

Running a VS Vehicle Model in Simulink

This memo describes how a VehicleSim (VS) model from a product such as CarSim or TruckSim runs within a Simulink model. A simple antilock brake system (ABS) example is used in various forms all VS products. (The examples shown in this memo were taken from TruckSim 7.)

This memo is not intended for specialists in ABS or brake system model, and does not go into detail about the brake system aspects of the Simulink model. Rather, the point is to show how to set up a VS vehicle model to work in the Simulink environment.

Locating a Simulink ABS Example

The example for this memo is a Simulink ABS model that is used for a few vehicles and maneuvers. To find a similar ABS example in your software, locate the dataset from the **Run Control** library: from the **Dataset** menu ①, find the category **Simulink** ② and a dataset name similar to the TruckSim example **ABS, Split Mu: 2A Conv. Van** ③.

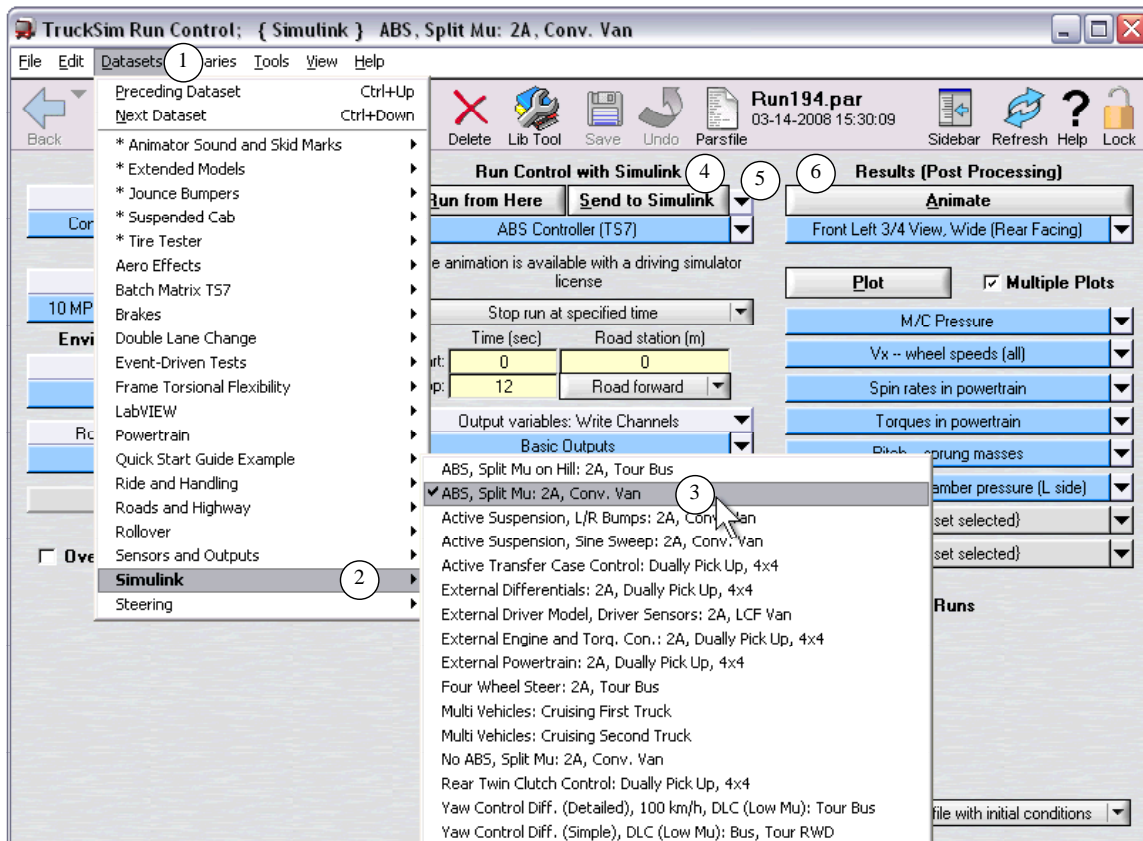


Figure 1. Navigating to the Simulink example for an ABS system.

Once you are viewing a run dataset involving a Simulink model, click the **Animate** button (6) to view an animation of the run made with Simulink, showing a vehicle coming to a controlled stop on a challenging surface (e.g., a truck on a split-mu surface).

Making a New Run Without Simulink

The vehicle model used in TruckSim for this example has a full brake system that can be automatically replaced when the Simulink model is added.

To make a run without Simulink, create a copy of the example dataset from Figure 1. Click the pull-down control (5) and select **[No linked library]** to break the link with the Simulink dataset. The buttons **Run from Here** and **Send to Simulink** will automatically be replaced with a single button: **Run Math Model**. Click this button to make a run without Simulink, and then click the **Animate** button (6). It should show the same vehicle on the same surface, but with less desirable behavior (Figure 2).

