How to use BevModels.mo

1 General

The folder where you read this document contains file BevModels.mo, plus some additional data files having “txt” as extension.

BEVModels.mo is a very simple archive containing two models that have been used to reproduce part of the results of the paper *Battery electric vehicles: how many gears? A technical-economic analysis* Submitted for publication on MDPI’s Vehicles. In the reminder of this document, this paper will be referred to as just *the paper*.

These models are very small, since it takes advantage of the modelica library EHPTlib, open source and available from https://github.com/ceraolo/EHPTlib

The usage of all models of this library, with a lot of examples are explained in detail in the open source modelica library available from https://github.com/ceraolo/EHPTexamples

This library contains a full-fledged 78-page tutorial (file Tutorial 2.1.1 Jan 2023.pdf)

2 Prerequisites

The models in this archive are very simple and can be used directly from the graphical user interface, through diagrams. As all modelica models, however, it can be used also using their textual view and writing modelica code.

Prerequisites to run these models are:

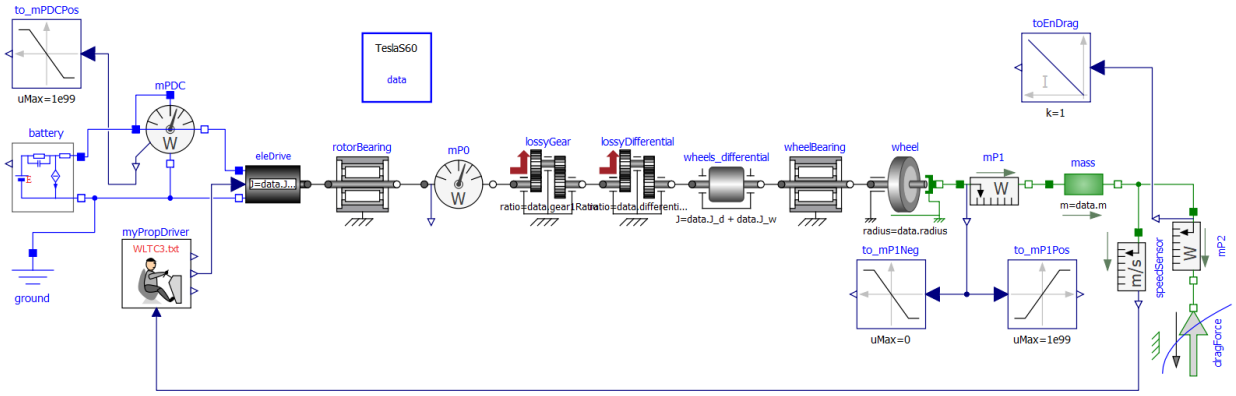
- Some basic knowledge of Modelica language and models.
  An introduction to Modelica can be found in www.openmodelica,org, accessing one of the two YouTube videos available on a single click from the bottom of the page.
- The availability of modelica-capable simulation tool. Many tools are commercially available. However, the proposed models work very well with the totally free, open source modelica simulation tool OpenModelica, available from www.openmodelica.org

3 Usage

The usage of the proposed models is very similar to the one of models in the archive EHPTexamples.EV (from the already mentioned EHPTexamples github repository), whose usage is explained in detail in its already-mentioned tutorial.

