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

CarSim 2023.0 New Features

This document lists notable new features in CarSim version 2023.0.

VS Solver Architecture

Stages of Model Calculations

The architecture of the VS Solver has traditionally had two stages available for model extensions via VS Commands and custom wrappers connecting with application program interface (API) functions: kinematics and dynamics. Wrappers for external software tools such as Simulink dealt only with the whole model, sharing import and export variables once per time step.

The architecture was extended to organize model calculations into four specific stages each timestep:

1. **State :** the vehicle state (position and speed state variables) is known, and information about the environment is also updated (station, ADAS targets and sensors, wind, and path information for the driver model).

2. **Control :** the built-in driver controls are applied to provide steering and speed control variables.
3. **Kinematics :** variables are calculated that depend on the position and velocity information from the vehicle and controllers.
4. **Dynamics :** variables are calculated using the remaining equations in the model. ODEs for the entire VS Math Model are integrated to obtain the values available at the start of the next timestep.

The new organization was made to provide tighter connections with external software for controllers and replacements for vehicle components. It also improves options available for advanced users extending models with VS Commands and embedded Python.

Import and Export Arrays for Connecting to External Software

VS Solvers may be run under the control of wrapper programs in simulation environments that combine external model components with the VS Math Model, exchanging information using arrays of import and export variables. In past versions, communication with the simulation environment takes place once each timestep, such that the VS Math Model receives import variables before performing any calculations, and then copies output variables into an export array that is shared after all calculations in the VS Math Model have been completed.

As noted above, the calculations made in the VS Math Model each timestep are now organized into four stages, to support closer timing when extending models with external software. Features were added to support the use of multiple wrappers with a single VS Math Model:

- `OPT_MULTI_WRAPPERS` is a new user parameter available when using import/export arrays. It can be set to 0 (default) or 1. When set to 1, the VS Solver will connect with up to four wrappers connected for the simulation (one for each stage of the model calculations). Six new arrays are added (three for import and three for export) in this mode, for a total of eight arrays for exchanging information.
- `IWRAPPER` is a new hidden index parameter, managed in the same way as `IUNIT`, `IAXLE`, etc. that allows existing `IMPORT` and `EXPORT` commands to be used as needed to set exchange arrays for four wrappers.
- Each timestep, the VS Math Model makes up to four passes through the model calculations (each under the control of a different wrapper), performing only the calculations associated with the currently active wrapper.
- Machine-generated documentation files for Import variables now identify the stage in which the variable is used.
- Machine-generated documentation files for output variables now mention the stage in which the variable is calculated.

To support these new capabilities, four new Simulink S-Functions are provided in the release for Windows and non-RT Linux, as described in the VS Wrappers section (page 4).

The new process has no noticeable effect on the computation speed for the simulation.

VS Commands

Several new commands were added, and the effects of some existing commands were changed.

1. New commands `EQ_STATE` and `EQ_KIN` were added to create new equations in more stages of the model calculations. Related commands were added: `DELETE_EQS_KIN`, `DELETE_EQS_STATE`, `RESET_EQS_KIN`, and `RESET_EQS_STATE`.
2. Equations added by the `EQ_IN` command now apply after the new State stage and before the new Control stage. (Most existing examples using `EQ_IN` continue to work as originally intended.)
3. The `DEFINE_OUTPUT` command was improved to work more effectively. `SET_OUTPUT_SHORT_NAME` was added to allow the manual creation of short (ERD compatible) output names by the user.
4. The `RETURN` command used in defined functions was improved to reduce problems. `DEFINE_LOCAL` was improved to eliminate option of indexed local variables.
5. Error checking was added for lengths of new variables and parameters. The name lengths were extended to 48 characters; longer names now generate error messages.