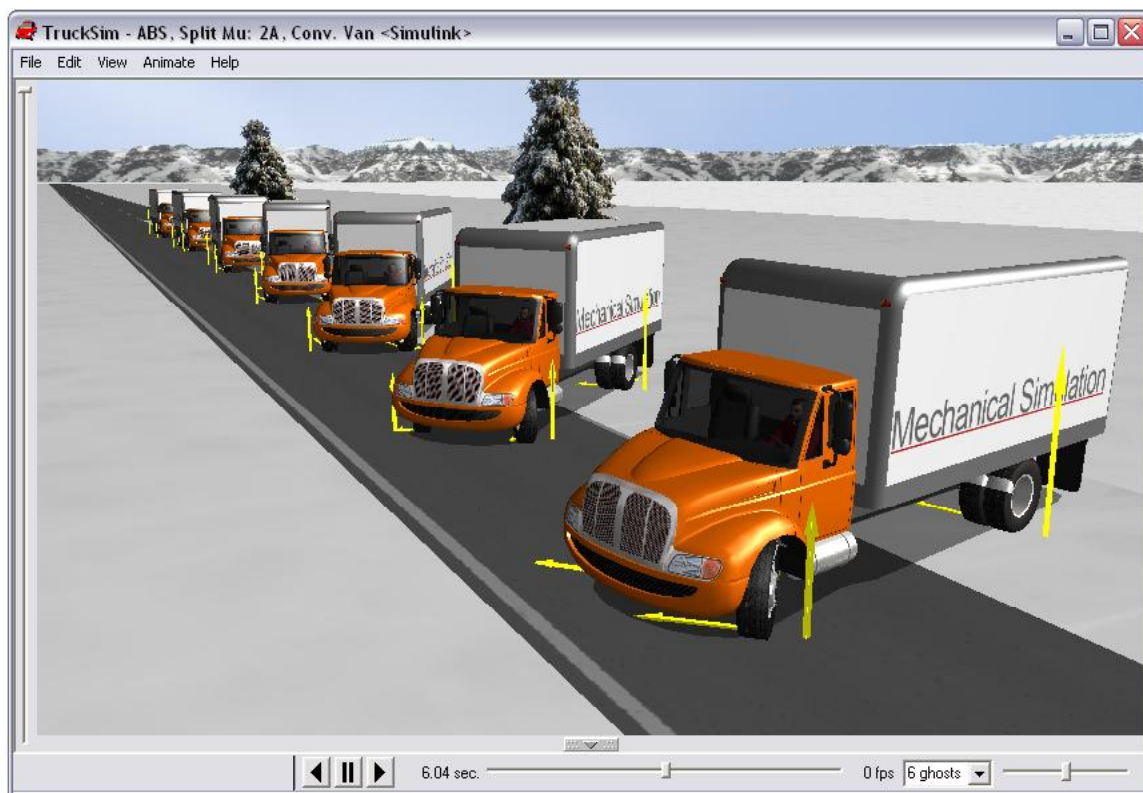


Figure 2. Braking test without Simulink ABS controller.

This quick comparison shows that the basic vehicle model works fine by itself or when extended using Simulink, and that the main difference is in the brake control that is modeled in Simulink.

Viewing the Simulink Model

Go back to the original dataset (e.g., **ABS, Split Mu: 2A Conv. Van**, as shown in Figure 1). Bring up the Simulink model, similar to the one shown in Figure 3, by clicking the button **Send to Simulink** (3).

