TSFS02 — Vehicle Dynamics and Control

Dymola – A Short User Guide Introduction

Division of Vehicular Systems
Department of Electrical Engineering
Linköping University
SE-581 33 Linköping, Sweden

Contents

1 Starting Dymola

To start Dymola, open up a terminal window (e.g., by right-clicking the Desktop and *Open in Terminal*) and write the following commands:

```
module add dymola
dymola
```

If it is the first time you start Dymola, you will have to set up the license server options. Go to *Help* in the menu bar, and select *License...* Go to the *Setup* tab and fill in the server name and port provided by a course assistant.

1.1 Load Libraries and Models

To load any of the available model libraries, go to *File - Libraries* in the menu bar and choose the library to load. The libraries used in TSFS02 is *Modelon* and *VehicleDyamics*. (Note that if a model refers to a library, this library will load automatically.)

To open a Dymola model (e.g., TSFS02_Lab1_Braking.mo), go to *File - Open* or use the shortcut button in the tool bar.

2 GUI Overview

When you have started Dymola (and opened a model) you should have a window similar to Figure 1. This is the *Modeling* mode, in which all modeling, setting up experiments, etc., takes place. In Figure 2 the *Simulation* mode is shown, from which simulations are started and results can be analyzed. You can switch between these two views with the *Modeling* and *Simulation* buttons in the window's lower right corner (or from *Window - Mode* in the menu bar).

2.1 Modeling Mode

In the *Modeling* mode view you will see two browser windows on the left; the *Package Browser* and the *Component Browser*. In the *Package Browser* all the loaded packages (models and libraries) are shown and can be explored. Use this when navigating to the different sub-models and experiments in the exercises. In the *Component Browser* you can explore all the components included in a specific model, however, this will not be necessary to use in the exercises.

The main window is showing the current open model, which can be displayed by different "views". Switch between these views through *Window - View* from the menu bar, or from the shortcuts in the tool bar. The two most useful views are *Diagram* (shown in Figure 1) and *Modelica Text*, which also are the only two you will need to use for the exercises. In the *Modelica Text* view, parts of the text (annotations, components, and connections) can be hidden. To switch between these views, right-click in the text window and choose *Expand* (you can also switch with the shortcuts Ctrl+1, 2, or 3).

2.2 Simulation Mode

In the *Simulation* mode view, Figure 2, the *Variable Browser* is seen on the left and the *Commmands* window on the bottom. The latter will not be necessary to use in the exercises. In the *Variable Browser*, results from the simulated experiments will show up (more about this in Section 4).

