

# Tracking Variables in a Simulink Model

By Parasar Kodati

Variables help us identify the parameters that determine model behavior. While it is easy to introduce variables into a model, it can be tedious to keep track of them. Variables with similar names are often defined in the MATLAB® base workspace, the model workspace, and the mask workspace of a subsystem and used in multiple places in a model. This makes it difficult to track usage of the variable and avoid redundancies.

The `Simulink.findVars` command makes it easy to find out whether a variable is being used in a given model, and where. With `Simulink.findVars` you can:

- Search a model for all places where a particular variable is used
- Identify unused variables in a workspace
- Avoid redundant usage by comparing workspaces of different models to find common and exclusive variables

## Where Is the Variable Defined?

Suppose that changes made to a variable `vehicledata` in the base workspace do not change the simulation results of your model. To find out why, you need to know whether the model is using the variable `vehicledata` and where it is defined (Figure 1). Using `Simulink.findVars`, simply input the name of the model and the variable that you're looking for. This generates the following output:

```
find_vehicledata = Simulink.findVars
('sf_car','Name','vehicledata')
find_vehicledata =

    Simulink.WorkspaceVar handle
    Package: Simulink
    Properties:
    Name: 'vehicledata'
    Workspace: 'sf_car'
    WorkspaceType: 'model'
    UsedByBlocks: {'sf_car/Vehicle'}
```

The Properties section of this output shows that `vehicledata` is indeed used by the model, more precisely by `sf_car/Vehicle`, and that the variable in use is in the model workspace (see the `WorkspaceType` property). The return object, `find_vehicledata`, is of type `Simulink.WorkspaceVar` and contains all the variables satisfying the arguments of the `Simulink.findVars` command. You

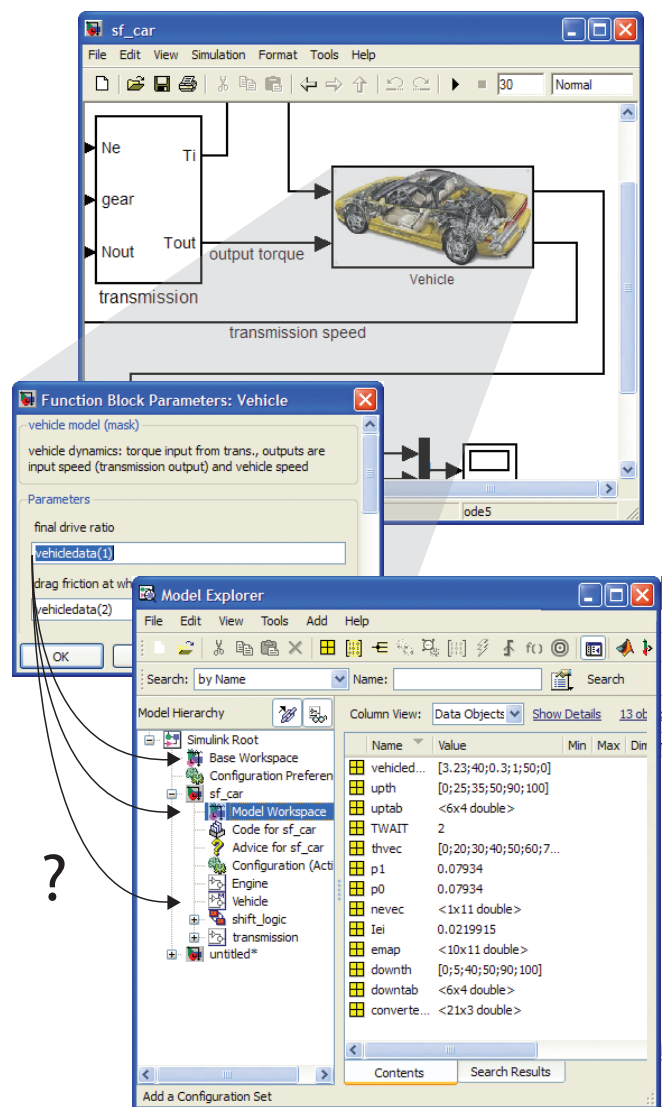


FIGURE 1. Where is `vehicledata`—in the base workspace, the model workspace, or the mask workspace?

can also find groups of variables in a model using properties like `WorkspaceType` and `UsedByBlocks`.

## Which Variables Do I Need?

Say you have many variables in your MATLAB base workspace and you want to know which ones are required for simulating a model and which you can clear. You do this by using methods like `intersect` and `setdiff` on `Simulink.WorkspaceVar` objects returned by `Simulink.findVars`.

First, get a `Simulink.WorkspaceVar` object for variables used by the model and defined in the base workspace.

```
f14_base_vars = Simulink.findVars('f14', ...  
'WorkspaceType', 'base');
```

Next, get a similar object for all the variables defined in the base workspace. Because no model is involved in this step, you cannot use `Simulink.findVars`. Instead, you use the `Simulink.WorkspaceVar` command:

```
all_base_vars = Simulink.WorkspaceVar(who, ...  
'base workspace')
```

Now the base variables not used by `f14` are those in the `all_base_vars` set but not in the `f14_base_vars` set. To get this subset of base variables, you use `setdiff` as

```
unused_base_vars = setdiff(all_base_vars, f14_base_vars);
```

To delete this subset of unused base variables, you can use the `clearvars`:

```
clearvars(unused_base_vars(:).Name);
```

Similarly, you can use the `intersect` method to find common variables in two `Simulink.WorkspaceVar` objects.

`Simulink.findVars` lets you efficiently find and update variables used by your model and remove unnecessary data from your workspaces.

## Tracking Variables via the Model Explorer Interface

The Model Explorer interface provides another way to search for variables used by the model. You select the search option “for Variable Usage” and specify workspace and other properties to find where in the model the variables are used (Figure 2). The contents pane lists all

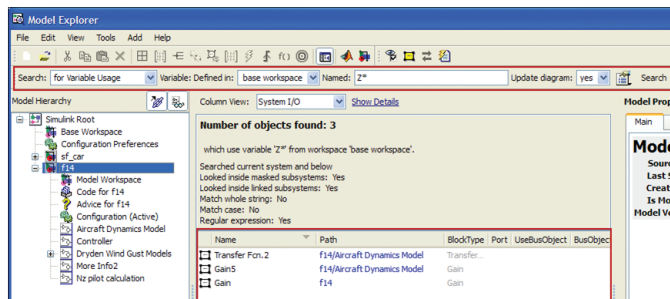


FIGURE 2. Sample search in the Model Explorer interface.

blocks that use the variable. You can use a wildcard (\*) in the variable name field to search for the usage of a group of variables.

## Diagram Update or Cached Results

Whether you are working at the command line or in Model Explorer, you have the option of using results cached by a previous compilation instead of updating the model diagram every time you search for variables. This option is especially useful if you have a large model and you know that its variable usage has not changed.

You specify this option when using `Simulink.findVars` as follows:

```
find_vehicledata = Simulink.findVars ...  
( 'sf_car', 'Name', 'vehicledata', 'SearchMethod', 'cached');
```

To use cached results when searching for variables in Model Explorer, you set the Update diagram option to NO. ■

## Learn More Online

**BLOG: Seth on Simulink®**  
[blogs.mathworks.com/seth](https://blogs.mathworks.com/seth)

**DEMO: Discovering Relationships Between Simulink Models and Workspace Variables**  
[www.mathworks.com/demos/simulink\\_workspace](https://www.mathworks.com/demos/simulink_workspace)