Echo Files

The display of information in Echo files generated for each simulation has been extended to indicate which parameters or table properties have been assigned values by reading from file, as opposed to those that keep default values. Some changes in this release are:

1. The convention for identifying a parameter whose value was not specified by any of the input files is to prefix the description with `[D]` (default). This has been extended to include properties of Configurable Functions.
2. A parameter `OPT_ECHO_DEFAULT` can disable the display of the `[D]` prefix, which is sometimes helpful when monitoring differences in Echo files. Another option has been added, in which the Echo file shows only parameters and Configurable Function properties that were specified in input Parsfiles. These files are much shorter and can be convenient for some applications involving automation and large numbers of runs.

Other Improvements

1. All error messages were reviewed; many were updated to provide more specific information about the cause of the error.
2. New VS API functions were added to provide a location in the equations of the model for import and output variables; these are described in the API Reference Manual in the VS software development kit (SDK). C examples in the SDK demonstrate the new functions.

VS Math Models

Terramechanics-Based Tire Model

There is a new terramechanics-based tire model for predicting tire forces and moments when operating on soft soil. The tire is treated as rigid, in contrast to a typical CarSim or TruckSim tire (soft soil, rigid tire vs. hard pavement, flexible tire). While this type of tire model has no standard implementation, these kinds of models are often referred to as Bekker-Wong models, referring to two prominent terramechanics researchers.

The new terramechanics-based model is referred to as VS Terramechanics: Rigid Wheel (VS TMRW). The implementation makes use of the standard tire interface (VS STI) and is specific to CarSim and TruckSim, including some additional effects for use in this setting. As VS TMRW uses VS STI, there are no specific VS Browser screens associated with it. Rather, the Tires (External) and Generic Table screens are used to create and populate instances of VS TMRW.

VS TMRW is only supported for CarSim and TruckSim 2023.0 running on a 64-bit Windows operating system. For more information on other limitations of the new model and how to use VS TMRW, refer to Help > Tires > Terramechanics Models.

Trailer Aerodynamics and Crosswind

Wind has traditionally been defined using direction and speed that are calculated with Configurable Functions of time and location (station along the vehicle reference path). For vehicles with trailers, there was no easy way to account for a crosswind hitting the trailer after it reaches a station that the leading unit had already crossed. The VS Math Model now includes multiple wind Configurable Functions for simulations with multiple vehicle units. For trailers, an equivalent station variable is calculated using station for the leading unit, adjusted by the distance between the front reference point and trailing hitch point for the leading unit.

VS Tire Tester

For external tires, VS Tire Tester iteratively solves for the wheel center height needed to produce the requested tire vertical force. Beginning in version 2023.0, the maximum number of allowed iterations, as well as the solution tolerance, may be adjusted with the math model keywords `TT_MAX_ITERATIONS` and `TT_TOLERANCE_FZ`, respectively. The default values of 1000 iterations and 1.0 N are consistent with the calculations of previous versions.

Importing a Path Preview for Controllers

The new VS Command `INSTALL_PATH_IMPORT` defines 20 import variables `IMP_PATH_X`*i* and `IMP_PATH_Y`*i* (*i* = 0, 1, …, 9). These are the coordinates of 10 preview points, expressed in the vehicle coordinate system. If all 20 are activated for import using the `IMPORT` VS Command, then the steering controller and/or speed controller will not reference the built-in path (that of `PATH_ID_DM` and potentially `LTARG_ID_DM`), but instead use the imported path preview. See the Driver Controls document for further details.

Support Electric Motor on Twin-Clutch Differential

Electric motor has not been available on the twin-clutch differential axle. In this release, the solver changes to support an electric motor on the twin-clutch differential axle. Therefore, an electric motor can be installed on axle by command line or `OPT_MOTOR_ON_WHEEL 0` when `opt_twin_clutch` is 1.

Advancing the Interface to External Tire Models

In past versions, VS Math Models connected to external tire models one tire at a time to get forces and moments of each tire. Some third-party tire models support parallelized calculation of all tires. The VS Solver and external tire model interface have been extended for the parallelized calculation of the external tire model namely COSIN FTire. The new interface speeds up the simulation by about a factor of three compared with the old interface for FTire.