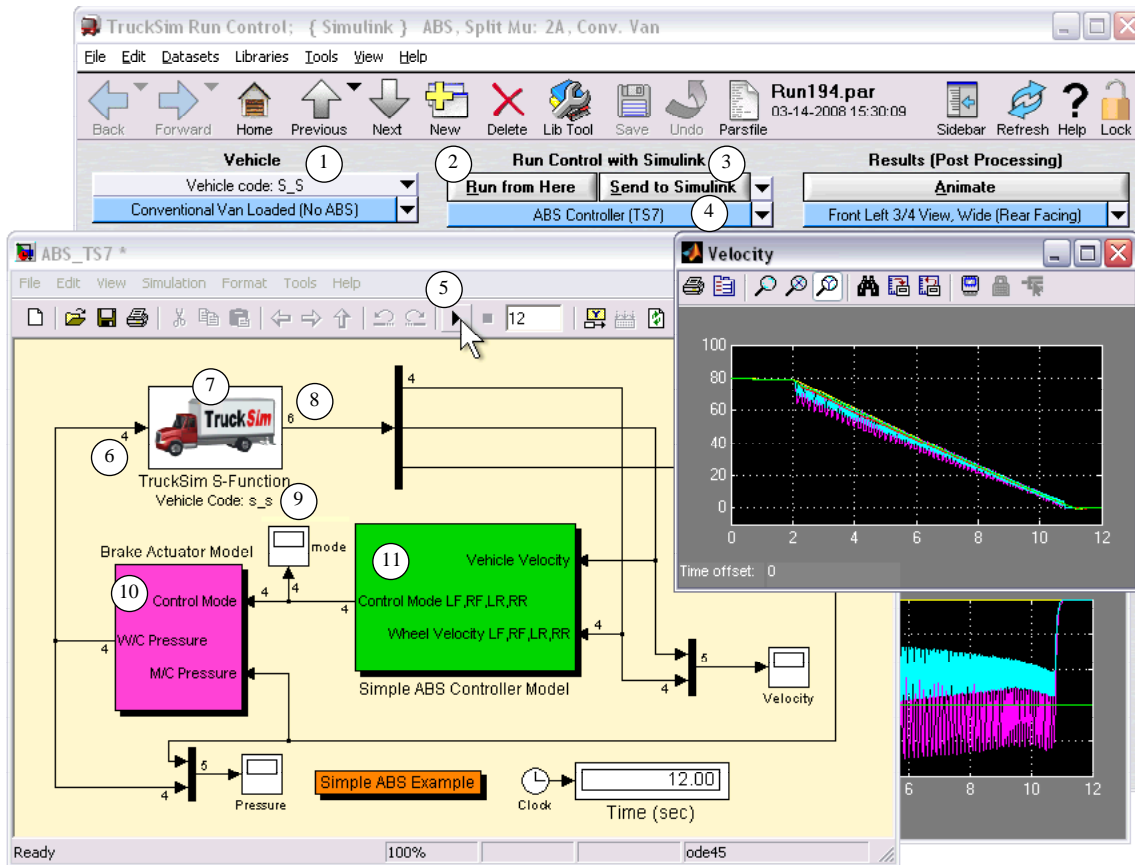


Figure 3. Simulink model with one VS vehicle model.

Note The model shown in Figure 3 was edited from the installed example in TruckSim to show the signal dimensions and to display a more print-friendly background color.

To make a run from Simulink, click the play button (5) (or use the **Simulation** menu item **Start** or the keyboard shortcut Ctrl+T). Alternatively, the run can be started from the VS browser by clicking the **Run from Here** button (2).

The Simulink model shown in Figure 3 has three main parts:

1. the vehicle model (with the TruckSim picture (7)),
2. a control model (the green box (11)), and
3. an actuator model (the magenta box (10)).

The remainder of this memo will describe the main features of the vehicle block (7).

The Vehicle S-Function Block

The vehicle block (7) is a Simulink S-Function with several properties that are set dynamically to both represent the vehicle as specified in the VS software (e.g., TruckSim), and to fit with the signal requirements of the Simulink model. Three of the properties that are set automatically are visible for confirmation in Simulink:

1. The vehicle code indicates the vehicle configuration that will be simulated. For this example, the vehicle has two axles with solid-axle suspensions and the code is `s_s`. This code is shown on the **Run Control** screen (1) and confirmed by the display of the S-Function block in Simulink (9).
2. The signal dimension for input to the VS S-Function block (e.g., 4 input variables) (6) is sent in that parameter file (called a *parsfile*) from the VS database, and must match the requirements of other blocks in the Simulink model that connect to the VS model. A VS model can be set up to handle anywhere from zero to several hundred input variables.
3. The signal dimension for output from the VS S-Function block (e.g., 6 output variables) (8) is also sent in the *parsfile* from the VS database, and must match the requirements of other blocks in the Simulink model that connect to the VS model. A VS model can be set up to handle anywhere from zero to several hundred output variables.

The VS browser handles the vehicle code automatically. If you change the dataset in the **Run Control** screen to specify a different vehicle with a different configuration (e.g., `i_s`, a vehicle with an independent front suspension), then both codes are changed to match the new vehicle. On the other hand, the definitions of the input and output signals for the VS S-Function block must be set up manually, as described later.

Note If you change the vehicle, its properties, or the sets of import or export variables from the VS browser, you must click the **Send to Simulink** button (4) to update the model in Simulink.

Simulink Configuration Parameters

The Simulink model will typically involve underlying differential equations that are solved by Simulink using numerical methods. For example, the ABS model shown in Figure 3 has a brake actuator system (the magenta box (10)) with fluid dynamics represented with transfer function blocks. Some of the methods used to solve the equations can be viewed and adjusted. To do this, use the Simulink **Simulation** menu item **Configuration Parameters** to display a dialog box such as the one shown in Figure 4. (The appearance on your screen might differ from the figure, depending on your version of Simulink.)

