

BikeSim 2020.1 New Features

- VS Solver: Architecture 1
 - Specifying Model Layout 1
 - VS Commands 2
- VS Solver: Models 2
 - New Options for Attaching ADAS Sensors and Moving Objects 2
 - Path Detection 2
 - Surface Rolling Resistance Coefficient is a Function of S and L 2
 - Flexible Swing Arm Frame 3
 - Front Fork CG Lateral Offset 3
 - Rider Path Preview Point Capability 3
 - Consecutive Runs on dSPACE ds1006 Board Without Unloading 3
- VS Browser: Graphic User Interface (GUI) 3
 - Find All Unreferenced Datasets 3
 - Find All Deprecated Assets 4
 - Re-Write All Runs Menu Item is Renamed 4
 - Clutch Timeline and Parameter Screens 4
 - Suspension screens 4
 - Miscellaneous Changes to Support ADAS Scenarios 5
- Other Tools: Stacks 5
 - VS Software Development Kit (SDK) 5
 - VS Terrain Utility 5
 - HPC Licensing Logging 6
 - VS Visualizer 6
- Documentation 6
- Database 7
 - Deprecated Animator Assets 7
 - New Examples 7

This document lists notable new features in BikeSim version 2020.1.

VS Solver: Architecture

Specifying Model Layout

When running a simulation, previous versions received the model layout (suspension types) from the Simfile. This is now done with a VS Command `MODEL_LAYOUT` that is located at the top of the Parsfile and provides a text description of the layout, such as `FIXED_CASTER` for a bike with a fixed-caster front suspension. Until this command is provided, the VS Solver only recognizes some system-level parameter names. Nothing related to the math model is installed until the `MODEL_LAYOUT` Command is applied.

VS Commands

Defined Functions

Defined functions no longer require a return variable when called. If a Defined function has a nominal return that is not needed, it can be called without the return and the return value is ignored.

Indexed Parameters

Indexed parameters (such as `IYY_WHL(1)`) can now accept variables or equations for their indices. The parsing of indexed parameters is also more robust.

VS Solver: Models

New Options for Attaching ADAS Sensors and Moving Objects

ADAS sensors may now be attached to moving target objects that are not part of the vehicle. These sensors can be used to sense other targets, using the standard sets of ADAS detection variables.

Target objects may now be attached to the vehicle sprung mass. These may be detected from sensors attached to moving objects. This supports the use of sensors in targets to detect imminent collision.

Path Detection

The *path detector*, for a given path ID and lateral target ID, fits a cubic polynomial to three target points to create an instantaneous preview path. The cubic polynomial is constructed such that it approximates a clothoidal segment (Euler spiral). The preview path outputs include curvature, rate of change of curvature, and lateral offset of the path relative to the vehicle. The implementation uses a Kalman filter to estimate the curve fit based on noisy target point measurements.

The path detector is a standalone feature which calculates outputs based on the specified path information only; it does not control the vehicle or account for the occlusion of the path by moving objects. In this regard, it can be considered a more advanced version of the traditional preview point.

The path detector may have multiple instances, allowing, for example, the left and right markings of a lane to be detected based only on a reference path and LTARG datasets. Lane centering examples demonstrating this are included; they use outputs from the path detector(s) to calculate steering requests via VS Commands.

For more information on the path detector, refer to **Help > Model Extensions and RT > Path Detectors**. This covers the basic operation, technical details, and the new library/screen **Custom Forces and Motion Sensors > Path Detectors for External Driver Control**.

Surface Rolling Resistance Coefficient is a Function of S and L

In past versions, the road parameter `RR_SURF` defined the surface coefficient for tire rolling resistance and was fixed for a given road surface. The coefficient is now defined with a Configurable Function that varies with S and L, as is done with road friction. The GUI was updated

to replace the keyword `RR_SURF` with `RR_SURF_CONSTANT`. Tables can now be used for road surfaces where the rolling resistance varies with position.

Flexible Swing Arm Frame

A flexible-frame option is added either to enable or disable the lateral bending and twisting flexibility of the rear swing arm suspension. This flexibility causes the lateral displacement of wheel center, camber and steer angle change on the rear suspension. The lateral bend is characterized by the lateral stiffness and damping at the wheel center, while the steer angle change is constrained with a kinematical function of the lateral displacement. The twisting motion (for camber change) is characterized by the torsional stiffness and damping of the swing arm.

If the frame flexibility is disabled, the rear swing arm frame is treated as rigid and the swing arm doesn't bend and twist relative to the main frame.

