TruckSim 2022.1 New Features

This document lists notable new features in TruckSim 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. A calculated read-only parameter `NDIFF` is shown, in support of user-defined VS Commands. This is the maximum number of differentials that might exist, given the number of drive axles created with the `INSTALL_POWERTRAIN` command.
3. Parameters `LIMIT_DOWNSHIFT` and `LIMIT_UPSHIFT` were added, to enable automatic shifting that can skip gears. For example, if `LIMIT_UPSHIFT` = 2, then the shift controller can potentially shift directly from gear 2 to 4. The default limits are 1, which replicates shifting in older versions.
4. 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.
5. 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.
6. 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.
7. A parameter `AV_ENG_LOW_CLUTCH` was added to support the automatic disengagement of a clutch at low speed if there is risk of stalling the engine.
8. A parameter `TH_MIN_CL_ACCEL` was added to support the automatic re-engagement of a clutch when accelerating from a stopped condition.
9. 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_CLT_D1_2`, `IMP_CLUTCH_D1`, `IMP_GEAR_TRANS`, `IMP_IENG`, `IMP_INV_CAP_TC`, `IMP_MENG_REACT`, `IMP_MODE_TRANS`, `IMP_MY_OUT_D1_L`, `IMP_MY_OUT_D1_R`, `IMP_M_DIFF_D1`, `IMP_M_OUT_D3_F`,

IMP_RM_TC, IMP_R_EFF_D1, IMP_R_EFF_TR, IMP_R_GEAR_D1, and IMP_R_GEAR_TR.

Trailer Front Hitch

The origin of a trailer sprung mass coordinate system is used to locate the front hitch point, the center of gravity (C.G.) of the sprung mass, the reference locations of the axle suspension(s), and possibly the location of a rear hitch point. The X-Y-Z local coordinates of the front hitch point for a ball/pintle or generic hitch are (0, 0, H_H_FRONT) where H_H_FRONT is a parameter for the height of the point relative to the origin. Output variables have always been available for the global coordinates of the origin, as Xo_*i*, Yo_*i*, and Zo_*i*, where *i* is the number of the trailer units (2 or higher).

Additional outputs have been added in this release for the global coordinates of the front hitch point: Xo_HF_*i*, Yo_HF_*i*, and Zo_HF_*i* (where HF indicates Hitch, Front).

The location of the front hitch point can be more convenient for animation objects attached to the trailer sprung mass, such as payloads, hitch structures, and miscellaneous structures that make up the sprung mass. The Browser screens for specifying the animation assets for a trailer unit were updated to allow the new front hitch points to be used to animate items attached to the sprung mass, sometimes more conveniently than use the origin point used to locate suspension heights.

Trailer Backing Controller

The *trailer backing controller* (TBC) calculates the steering wheel angle if the vehicle is reversing, the single preview point driver model is being used, and the vehicle is a supported type. The steering wheel angle is manipulated such that the hitch articulation angle steers the trailer along the desired path.

The INSTALL_DM_TBC VS Command is used to install the TBC, typically from the miscellaneous yellow field on the closed-loop driver model screen, as this is an extension of the single preview point mode. There is one tuning parameter, the proportional control gain, with math model keyword TBC_GAIN.

For more information on the TBC, please refer to the help file for the menu item **Help > Controls > Trailer Backing Controller**.

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 `TruckSim.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 `TruckSim_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: `TruckSim.exe` (still 32-bit) and `TruckSim_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.

Vehicle Sprung Mass

VS Math Models have not included dimensions of the sprung masses, as they are not part of the equations of motion. However, in version 2020.1, VS Math Models could attach target objects to sprung masses so they can be detected by ADAS sensors. Dimensions were added to the VS Math Models (they are `LEN_SM`, `WID_SM`, and `HT_SM`) and those dimensions are available to define sizes of moving objects or for advanced users to include in VS Command equations.

The sprung mass screens have always had three yellow fields for dimensions used by VS Visualizer to scale animation assets (these dimensions are `X_LENGTH`, `Y_LENGTH`, and `Z_LENGTH`). In versions 2020.1 to 2022.0, the values from these yellow field were written twice, using both sets of keywords.

In 2022.1, the two library screens for sprung masses were extended to explicitly provide two values for each dimension: one set is for scaling animation assets for the video and the other set is for sizing a target moving object if one is attached to the sprung mass. The library screens are:

1. Vehicle: Lead Unit Sprung Mass,
2. Vehicle: Trailer Sprung Mass.

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.

Animation: Vehicles and Targets

As mentioned earlier, the VS Math Model now provides X-Y-Z coordinates for a front hitch point for trailers. This screen has a drop-down control to choose whether animation assets should be located relative to the sprung mass origin point (typically close to the ground) or at the front hitch point. The setting is written to the Parsfile for the screen with the keyword `OPT_TRAILER_REF_FRAME`. This is used when a run or video is made from the **Run Control** screen: the value of the parameter used by the Browser to generate Reference Frames for VS Visualizer using the selected point. Note that the keyword is only used within the Browser; it is not recognized by the VS Math Model.