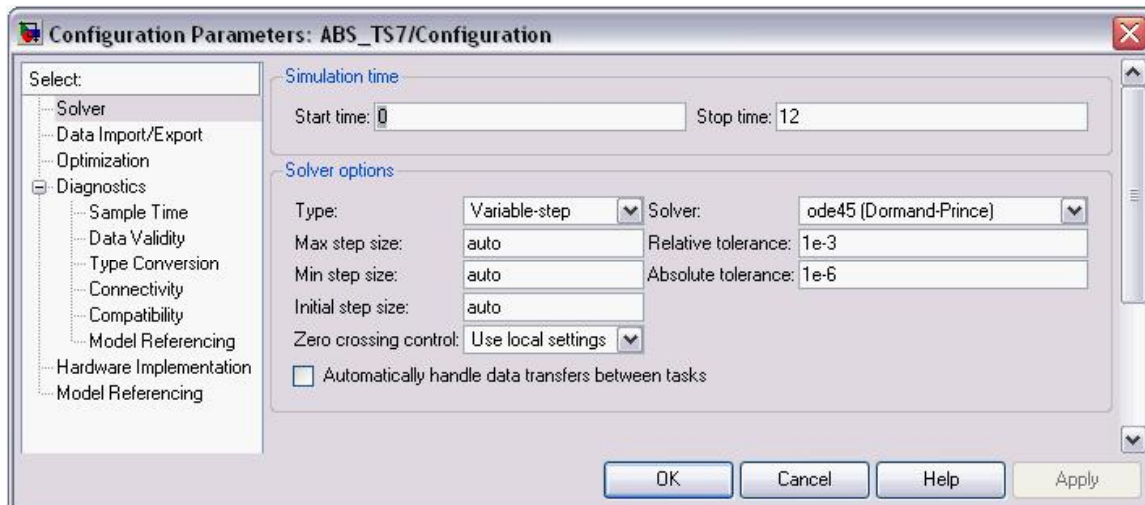


Figure 4. Simulink Configuration Parameters.

The VS S-Function block is a discrete system. This means that the VS solver rather than Simulink handles the internal differential equations of the vehicle model.

You can configure the Simulink model as a continuous system with either fixed or variable-step solvers, so long as the Simulink parameter settings are compatible with the discrete time step used by the VS solver (described in the next section).

When transferring control from the VS database to Simulink, simulation parameters such as the start time and stop time are sent automatically.

The VS Simulink Dataset

The connection between the VS database and Simulink is made with a dataset from the **Models: Simulink** library. In the example, the linked dataset is **ABS Controller (TS7)** (4) (Figure 3). Click on the link to view the **Models: Simulink** screen, shown in Figure 5.

This screen has four main settings that specify how a vehicle model will be connected to Simulink. The most important is the name of the Simulink MDL file (1). This pathname is used when you click the **Send to Simulink** or **Run from Here** buttons from the **Run Control** screen ((2) and (3) in Figure 3).

The second setting that will be mentioned here is the checkbox ((3) in Figure 5) that determines the relationship between the time step set on the **Run Control** screen and the one sent to Simulink. As noted in the on-line pop-up help in the figure (activated by right-clicking on the checkbox), the VS solver supports numerical computation methods that perform calculations at one half of the interval specified on the **Run Control** screen. Communication with Simulink can be done at every opportunity (the half-step) or only at major steps. When this box is checked, the time step seen by Simulink is $\frac{1}{2}$ the time step specified on the **Run Control** screen. (This box should normally be checked.)