Note: The above frame flexibility option is identified by the BikeSim model layout codes: `fixed_caster`, `variable_caster`, and `three_wheeled` with and without suffix `_flex`. The model layout code appears above the **Vehicle** blue link on the **Run Control** screen. Also, it appears in the beginning of the Echo files, for example:

```
MODEL_LAYOUT VARIABLE_CASTER_FLEX.
```

In the model layout codes, the rear frame flexibility option (suffix `_flex`) is distinguished from the existing front fork frame flexibility option (prefix `rigid_`.) For example, if the front fork is rigid and the rear swing arm is flexible, the model layout code: `rigid_variable_caster_flex`.

Front Fork CG Lateral Offset

The lateral coordinate of the front fork suspension CG is now available as an input parameter.

Rider Path Preview Point Capability

The rider path preview point is viewable in the VS Visualizer by new output variables for the preview point coordinates: `X_RM`, `Y_RM`, and `Z_RM`.

Consecutive Runs on dSPACE ds1006 Board Without Unloading

With a one-time download of the VS Solver to the dSPACE DS1006 board, the simulation can be executed at least 8000 times consecutively.

VS Browser: Graphic User Interface (GUI)

Find All Unreferenced Datasets

A new option **Find All Unreferenced Datasets** has been added to the VS Browser under the **Tools** menu. This produces a list of every dataset in the current database that has no dataset referencing it. This option can be helpful in maintaining the database and identifying currently unused datasets.

Find All Deprecated Assets

A new option **Find All Deprecated Assets** has been added to the VS Browser under the **Tools** menu. This will produce a list of every dataset in the current database that has a reference to a deprecated animator asset. Deprecated assets reside in the `BikeSim_Prog\Resources\Animator\Deprecated_Assets` folder of your installation directory. Selecting this option from the **Tools** menu will also produce a list of missing animator assets and write the results to the log file.

The recommended practice is to perform this step on a database before upgrading it to a newer version of the software. If any deprecated animator assets are identified, they should be corrected prior to the database upgrade.

Re-Write All Runs Menu Item is Renamed

The **Tools** menu command to **Re-Write All Runs** has been renamed to **Re-Write every 'All Parsfile' sent to the Solver or VS Visualizer**. This renaming was done to provide more clarity for what action was being performed: to re-write every Parsfile in the **Runs** library in the database. This includes the `All.Par` files that are used to provide information to the VS Solvers and VS Visualizer.

Clutch Timeline and Parameter Screens

BikeSim has two screens for specifying automatic shifting with a clutch. One is named **Control: Clutch Shifting Parameters (Closed Loop)** and the other **Control: Clutch Shifting Timelines (Closed Loop)**. The Timeline screen has generally been more popular, but the VS Solvers used the parameters shown on the Parameter screen and showed those keywords and values in Echo file regardless of which screen was used for a simulation.

In 2020.1, the Solvers were changed to use the Timeline parameters, such that they are now shown in Echo files. The Timeline screen has been changed to use the keywords now recognized by the VS Solver. Further, the Parameter screen has been modified to calculate the Timeline parameters and include the proper keywords in the dataset Parsfile.

The 2020.1 VS Browser is backward compatible with datasets imported from older versions. Be aware that the older Parameters screen is now deprecated and is scheduled to be removed from the browser in version 2021.0.

Suspension screens

Suspension: Fork screen

The lateral coordinate for the front fork suspension's CG is now available as a user-defined parameter.

Suspension: Swing Arm screen

A checkbox for the flexible-frame option is added either to enable or disable the lateral bending and twisting flexibility of the rear swing arm suspension. The screen layout has been reorganized to support additional data fields and links.

Miscellaneous Changes to Support ADAS Scenarios

ADAS Sensors for Range and Tracking screen

The options for attaching the sensor to a part in the vehicle have been extended to include two new options:

1. The vehicle sprung mass.
2. The current moving object. This allows the sensor to be attached to a target object, enabling detection of collisions with a target object attached to a vehicle.

Single Moving Object (Custom) screen

The options for specifying the motion of the object have been extended to include attaching the object to the sprung mass of the vehicle unit. This is done in support of simulations with multiple vehicles, as well as for simulations where the sensor is attached to a moving object with the intent of detecting where a collision occurs with the vehicle.

Links to Procedures datasets

Blue links on Generic group screens can now link to **Procedures** datasets. This greatly enhances setting up simulations with multiple vehicles.