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: Engine** screen includes a new yellow field for a minimal speed setting for the clutch controller, with keyword `AV_ENG_LOW_CLUTCH`. If the powertrain includes a clutch operating with the built-in closed-loop controller, this engine speed might be used to dis-engage the clutch at very low speed to avoid stalling the engine.
4. 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`).
5. 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.
6. Two more plot links were added to the **Generic VS Commands** screen.
7. The **Tools** menu was modified to clarify the searching of existing runs for uses of the dataset currently in view.
8. The **Control: Steering by the Closed-Loop Driver Model** screen automatically applies the VS Command `INSTALL_DM_IMPORTS` if the single-point preview is selected (`OPT_DM = 3`). The Imports are not activated, but they are available. Previously, this did not occur and there was no easy way to access the imports.

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.

Simulink S-function Wrapper VS Connect Server

The Simulink s-function wrapper `vs_sf` has been enhanced with the addition of an embedded VS Connect server. Parsfile keywords added to yellow fields in the VS Browser can be used to enable and configure the VS Connect server within the `vs_sf` s-function. When enabled and configured via Parsfile parameters, this feature provides access for remote VS Connect Nodes to set imports and retrieve outputs of the VS Solver as it executes within Simulink.

This feature can also provide remote VS Connect read/write access to other data (signals) within Simulink that are external to the VS Solver by employing custom solver imports/exports for this purpose.

Examples of remote VS Connect clients that can connect to `vs_sf` include:

- The VS Connect Client S-function (`vs_connect_sf`), which may be running in the same Simulink model, or in a separate model which may itself be running on a separate computer.
- Custom C/C++ code utilizing the VS Connect library (available in the VS SDK).
- Custom Python code utilizing the VS Connect Python wrapper (available in the VS SDK).
- The forthcoming release of the VehicleSim Dynamics Plugin for Unreal Engine, which will include components that can be easily configured to provide "Live Animation" within Unreal Engine of VehicleSim solvers running remotely within Simulink.

For an example of how to configure the VS Connect server of `vs_sf`, see the example Run:

```
{External Control, Wrappers} Unreal Engine Live Animation
```

Documentation

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

1. Controls > Trailer Backing Controller

2. Technical Memos > Change Units of VS Math Model Variables
3. Technical Memos > Example: Self-Steer Axle
4. Technical Memos > vs_sf VS Connect Server
5. Tools > Database Builder

The following Guides and Tutorials were updated:

6. Quick Start Guide

The following Reference Manuals have been updated:

7. System Parameters in VS Math Models
8. VS Browser (GUI and Database)
9. VS Commands
10. VS Commands Summary
11. VS COM Interface
12. VS Math Models
13. VS SDK: The VehicleSim Software Development Kit
14. VS Table Tool
15. VS Visualizer

The following Screen documents have been updated:

1. ADAS Sensors and Target Objects
2. Aerodynamics
3. Animator > Camera Setup
4. Animator > Shapes and Groups
5. Animator > Reference Frames
6. Animator > Sounds
7. Animator > Vehicles and Sensor Targets
8. Brake System
9. Controls > Driver Controls
10. Controls > Electronic Stability Control (ESC)
11. Generic Data > Generic Data Screens
12. Generic Data > Generic Table
13. Generic Data > External Parsfile
14. Hitch

15. Model Extensions and RT > Custom Forces and Motion Sensors
16. Model Extensions and RT > External Models and RT Systems
17. Model Extensions and RT > Import and Export Variables
18. Model Extensions and RT > Path Detectors
19. Paths, Road Surfaces, and Scenes > Paths and Road Surfaces
20. Paths, Road Surfaces, and Scenes > Road Surface Visualization
21. Paths, Road Surfaces, and Scenes > VS Terrain
22. Payloads
23. Plot Setup
24. Powertrain > Electric and Hybrid Electric Systems (BEV/HEV)
25. Powertrain > Powertrain System
26. Procedures and Events
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

The following Deprecation Memo was updated:

35. Powertrain External Transmission Screens

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

The following miscellaneous documents have been updated:

46. 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.

Articulated bus

Examples provided with the previous release (2022.0) were re-done to better show the articulations of the connection.

Automatic clutch control

Improvements in the built-in closed-loop controller for mechanical clutches are shown for some examples involving bringing the vehicle to a start and then restarting.

Impaired driver

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.

New vehicle configurations

A new vehicle was added: a two-axle pickup towing a two-axle gooseneck trailer.

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. 5) 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.

Self-steer axle examples

A new set of examples demonstrate a self-steer axle implemented as an external steering model with VS Commands. The modeling concept and implementation details are discussed in a new technical memo, **Help > Technical Memos > Example: Self-Steer Axle**. Various examples show low speed maneuvering, performance under braking, and raising/lowering the axle using additional VS Commands. These are included in the *** Self-Steer Axle** category/CPAR.

Trailer backing controller

Various trailer backing examples are now updated to use the built-in trailer backing controller (TBC); these have been re-organized into the *** Trailer Backing Controller** category/CPAR. For more information on the TBC, please refer to the Trailer Backing Controller help file, located in Help > Driver Controls.

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).