

# Übersicht über Variablen in Simulink-Modellen

Von Parasar Kodati

Variablen helfen, diejenigen Parameter zu identifizieren, die das Verhalten eines Modells bestimmen. Variablen in einem Modell einzuführen ist einfach – schwieriger ist es da schon, den Überblick über sie zu behalten. Oft werden im MATLAB®-Workspace, dem Modell-Workspace und dem Masken-Workspace eines Subsystems Variablen mit gleichem Namen definiert und dann an den verschiedensten Orten im Modell genutzt. Das macht es schwer, die Verwendung dieser Variablen im Blick zu behalten und Redundanzen zu vermeiden.

Der Befehl `Simulink.findVars` bietet eine einfache Möglichkeit herauszufinden, ob und wo eine Variable in einem bestimmten Modell benutzt wird. Mit `Simulink.findVars` kann man:

- Modelle nach allen Orten durchsuchen, an denen eine bestimmte Variable verwendet wird
- Ungenutzte Variablen in einem Workspace identifizieren
- Redundante Verwendungen vermeiden, indem die Workspaces verschiedener Modelle auf gemeinsame und einmalige Variablen untersucht werden

## Wo ist die Variable definiert?

Angenommen, man verändert die Variable `vehicledata` im Base-Workspace, aber die Simulationsergebnisse des Modells bleiben gleich. Um herauszufinden, warum das so ist, muss man wissen, ob das Modell die Variable `vehicledata` wirklich verwendet und wo diese definiert ist (Abb. 1). Dazu wird einfach `Simulink.findVars` mit dem Namen des Modells sowie der gesuchten Variablen ausgeführt. Man erhält das folgende Ergebnis:

```
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'}
```

Der Abschnitt `Properties` dieses Ergebnisses zeigt, dass `vehicledata` tatsächlich vom Modell verwendet wird, genauer vom Block `sf_car/Vehicle`. Aus der Eigenschaft `WorkspaceType` geht

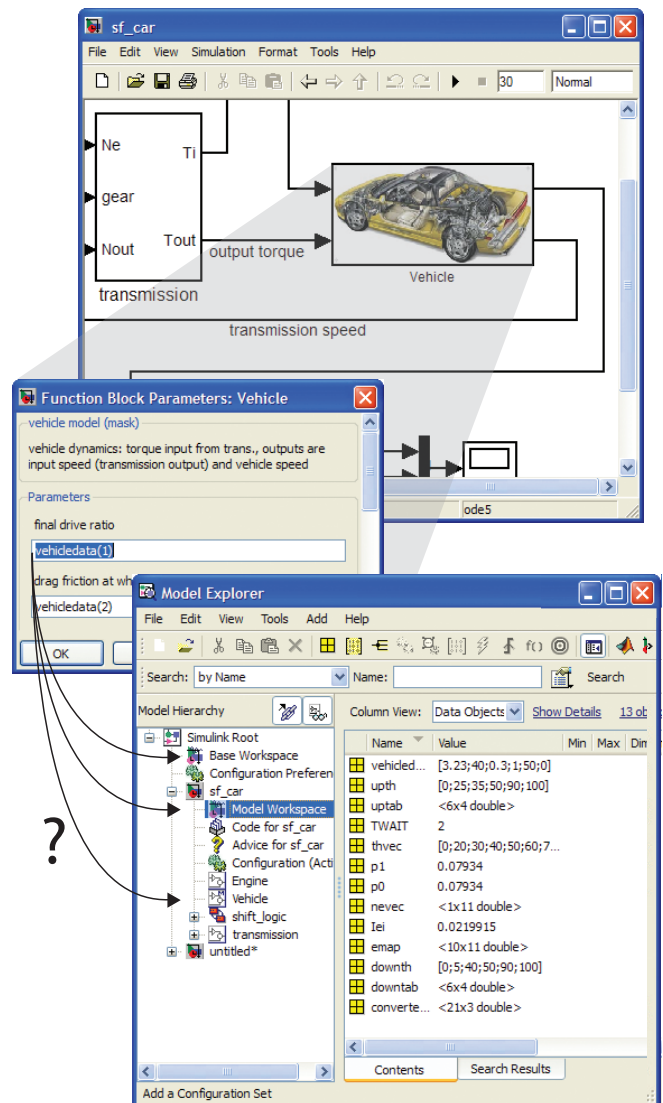


ABB. 1. Wo befindet sich `vehicledata` – im Base-Workspace, dem Modell-Workspace oder dem Masken-Workspace?

