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

BikeSim 2022.1 New Features

This document lists notable new features in BikeSim version 2022.1.

VS Solver: Architecture

VS Commands

1. The `PARTIAL` and `PARTIAL2` VS Commands have been added to return the partial derivative of a Configurable Function in either the row or column direction. The new `INVERSE` function returns the inverse of a Configurable Function if it is available.
2. `INSTALL_DM_IMPORTS` has been added to allow for installing Driver Model imports only. Previously, the only way to install them was to use `INSTALL_DM_OUTPUTS`, which also (still) installs them.

Other Improvements

1. The embedded Python included with all products has been updated to 3.10.2.
2. The user can now specify pre-processing and post-processing callback functions to be before and after the solver executes. These are available via the VS API or the GUI.
3. Support dSPACE SCALEXIO Linux 64-bit real-time system (dSPACE RLS 2022A and newer)
4. Support OPAL-RT RT-Lab Linux 64-bit real-time system.
5. The solver is now working with 32-bit and 64-bit LabVIEW.

VS Math Models

Powertrain Improvements

Improvements were made in the automated shifting behavior in the powertrain, which affects the powertrain behavior and the options for setting up driver control options. Most of the improvements involve the closed-loop clutch controller.

1. The location of parameters in the Echo file was adjusted to better link parameters to parts of the overall powertrain model. For example, throttle delay was moved from the Engine section (that only exists for powertrains with an internal combustion engine) to the overall powertrain system section.
2. Parameters `T_CL_START` and `T_TH_START` were added to provide more options for setting the timing for automated clutch and throttle modulation during shifting.
3. The calculations of clutch, throttle, and shift were improved to handle new requests for clutch/throttle activity if an activity was already in progress. For example, if a new shift is request before a shift in progress has ended, the new shift is started, with the start time adjusted automatically to avoid discontinuous jumps in clutch.
4. The closed-loop clutch behavior was extended to included stopping (dis-engaging the clutch at very low speed to avoid stalling the engine) and re-starting (re-engaging the clutch when attempting to accelerate). Several new state variables were added to support the new transition events.
5. A parameter `TH_MIN_CL_ACCEL` was added to support the automatic re-engagement of a clutch when accelerating from a stopped condition.
6. Upgraded some Import variables to support interactions with internally calculated values via `ADD`, `MULTIPLY`, and `REPLACE`. The Imports are: `IMP_AT_CLUTCH`, `IMP_AV_ENG`, `IMP_GEAR_TRANS`, `IMP_IENG`, `IMP_INV_CAP_TC`, `IMP_MENG_REACT`, `IMP_MODE_TRANS`, `IMP_RM_TC`, `IMP_R_EFF_TR`, and `IMP_R_GEAR_TR`.
7. Two types of the electrified powertrain model, **Parallel hybrid** (`OPT_HEV` = 4) and **Full electric** (`OPT_HEV` = 2), are supported in BikeSim.
8. Target acceleration speed control (`OPT_SC` = 5) is supported in BikeSim.

Improvements for Moving Object Import Variables

Upgraded some Import variables for moving objects to support interactions with internally calculated values via `ADD`, `MULTIPLY`, and `REPLACE`. The Imports (for object #1) are: `IMP_HEAD_OBJ_1`, `IMP_MSG_OBJ_1`, `IMP_PITCH_OBJ_1`, `IMP_ROLL_OBJ_1`, `IMP_S_OBJ_1`, `IMP_TYPE_OBJ_1`, `IMP_VIS_OBJ_1`, `IMP_V_OBJ_1`, `IMP_X_OBJ_1`, `IMP_YAW_OBJ_1`, `IMP_Y_OBJ_1`, and `IMP_Z_OBJ_1`.

VS Browser: Graphic User Interface (GUI)

64-bit Version of the Browser

The browser `BikeSim.exe` is a 32-bit application that runs on both 64 and 32-bit versions of Windows. As such, it can load 32-bit plug-in libraries such as the VS Solver `bikesim_32.dll` but is not able to use 64-bit libraries.

Most users have been working with 64-bit versions of Windows, and many engineering software tools are now available only as 64-bit applications and libraries. For example, the last version of 32-bit MATLAB from MathWorks was 2015b. That means any recent versions of MATLAB and Simulink will work only with the 64-bit VS Solver plug-in libraries.

The 2022.1 release includes two versions of the Browser: `BikeSim.exe` (still 32-bit) and `BikeSim_64.exe` (64-bit). The plan from Mechanical Simulation is to drop the 32-bit versions of our tools in the 2023.0 release. (Recent releases have already included both 32-bit and 64-bit versions of the VS Solver libraries, VS Visualizer, and other tools.)

Mechanical Simulation recommends using the 64-bit version unless there is a need to maintain compatibility with 32-bit tools. Given that recent versions of MATLAB/Simulink are only 64-bit, there is slightly better compatibility if the runs made without Simulink use the same VS Solver library as the runs made with Simulink.

Improvements in Existing Library Screens

Existing Library screens were modified.

I/O Channels: Write

The Browser now supports access to pre-processing and post-processing callback functions (I/O Write screen) so the user has the option to execute their own programs before or after the solver is run. As mentioned above, this capability can also be accessed with the VS API.

The filename suffix option has been restored for auxiliary outputs.

Electrified Motorcycles

Two existing library screens are modified, and three new library screens are added to support electrified motorcycles. The library screens are:

1. Powertrain: Shaft Drive (modified),
2. Powertrain: Chain Drive (modified),
3. Powertrain: Hybrid/Electric System (added),
4. Powertrain: Hybrid/Electric Power Management Control (added), and
5. Powertrain: Electric Motor Torque (added).