The external tire model interface to COSIN FTire also adopts a dynamic library loader that automatically detects the FTire solver as installed by the FTire installer.

Miscellaneous

1. In past versions, the path follower driver model was always included in the model. It has been made optional and is installed with a new command `INSTALL_DM_PATH_FOLLOWER`. As before, the DM controller can be deactivated by setting the parameter `OPT_DM` to zero.
2. The parameter `STEER_MAX_TORQUE` can be set to shut down a run if the steering torque exceeds the given value. `STEER_MAX_TORQUE` is turned off by default. A value greater than 0 will turn on the max torque limit functionality.
3. Back-spinning of wheel with electric motor is prevented when the battery is recharged with regenerative brake.

VS Wrappers

Multiple Simulink S-function Blocks

Four versions of the VS S-Function have been made that connect with the VS Math Model during different the stages of the calculations made each timestep that were described earlier (page 2): `vs_state` (State), `vs_ctl` (Control), `vs_kin` (Kinematics), and `vs_dyn` (Dynamics). The new S-Functions were added to the VS library that is accessed from Simulink. These VS S-Functions provide multiple connection points between the VS Math Model and Simulink that are applied in series each timestep by Simulink.

Any combination of two to four of the new S-Functions may be used. The timing in Simulink is such that the last S-Function is applied at the end of the timestep, and the others are applied earlier, in sequence with other blocks in the Simulink model. The result is that replacement of parts of the vehicle model (tires, springs, brake controllers, etc.) is handled inline, without the delay of waiting until the next timestep.

Examples have been added to illustrate the use of the new serial S-Functions, and a technical memo has been added to describe the operation.

dSPACE RT Unreal Live Animation and Remote Data Access

The CarSim solver wrapper for dSPACE SCALEXIO 64-bit Linux has been enhanced to utilize VS Connect to provide remote data access and synchronization capabilities.

This feature can be used to produce Live Animation of RT simulation runs using the VehicleSim Dynamics plugin for Unreal Engine. The VS Connect features of this wrapper also allow advanced users to use the VS Connect API to write custom C/C++ programs that remotely access Imports and Outputs of the CarSim solver while it is running on the RT system.

For more information, see the example Run **{RT: dSPACE} SCALEXIO: Unreal Engine Live Animation**", and the new tech memo **Unreal Engine: Live Animation with Simulink**.

Support for Speedgoat Real-Time Platform

The VS solvers now support the Speedgoat Real-Time platform: Performance and Mobile. The RT system is QNX 7.1 64bit. Software is MATLAB/Speedgoat R2001a or newer. The Speedgoat platform is supported by the existing VS Browser (using the **Models: Transfer to Remote RT Target** library). Two example runs were added to the database and the **Model Extensions and RT > External Models and RT Systems** documentation was updated.

VS Browser: Graphic User Interface (GUI)

Minor changes were made to some of the existing screens.

1. The **Models: Simulink** screen has a new drop-down control for models with three kinds of VS S-Functions:
    a. The Simulink model has a single VS S-Function for the simulation.
    b. The Simulink model has multiple VS S-Functions, each for a separate vehicle. The vehicle simulations are run in parallel in Simulink using the **Tools > Parallel VS Math Models** screen.
    c. The Simulink model has multiple VS S-Functions, each for a different stage of the VS Math Model calculations. The S-Functions are run in series to support in-line extensions to the VS Math Model from Simulink. When this mode is selected, the screen blue links to four sets of I/O ports, each for a different stage and associated S-Function.
2. The **Payload: Custom** screen has a new drop-down control to specify the reference for the Z coordinate of the payload: sprung mass coordinate system or trailer front hitch height. (This option is like one that was added in 2022.1 to the **Payload: Box Shape** screen.)
3. The TruckSim **Vehicle Dolly** screens have a similar drop-down control for locating animation shapes relative to the sprung mass coordinate system or trailer front hitch height.
4. The **Front Twin-Clutch Differential** and **Rear Twin-Clutch Differential** screen adds a drop-down control to specify either with or without electric motor on center and two electric motor links.