zudem hervor, dass sie sich im Modell-Workspace befindet. Das zurückgegebene Objekt, `find_vehicledata`, ist vom Typ `Simulink.WorkspaceVar` und führt alle Variablen auf, die die Argumente des Befehls `Simulink.findVars` erfüllen. Über Eigenschaften wie `WorkspaceType` und `UsedByBlocks` kann man außerdem auch Gruppen von Variablen in einem Modell finden.

### Welche Variablen werden benötigt?

Im MATLAB-Workspace befinden sich oft viele Variablen und man möchte wissen, welche zur Simulation eines Modells erforderlich sind und welche man löschen kann. Ermöglicht wird dies durch Anwendung der Methoden `intersect` und `setdiff` auf `Simulink.WorkspaceVar` Objekte, die von `Simulink.findVars` ausgegeben werden.

Zunächst erzeugt man ein `Simulink.WorkspaceVar` Objekt für Variablen im Base-Workspace, die vom Modell `f14` verwendet werden.

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

Das gleiche wiederholt man für alle im Base-Workspace definierten Variablen. Weil an diesem Schritt kein Modell beteiligt ist, funktioniert dies jedoch nicht mit `Simulink.findVars`. Stattdessen verwendet man den Befehl `Simulink.WorkspaceVar`:

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

Die von `f14` nicht verwendeten Variablen sind nun diejenigen, die sich zwar in der Menge `all_base_vars`, nicht aber in der Menge `f14_base_vars` befinden. Diese Untermenge isoliert man mit `setdiff`:

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

Schließlich löscht man die unbenutzten Variablen mit dem Befehl `clearvars`:

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

Auf die gleiche Weise kann man mit der Methode `intersect` gemeinsame Variablen in `Simulink.WorkspaceVar` Objekten finden.

Mit `Simulink.findVars` lassen sich also auf effiziente Weise die vom Modell verwendeten Variablen finden und aktualisieren sowie überflüssige Daten aus dem Workspace entfernen.

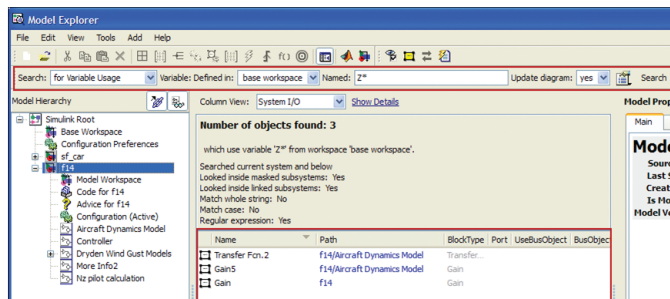


ABB. 2. Beispielsuche im Model Explorer.

### Verfolgung von Variablen im Model Explorer

Auch mit dem Model Explorer kann man herausfinden, ob und wo Variablen in einem Modell verwendet werden. Man wählt dazu die Suchoption "for Variable Usage" und legt dort den Workspace und andere Sucheigenschaften (Abb. 2) fest. Im Contents-Fenster werden daraufhin alle Blöcke aufgelistet, die die Variable verwenden. Gruppen von Variablen kann man suchen, indem man im Feld für den Variablennamen eine Wildcard (\*) einsetzt.

### Aktualisierte oder im Cache abgelegte Daten

Sowohl über die Kommandozeile als auch im Model Explorer besteht die Möglichkeit, im Cache gespeicherte Ergebnisse abzurufen, anstatt das Modell bei jeder Variablensuche zu aktualisieren. Diese Option ist vor allem dann nützlich, wenn das Modell sehr groß ist und man weiß, dass sich die Verwendung der Variablen nicht geändert hat.

Mit `Simulink.findVars` wird diese Option wie folgt definiert:

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

Im Model Explorer setzt man zur Suche nach zwischengespeicherten Ergebnissen einfach die Option „Update Diagram“ auf NO. ■

### Weitere Informationen online

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

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