Miscellaneous Changes

1. Changes were made in Powertrain screens such that unused options are not installed. For example, the `INSTALL_ENGINE` command is not applied for electric powertrains.
2. The **Control: Clutch Shifting Timelines (Closed Loop)** screen was modified to include yellow fields for the new parameters `T_CL_START` and `T_TH_START`.
3. The **Powertrain: Electric Motor Torque** screen includes a new checkbox and a new yellow field for an optional reduction gear. If the checkbox is checked (default is unchecked), the yellow filed appears to set the reduction gear ratio which is used to scale the input and output of the motor torque configurable table, i.e. `SPIN_SCALE_M_MOTOR_MAX` and `MMOTOR_MAX_GAIN`. Also, the gear ratio is used to modify the motor rotor inertia (`I_MOTOR`).
4. The **Control: Shifting (Open Loop)** and **Control: Shifting (Closed Loop)** screens were both changed to allow only three kinds of Configurable Functions that are appropriate for gear as a function of time: Constant, Table (steps, flat-line extrapolation), and Equation.
5. Two more plot links were added to the **Generic VS Commands** screen.
6. The **Tools** menu was modified to clarify the searching of existing runs for uses of the dataset currently in view.

VS Visualizer

VS Visualizer has added a preferences option to force X or Y plot axis labels to show. Users with a small VS Visualizer window and many plots may have hidden axis labels due to automatic plot window scaling. Forcing the axis labels to show will allow users to view VS Visualizer at their preferred window size.

Licensing

The Command-Line License Manager can now run as a Windows Service, allowing for the application to be started automatically when the system is booted. Additionally, running the License Manager as a Service allows for the application to be started, paused, or stopped using the Microsoft Management Console.

Documentation

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

1. Powertrain > Electric and Hybrid Electric System (BEV/HEV)
2. Technical Memos > Change Units of VS Math Model Variables
3. Technical Memos > vs_sf VS Connect Server
4. Tools > Database Builder

The following Guides and Tutorials were updated:

5. Quick Start Guide

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 COM Interface
11. VS Math Models
12. VS SDK: The VehicleSim Software Development Kit
13. VS Table Tool
14. VS Visualizer

The following Screen documents have been updated:

1. ADAS Sensors and Target Objects
2. Aerodynamics
3. Animator > Bike Shape Assembly
4. Animator > Camera Setup
5. Animator > Rider Shape Assembly
6. Animator > Shapes and Groups
7. Animator > Reference Frames
8. Animator > Sounds
9. Animator > Vehicles and Sensor Targets
10. Brake System
11. Generic Data > Generic Data Screens
12. Generic Data > Generic Table
13. Generic Data > External Parsfile
14. Model Extensions and RT > Custom Forces and Motion Sensors
15. Model Extensions and RT > External Models and RT Systems
16. Model Extensions and RT > Import and Export Variables
17. Model Extensions and RT > Path Detectors
18. Paths, Road Surfaces, and Scenes > Paths and Road Surfaces
19. Paths, Road Surfaces, and Scenes > Road Surface Visualization

20. Paths, Road Surfaces, and Scenes > VS Terrain
21. Payloads
22. Plot Setup
23. Powertrain > Electric and Hybrid Electric Systems (BEV/HEV)
24. Powertrain > Powertrain System
25. Procedures and Events
26. Rider Controls
27. Steering Systems
28. Suspension Systems
29. Tire Models
30. Tools > Atlas GPS Tools
31. Tools > Calculator Screen
32. Tools > Calculator Tool for Tables
33. Tools > VS / ERD File Utility
34. Vehicles > Motorcycles and Rider Bodies
35. Vehicles > Three-Wheel Motorcycles

The following Technical Memos have been updated:

36. HPC Licensing
37. Numerical Integration in VS Math Models
38. Validation of VS Vehicle Models
39. VehicleSim License Manager (VSLM)
40. VS Solver Wrapper

The following Real-Time document was updated:

41. RT-Lab Guide

The following SDK documents have been updated:

42. The VehicleSim API
43. The VS Vehicle Module Simulation Integration Utility
44. VS Output API: Reading and Accessing VS Output Files
45. VS SDK: The VehicleSim Software Development Kit

Database

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

Electric Powertrain

Two full electric powertrain examples; two pre-transmission parallel hybrid powertrain examples; and three post-transmission parallel hybrid powertrain examples are added under *** Electric Powertrain** category. Those examples involve EPA Urban and EPA Highway battery/fuel economy speed profiles; and open-loop acceleration and re-generative brake maneuver.

Impaired rider

The closed-loop driver model includes a parameter which can be used to delay the application of the calculated steering input by the desired time. New examples have been added which make use of this parameter to model the effects of an impaired driver. The preview time is also shortened proportionally to include an impaired driver's reduced ability to focus on the road ahead.

Parametric sweep example

A new example in the category *** Parametric Sweep** uses the new pre- and post-processing callback feature (I/O Channels: Write, p. 3) to perform a sweep of a parameter of interest. Using a Python script, the run data is duplicated, adjusted, and run with various parameter settings in a pre-processing step. The post-processing step is then used to overlay the results.

Path data from parametric equations using Calculator: Symbolic screen

A new example (**Roads and Intersections** > **Path from Parametric Equations**) shows how the Calculator: Symbolic library screen can be used to convert a parametric plane curve into X-Y coordinates. These coordinates are then used to create a reference path for the bike to follow.

Speed controller

A target acceleration speed control example which utilizes speed controller option of `OPT_SC` = 5 is added under *** Speed Controller** category.

VS Command Examples

Copies of the Quick Start run were made to add outputs generated using new VS Commands Inverse (inverse of a Configurable Function) and Partial (partial derivative of a Configurable Function expression).