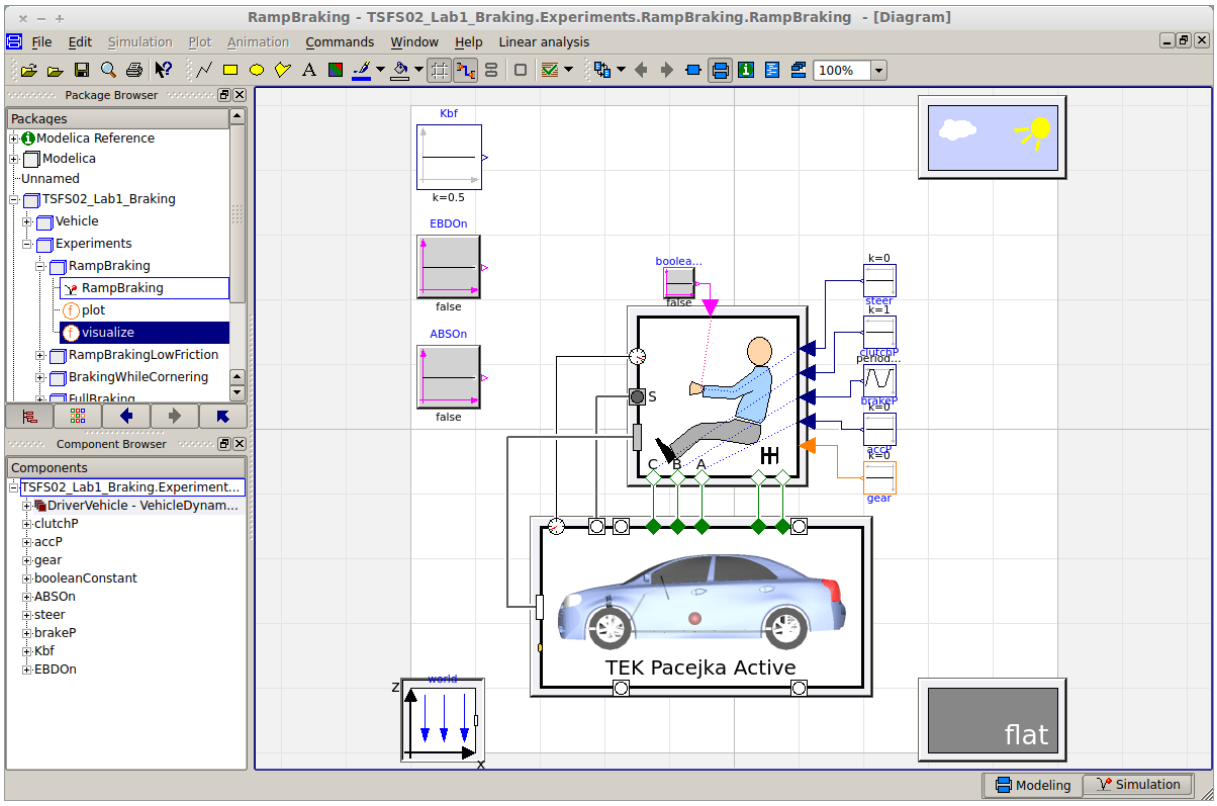

Figure 1 Modeling mode view in Dymola.

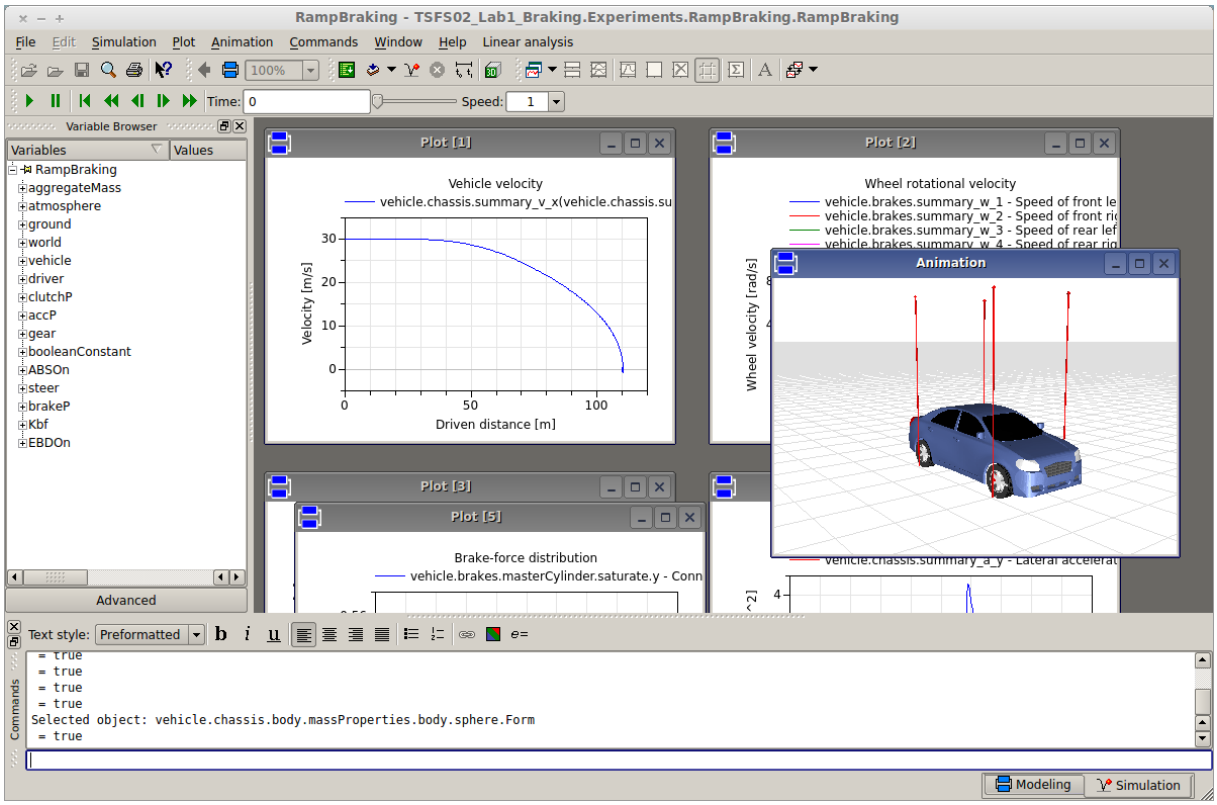

Figure 2 Simulation mode view in Dymola.

3 Modeling

Building up models and experiments in Dymola can be done both from the graphical interface (*Diagram* view), being a bit similar to Simulink, and from a text-code interface (*Modelica Text* view), which is more of a "traditional" coding environment. All changes in the *Diagram* view will show up in the *Modelica Text* view, but not necessarily the other way around. You will mostly be working from the *Modelica Text* view when implementing your own models, and from the *Diagram* view when running prepared experiments.