Generic screens

More miscellaneous blue links now allow Event datasets and ADAS Sensor and Moving Object datasets to be linked.

Run Control screen

It is now possible to link to a dataset from the **Generic Data Group (More)** library for the Math Model blue link.

Other Tools: Stacks

VS Software Development Kit (SDK)

Python APIs for `vs_output` and `vs_vehicle` have been added to the SDK. A standalone C example for the `vs_vehicle` API has also been added. The Python API for the solvers has been updated and the old API (`Simulation.py`) has been deprecated.

VS Terrain Utility

The tool has been upgraded to support conversions of OBJ files. It uses materials from the MTL file similarly to the materials specified in an FBX to associate friction and rolling resistance values.

A merge function has been added to consolidate multiple VSTERRAIN files. This is useful if converting multiple lanes generated from the VS Browser to form a road surface representation.

The performance test has been updated to be more precise.

HPC Licensing Logging

For installations using the HPC (High Performance Computing) Licensing option, the HPC Licensing Server application now writes log files to the Windows user directory or Linux /var/log directory.

VS Visualizer

VS Visualizer can now reload/refresh simulation data without the need to be closed and then relaunched. If the simulation results have changed (i.e., the simulation was modified and rerun) but VS Visualizer is still open, the user may press the hotkey Ctrl+F5 or select **Reload data** from the VS Visualizer's **File** menu. This will load the new data to the currently active VS Visualizer window.

Documentation

The following Reference Manuals were removed from the **Help** menu and moved to the VS SDK:

1. VS API
2. VS Connect API
3. VS Vehicle
4. VS Output/Table API

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

1. Clutch Control (Closed Loop) Screen (Deprecated Items sub-menu)
2. Deprecated Animator Assets (Deprecated Items sub-menu)
3. Path Detectors (Model Extensions and RT sub-menu)
4. VS Commands Summary (Reference Manuals sub-menu)
5. VS SDK: The VehicleSim Software Development Kit (Reference Manuals sub-menu)

The following Reference Manuals have been updated:

6. System Parameters in VS Solvers
7. VS Browser (GUI and Database)
8. VS COM Interface
9. VS Commands
10. VS Solver Programs

The following Screen documents have been updated:

11. ADAS Sensors and Moving Objects
12. BikeSim Suspension Systems
13. BikeSim Vehicle

14. Paths and Road Surfaces (Paths, Road Surfaces, and Scenes sub-menu)
15. Preferences (Tools sub-menu)
16. Procedures and Events
17. Rider Controls
18. Running with Parallel Solvers (Tools submenu, renamed from “Multiple Vehicles”)
19. Vehicles and Sensor Targets (Animator sub-menu)
20. VS Scene Builder (Paths, Road Surfaces, and Scenes sub-menu)
21. VS Terrain (Paths, Road Surfaces, and Scenes sub-menu)

The following Technical Memo was updated:

22. Automating Runs with the VS API
23. Validation of VS Vehicle Models

Database

Deprecated Animator Assets

Animator assets that have been moved to the folder `BikeSim_Prog\Resources\Animator\Deprecated_Assets` are marked for deprecation. Users can find these assets using the **Find Deprecated** assets tool option in the VS Browser. After two product releases (1 year), the assets will then be removed from the product installations. There are a few methods to include these assets if upgrading an old database to the current version, described in the *Deprecated Animator Assets* document.

New Examples

The following subsection titles correspond to categories for new examples that all share the prefix “* BS 2020.1,” e.g., * **BS 2020.1 – Flexible Swing Arm**.

Flexible Swing Arm

New examples were added that use the new option to have twist flexibility in the swing arm. This includes a highspeed simulation showing chatter, some K&C tests showing the new flexibility, and linearization examples showing root locus plots and Bode frequency-response plots.

Parallel Solvers

Two example simulations that were shown in past versions are redone to use the Parallel Solvers tool. The two simulations are done from the Parallel Solvers screen (**Tools** menu). One has the sensor calculations in Simulink, while the other uses an ADAS sensor in vehicle 2.

Path Detector

Four examples have been added to demonstrate various uses of the new Path Detector feature:

- Detecting a single path (*Path Detector Outputs*)
- Detecting multiple paths (*Multiple Path Detections*)
- Using a detected path for closed-loop steering control (*Lane Keeping Assist System* in CarSim/TruckSim, *Target Lean from Detected Path* in BikeSim)
- Detecting an imported target path (*Imported Target Points*).