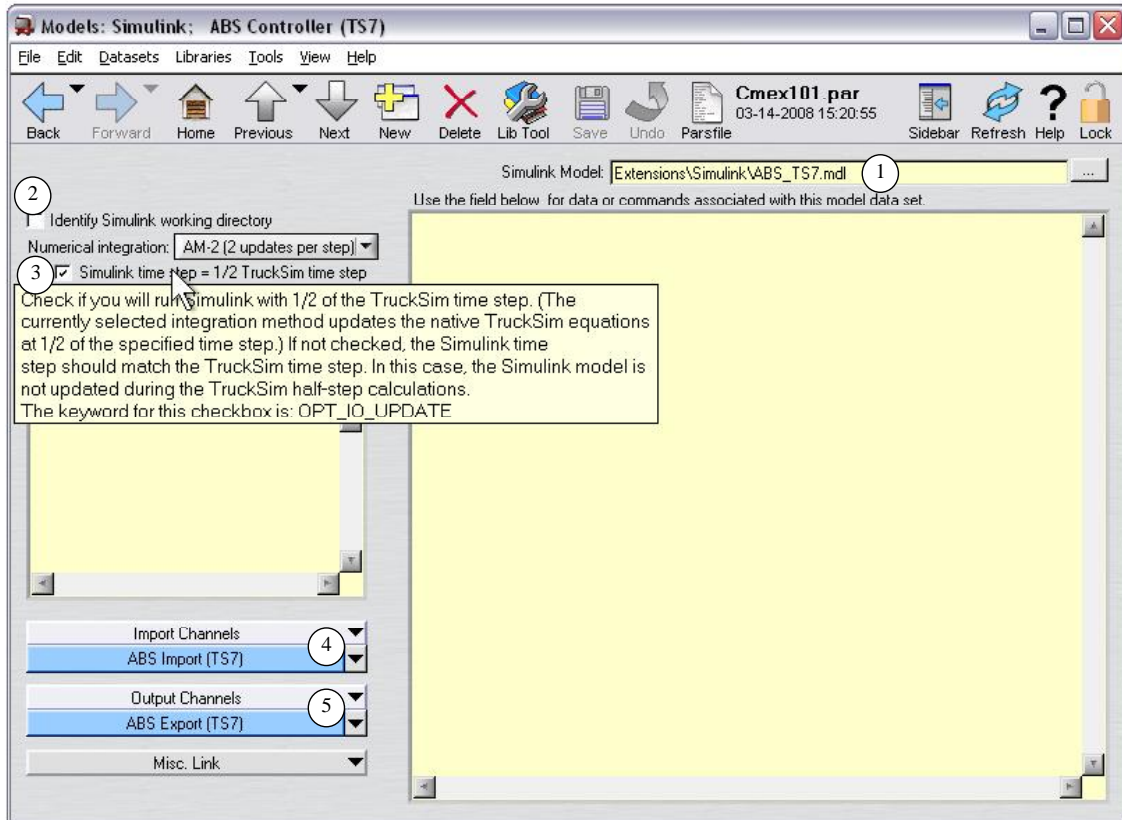


Figure 5. Models Simulink dataset for the ABS controller.

If the Simulink model is set to run with a fixed time step, as required for real-time systems, then the VS solver time step must be an integer multiple of the Simulink time step. When the $\frac{1}{2}$ step box is checked (③ in Figure 5), then the $\frac{1}{2}$ step must be an integer multiple of the Simulink time step. For example, if the time step specified on the **Run Control** screen is 0.001 and the $\frac{1}{2}$ step box is checked, then the vehicle S-Function appears to Simulink as a discrete system that must be updated at time intervals of 0.0005. The Simulink time step can be 0.0005, 0.00025, 0.00001, or any other value that divides evenly into 0.0005.

Note For more information about the numerical integration methods used by the VS solver programs, see the *VehicleSim Solver Program Reference Manual*.

The third and fourth settings on this screen are links to datasets that define the sets of variables that will be activated for import into the VS model (④) and those activated for export out from the model (⑤).

Although it is not used for this example, the checkbox (②) is available for Simulink users who want to run the vehicle models under full control of Simulink. In those cases, the MDL file is loaded outside the control of the VS browser and the current working directory is set according to how the Simulink environment is started. To support this type of operation, the box would be

checked and another data field is displayed to identify the working directory, where the VS solver will send data files needed by the VS S-Function block.

Import Variables

Click the blue link button (4) (Figure 5) to transfer to the **I/O Channels: Import** dataset for this example. As shown in Figure 6, this screen lists the available import variables in the model, and sets the number of import variables that will be handled by the vehicle block (e.g., the dimension 4 (6) shown in Figure 3).

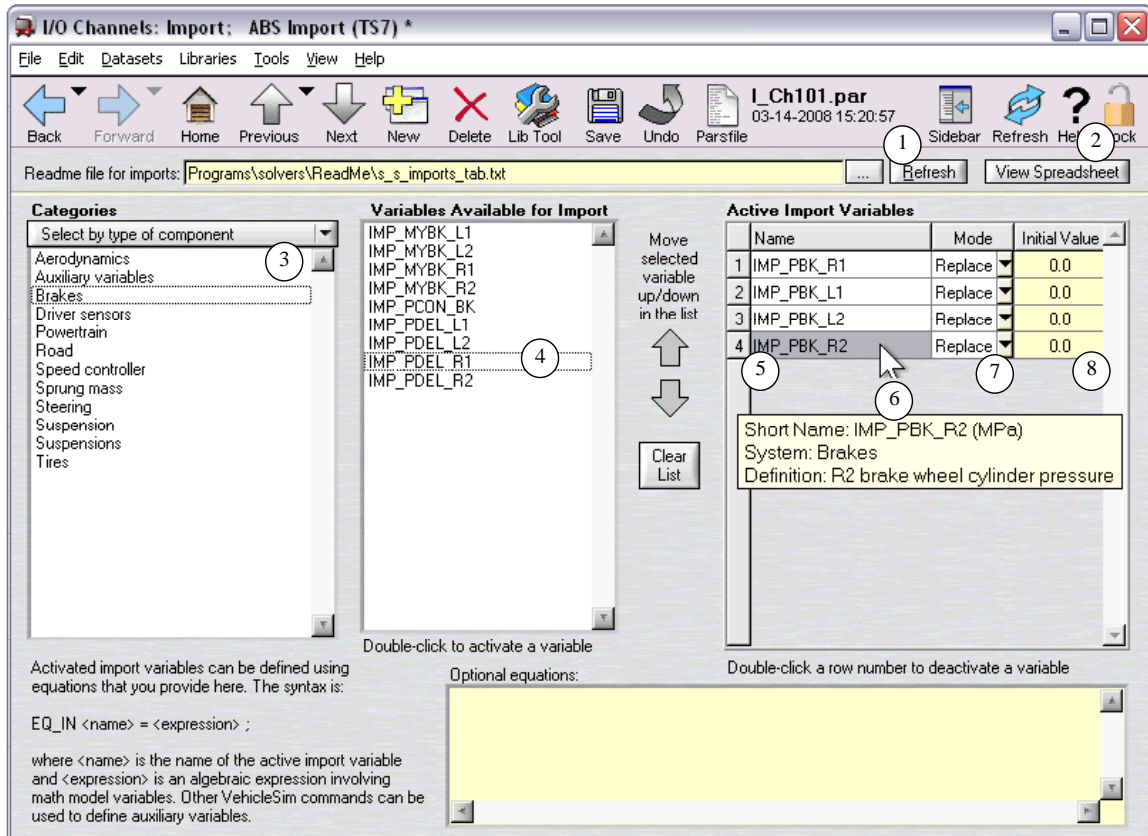


Figure 6. Activating variables for import into the vehicle S-Function block.