3.1 Modelica Code

A simple Modelica model (when shown in *Modelica Text* view) is structured as below.

```
model ModelName
  input u1 "Description of u1 (not necessary)";
  input u2;
  output y;
  parameter Real k1;
  parameter Real k2;
  Real x1;
  Real x2;
equation
  y = k1*x1 + x2;
  x1 = u1;
  x2 + u2 = k2;
  // Comment a row
  /* Comment
  multiple
  rows */
end ModelName;
```

Above `equation`, all inputs (`u1`, `u2`), outputs (`y`), parameters (`k1`, `k2`), and variables (`x1`, `x2`) should be stated. Below `equation`, all equations and calculations are performed. Note that the equations do not have to be in sequential order, e.g., the equation for `y` can be written ahead of the equations for `x1` and `x2`. Neither does the equations have to be formulated to compute a specific variable, but rather just specify a relation between variables and parameter, e.g., it does not matter on which side of the equal sign the variables/parameters are.

if Statements

A general if statement can be formulated as following.

```
if a>1 and b>1 then
  c = 1;
elseif a==-1 or b==0 then
  c = 2;
else
  c = 3;
end if;
```

Derivatives and Integrals

Integrating and differentiating a variable is basically done in the same way. The time derivative of the variable `a` can be written as `der(a)`. For example, the expression $b = \frac{\mathrm{d}a}{\mathrm{d}t}$ is in Modelica code formulated as:

```
b = der(a);
```

To integrate a variable, the same method can be used. If `c` is the time integral of `d`, i.e., $c = \int d\mathrm{d}t$, then $\frac{\mathrm{d}c}{\mathrm{d}t} = d$ also yields, and we can formulate the expression as `der(c) = d`. However, we also need to specify the start value of `c` in the beginning where we define the variable. Below is a Modelica example where we set the start value of `c` to zero, and then integrate `c`.

```
model ModelName
  input d;
  Real c(start=0, fixed=true);
equation
  der(c) = d;
end ModelName;
```

4 Simulation and Results

When you are about to run a simulation, navigate to the experiment you intend to run in the *Package Browser* and switch over to the *Simulation* mode view. Run the simulation by choosing *Simulation - Simulate* in the menu bar, or pressing the *Simulate* shortcut in the tool bar (the icon with a red bouncing ball). A message window will show up, displaying information, warning, and error messages. A few warning messages will generally pop up, but don't worry about them here. If any error messages show up, something is wrong!

After the simulation is complete, the results will show up in the *Variable Browser*. Per default, results from the last two or three simulations are stored in the *Variable Browser*, while the rest will be discarded (if not saved). You can change this by opening *Plot - Setup...* from the menu bar, go to *Options* and change *Number of results to keep*. (Remember that keeping a lot of results in the workspace will eat up the memory.)

4.1 Save Results

Results can be saved by right-clicking it in the *Variable Browser* and selecting the *Export Result As...* option. Previously saved results can be loaded from *Plot - Open Result...* in the menu bar.

4.2 Build Road Commands

For some of the experiments you need to build up a mathematical description of the road prior to running the experiment. When this is necessary, a function called *buildRoad* is placed in the same experiment package. You only have to run this function once, before you simulate the experiment for the first time.

4.3 Plot Result

Most of the exercises in the course already have prepared plot functions for each experiment. These can be found together with the experiment model in the *Package Browser* (in the *Modeling* mode view), and are always named *plot*. Execute the function by right-clicking it, choose *Call Function...*, and then *OK* in the dialog box that appears.

To plot other variables, open up a new plot window (*Plot - New Window - New Plot Window* in the menu bar, or *New Plot Window* from the tool bar), then navigate to the variable of interest in the *Variable Browser* and click it.

4.4 Visualize Result (Animate)

Prepared functions for visualizations are also available in the experiment packages. These are named *visualize* and executed in the same way as the *plot* functions. A window called *Animation* will then pop up (or be updated) among the plot windows in the *Simulation* mode view, and an animation of the results

can be run with the green play button (in the tool bar).

In the animation window you can change the "camera view" in different ways. Left-click and drag for horizontal and vertical movement. Ctrl + left-click and drag to change the camera angle. Scroll, or Shift + left-click drag, for zoom.

By default (in the visualization functions) the camera will follow the selected component, which is set to the vehicle's center-of-gravity in the visualization function. By clicking any body element in the animation window, you will select it (it will highlight red), and the camera will follow that element instead.

You can animate different results together, which can be a nice way of illustrating and comparing different vehicle setups. Make sure the animation window is active, then right-click on the result you want to add and choose *Animate Together*.