
- Aero: Mz > Spline: -0.05 at 90 deg., now Aero Mz > Spline: 0.2 at 35 deg

Improved dually pickup

The “dually”, or dual rear wheel (DRW), pickup has been replaced with an updated version, with the following changes of note:

- Long (8 ft) bed, reflecting current market availability
- New sprung mass data, better representing the inertial properties of the new long bed configuration
- Spring and damping rates have been updated to better perform with the new mass data.

New Animation Assets

Creating Custom Signs

As signs vary around the world, it is prohibitive to attempt to design all possible sign images to provide within the VehicleSim product. For Version 2020.0, there is a new feature that enables users to create an '.OSG' Animation Shape File of a sign with a user-defined image to include in their databases. Using this feature, the user can easily apply any image onto a blank sign similar to other sign animation objects provided in the standard product. To be able to do this, the user will need an image editing program that can work with .tga files. A sample ‘Blank (Logo) Sign’ dataset has been added to the Animation Shape File Link library.