To browse through the available variables, unlock this screen and click the **Refresh** button (1). This makes the browser load a file that lists all of the variables available for import. You can also view all of the available variables in spreadsheet form by clicking the **View Spreadsheet** button (2) to load the same information into a spreadsheet program such as Microsoft Excel, as shown in Figure 7.

	A	B	C	D	E
37	IMP_DXEXT_7	1/s	Auxiliary variables	Time derivative of external state Xext_7	
38	IMP_DXEXT_8	1/s	Auxiliary variables	Time derivative of external state Xext_8	
39	IMP_DXEXT_9	1/s	Auxiliary variables	Time derivative of external state Xext_9	
40	IMP_MYBK_L1	N-m	Brakes	L1 brake moment	VARIABLE
41	IMP_MYBK_L2	N-m	Brakes	L2 brake moment	VARIABLE
42	IMP_MYBK_R1	N-m	Brakes	R1 brake moment	VARIABLE
43	IMP_MYBK_R2	N-m	Brakes	R2 brake moment	VARIABLE
44	IMP_PBK_L1	MPa	Brakes	L1 brake wheel cylinder pressure	VARIABLE
45	IMP_PBK_L2	MPa	Brakes	L2 brake wheel cylinder pressure	VARIABLE
46	IMP_PBK_R1	MPa	Brakes	R1 brake wheel cylinder pressure	VARIABLE
47	IMP_PBK_R2	MPa	Brakes	R2 brake wheel cylinder pressure	VARIABLE
48	IMP_PCON_BK	MPa	Brakes	Brake master cylinder pressure	VARIABLE
49	IMP_PDEL_L1	MPa	Brakes	L1 brake delivery pressure to ABS	VARIABLE
50	IMP_PDEL_L2	MPa	Brakes	L2 brake delivery pressure to ABS	VARIABLE
51	IMP_PDEL_R1	MPa	Brakes	R1 brake delivery pressure to ABS	VARIABLE
52	IMP_PDEL_R2	MPa	Brakes	R2 brake delivery pressure to ABS	VARIABLE
53	IMP_LX_SEN_1	m	Driver sensors	Longitudinal distance of driver sensor #1	VARIABLE
54	IMP_LX_SEN_2	m	Driver sensors	Longitudinal distance of driver sensor #2	VARIABLE
55	IMP_LX_SEN_3	m	Driver sensors	Longitudinal distance of driver sensor #3	VARIABLE
56	IMP_LX_SEN_4	m	Driver sensors	Longitudinal distance of driver sensor #4	VARIABLE
57	IMP_LX_SEN_5	m	Driver sensors	Longitudinal distance of driver sensor #5	VARIABLE
58	IMP_AV_D3_F	rpm	Powertrain	Output shaft speed of transfer case to 1st drive axle	

Figure 7. Spreadsheet showing all potential import variables.

When browsing through the available variables on the **I/O Channels: Import** screen (Figure 6), there are several categories that can be shown. The choice of category is made with the pull-down control (3). Variables in the current category that can be activated for import are shown in the middle list on the screen (4). A variable is activated (moved to the table (5)) by double clicking on it; an activated variable can be deactivated by double-clicking on the row number for the activated variable (5). Right clicking the mouse in the table of active variables brings up more information (6) about the currently selected variable (5). The same kind of information can also be obtained for a selected available variable in the middle list (4).

The table of active import variables configures the VS solver model in several ways with respect to how variables are imported and used.

1. The number of rows in the list is provided to the Simulink VS S-Function, which in turn provides Simulink with the dimension of the input signal. For this example, the dimension is 4.
2. The names of variables in the first column define which of the available import variables will be activated. Further, the order in which the variables appear in the table determines how the variables are indexed in Simulink. The row number shown in the table is the index for the input element in the S-Function block. For example, the variable named IMP_PBK_R2 (5) is element number 4 of the input signal in the Simulink model.
3. Many of the variables available for import can be set for one of three active modes, selectable with a pull-down control in the second column of the table (7). These modes are **Replace**, **Add**, and **Multiply**. If the mode is **Replace**, as shown for the four variables in Figure 6, then the internally defined values of the corresponding variables are replaced

with the values imported from Simulink. For this example, the hydraulic brake pressure for each actuator is replaced with a value provided by the Simulink model.


The Alternate option **Add** would add the imported value to the internal value, allowing a Simulink model to provide incremental changes to controls or other variables. The option **Multiply** uses the imported variable to scale the internally calculated value.