5. External tire interface to COSIN FTire model adopts a dynamic library loader which automatically detects FTire solver under FTire installation. Therefore, **Tire (External)** screen has changed to remove the **Tire program file (DLL)** field when FTire model is selected.

VS Visualizer Pro

VS Visualizer Pro uses the power of Unreal Engine to bring the simulation results from CarSim to a new level of realism. A demo release is available as a separate download from www.carsim.com with a CPAR with a collection runs that can be viewed and modified in CarSim and highlight some of what is possible with this next generation of VehicleSim visualization.

VS Software Development Kit (SDK)

The `vs_output` API has been updated to allow `.mat` file I/O.

Documentation

The following documents were added to the **Help** menu:

1. Deprecated Items > Solid Axle Kinematics
2. Real-Time and DS Systems > Speedgoat Guide for VehicleSim Products
3. Tires > Terramechanics Models
4. Technical Memos > VS Support for Multiple S-Functions
5. Technical Memos > Unreal Engine: Live Animation with Simulink

The following Reference Manuals have been updated:

6. System Parameters in VS Math Models
7. VS Browser (GUI and Database)
8. VS Commands
9. VS Commands Summary
10. VS Math Models

The following Screen documents have been updated:

11. Animator > Vehicles and Sensor Targets
12. Controls > Driver Controls
13. Model Extensions and RT > External Models and RT Systems
14. Paths and Road Surfaces
15. Payloads
16. Setting Up Import and Output Variables

17. Suspension Systems
18. Tires > Tire Models
19. Vehicle Screens and Outputs

The following Technical Memos have been updated:

20. Example: Extending a Model with VS Commands and the API
21. GPS and UTM Coordinates

The following Real-Time and DS System documents have been updated:

22. dSPACE RT Guide
23. VI-Grade DriveSim & VehicleSim Dynamics

The following SDK documents have been updated:

24. The VehicleSim API
25. VS Output API: Reading and Accessing VS Output Files

The following Unreal documents have been updated:

26. Unreal Engine & Python Co-simulation using VS Connect
27. VehicleSim Dynamics plugin for Unreal Engine example using VS Connect

Database

Additions were made in some of the run categories. The following new categories (with associated CPAR archive files) were added.

Aerodynamics

An aerodynamics dataset has been added which uses data representative of a small van to model the aerodynamics of an enclosed rental trailer. The aerodynamic effects of the trailer are then compared with the tow vehicle alone with an updated Crosswind Test and a new Virtual Wind Tunnel test. The Virtual Wind Tunnel test calculates the aerodynamic drag, side, lift, roll, pitch, and yaw coefficients from the measured forces and moments for validation purposes.

Alternative Coordinates

VS Math Models can now generate Universal Transverse Mercator coordinates, as shown in a new example run.

Driver Model

The path preview import variables, p. 4, are demonstrated with some new examples in the *** Driver Model** category/CPAR.

Echo File Options

The Quick Start Guide example was run with different settings that affect the information shown in Echo files.

Electrified Twin-clutch Differential

An example with an electric motor on each of front and rear twin-clutch differential is included.

MF-Tyre on Simulink

Several examples extend the VS Math Model by connecting with Siemens MF-Tyre using Simulink.

Multiple S-Function Wrappers

Several Simulink models were made using the new four S-Functions for different stages of the model. Some are in the category ***Multiple S-Function Wrappers**.

There is also an example with two S-Functions in the category *** MF-Tyre on Simulink**, and another in an older category **Kinematical Preview Demo**.

New Signs

Several runs were made showing many highways signs (VS Visualizer resources) that were added to the database.

New VS Commands

Several runs were made showing new VS Command features. Multi-index variable arrays are used to simulate multiple flat tires. An improved steer controller example has been added.

Terramechanics

Several runs illustrated the new soft-soil model option based on terramechanics.