Note Not all variables available for import support all three modes. The modes for **Replace** and **Multiply** are only valid if there is a corresponding variable that is defined internally in the vehicle model. If there is no internal variable, then the only option available is **Add**. This was also the only option available in versions of the VS products prior to CarSim 7 (2007), TruckSim 7 (2008), and BikeSim 2 (2008).

4. When the VS solver program starts, it must set up initial conditions. However, variables are not yet available from Simulink. When used in the modes **Replace** or **Add**, all of the imported variables start with default initial values of 0.0 unless specified otherwise. When the option **Multiply** is specified, then the default initial value is 1.0. If you want to set alternate initial values, the yellow fields in the third column ⑧ can be used for this purpose.

The initial values are only used once during the simulation. After the run starts, the values provided from the Simulink model are used.

Export Variables

A similar screen is used to define the set of variables that are exported from the VS solver to Simulink. If you are currently viewing the **I/O Channels: Import** dataset shown in Figure 6, you can navigate to the **I/O Channels: Export** screen by first clicking the **Back** button  to return

to the **Models: Simulink** screen shown in Figure 5 (page 6), and then clicking on the blue link for the **Output Channels** ⑤ (Figure 5). You should now be viewing the **I/O Channels: Export** screen, shown in Figure 8.

The **I/O Channels: Export** screen is similar to the import screen. It is used to browse through the hundreds of variables calculated by the VS vehicle model and activate those that will be exported to Simulink from the S-Function block. To browse, unlock the dataset and click the **Refresh** button ① to use this screen, or click the **View Spreadsheet** button ② to view all of the variables from a spreadsheet.

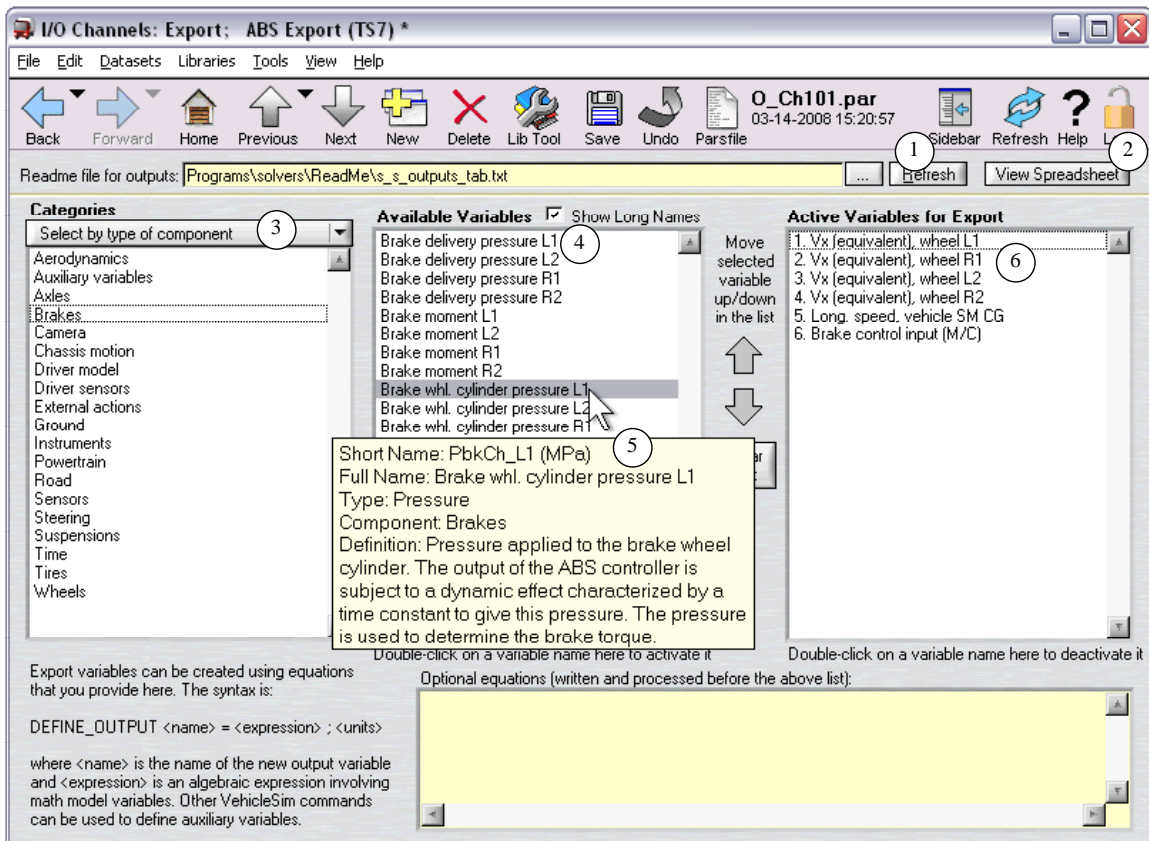


Figure 8. Activating variables for export out of the vehicle S-Function block.

The variables can be viewed by categories selected with a pull-down control (3), with the variables in the currently selected category (e.g., brakes) shown in the middle list (4). Right click in the middle list to view a detailed description of the currently selected variable (5). Double click on an available variable to activate it, moving it to the list of activated variable (6). Double click on an activated variable to deactivate it.

Because the variables available for export are also available for plotting, there is additional information for each variable that provides more options for browsing. For example, each variable has a short name (limited to 8 characters) and a full name (limited to 32 characters). You can view the variables using either name, choosing your option with the **Show Long Names** checkbox (4). Figure 9 shows the fields with the box unchecked for this example.

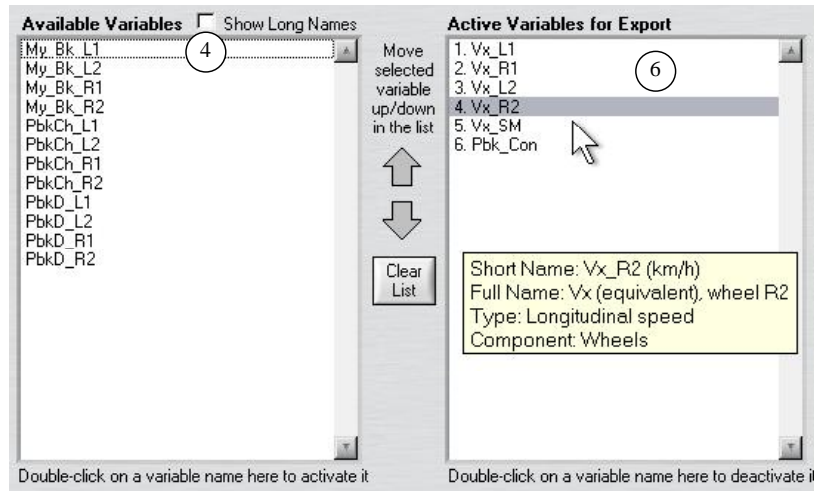


Figure 9. Display of output variables using short names.

The list of active export variables (6) configures the VS solver model in two ways with respect to how variables are exported.

1. The number of rows in the list is provided to the Simulink VS S-Function, which in turn provides Simulink with the dimension of the output signal. For this example, the dimension is 6 ((8) in Figure 3).
2. The names of variables in the list define which of the available output variables will be activated for export. Further, the order in which the variables appear in the list determines how the variables are indexed in Simulink. The row number shown in the list is the index for the output element in the S-Function block. For example, the variable named Vx_R2 is element number 4 of the output signal.

Putting a VS Vehicle Block into a Simulink Model

To add a VS block to a Simulink model, open the model in Simulink and open the Simulink Library Browser (see the Simulink **View** menu). Locate the S-Function block (e.g., TruckSim) and drag the block to your Simulink model as shown in Figure 10. Connect input and output signals as you would do with any block.

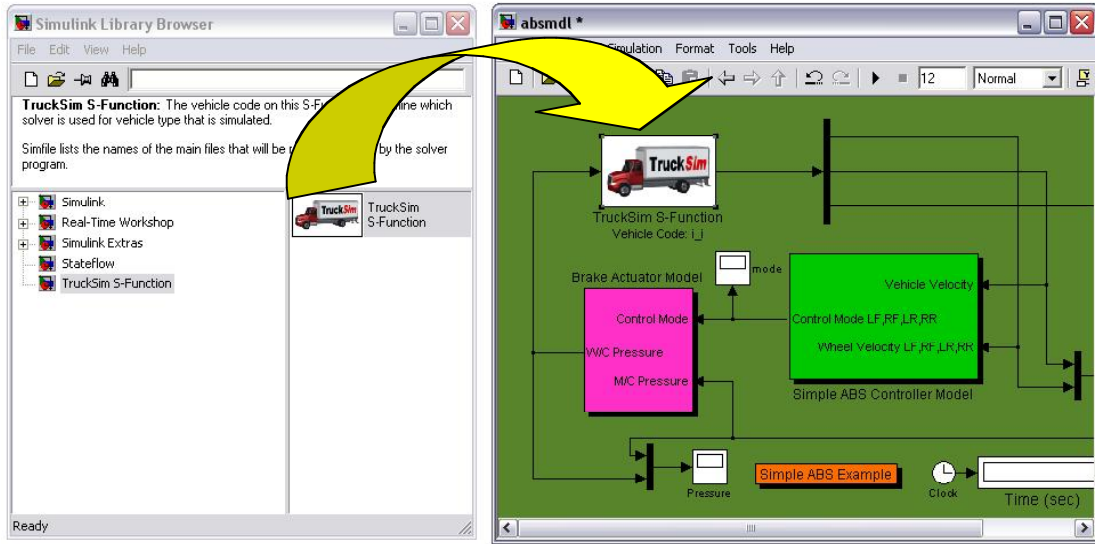


Figure 10. Drag and drop a VS S-Function block from the Simulink Library Browser.