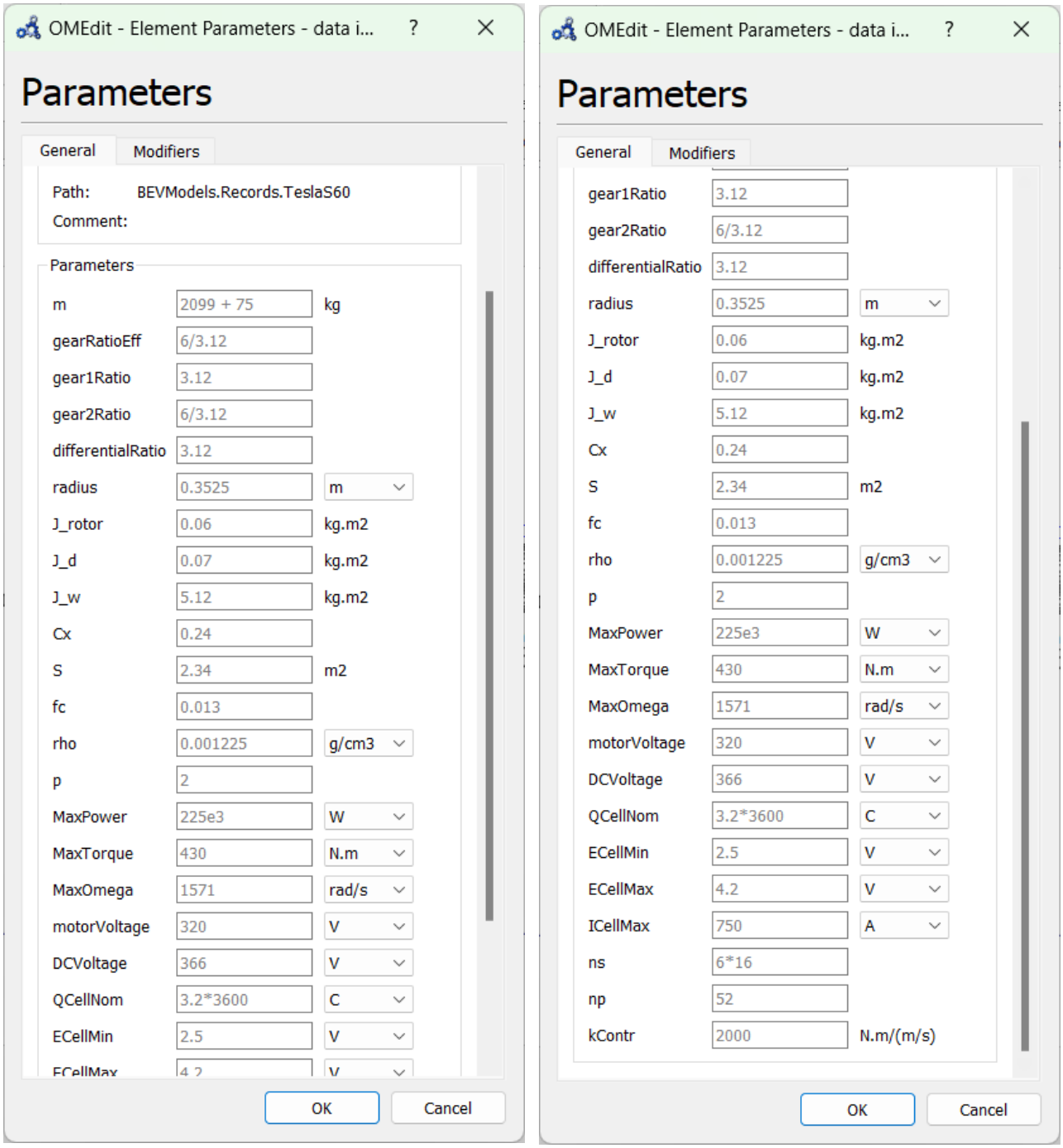
Here some short information is proposed, which takes as reference exactly the two models inserted in this GitHub archive.

The model named Tesla and Nissan in the archive, represent the model named “T” and “N”, respectively in the paper. To fix ideas, consider the model Tesla, whose diagram, is as follows:

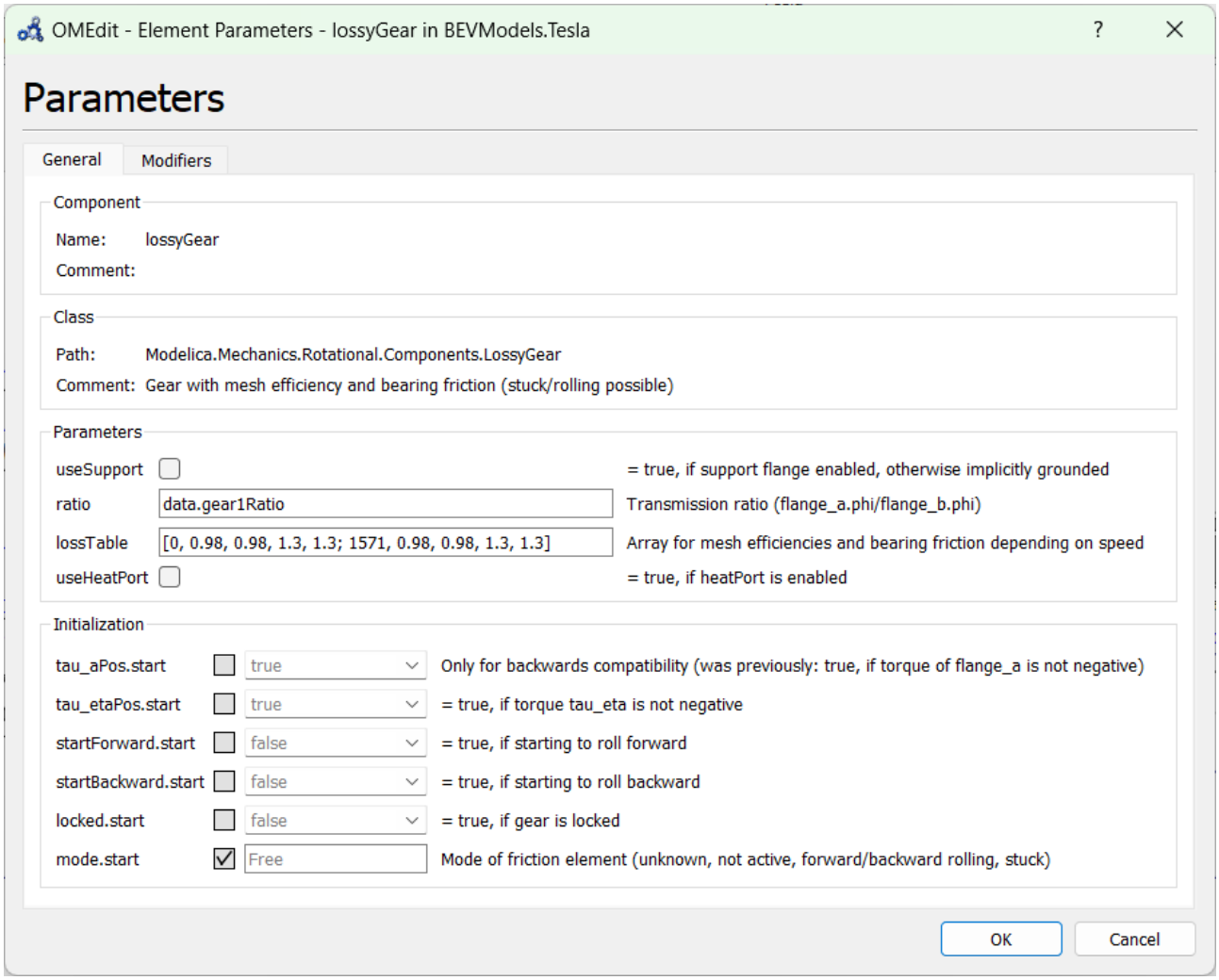


Note that even though this diagram has been created using OpenModelica, a virtually identical one is visible with any of the modelica-capable tools. This because Modelica language contains graphical directives, that make the very appearance of the model very similar across simulation tools.

To ease modifications and avoid common mistakes, many parameters are grouped in the block outlined in blue “data”. If users double click on it, they get the following dialog box:



This box contains all the single value parameters of the model. Other parameters are expressed as matrixes, and they are directly inserted in the involved model. Consider for instance model lossyGear. If we doubleclick on it we get:



So gear ratio is taken from the box data: data.gearRatio, while the lossy characteristics, expressed as a matrix ("lossTable") are directly inserted here. The meaning of this matrix is easily accessible as the "info", of the submodel lossyGear.

Finally, there are some data, such as efficiency maps, that is inconvenient to write manually in the model. Better is to put them in external files linked to the model.

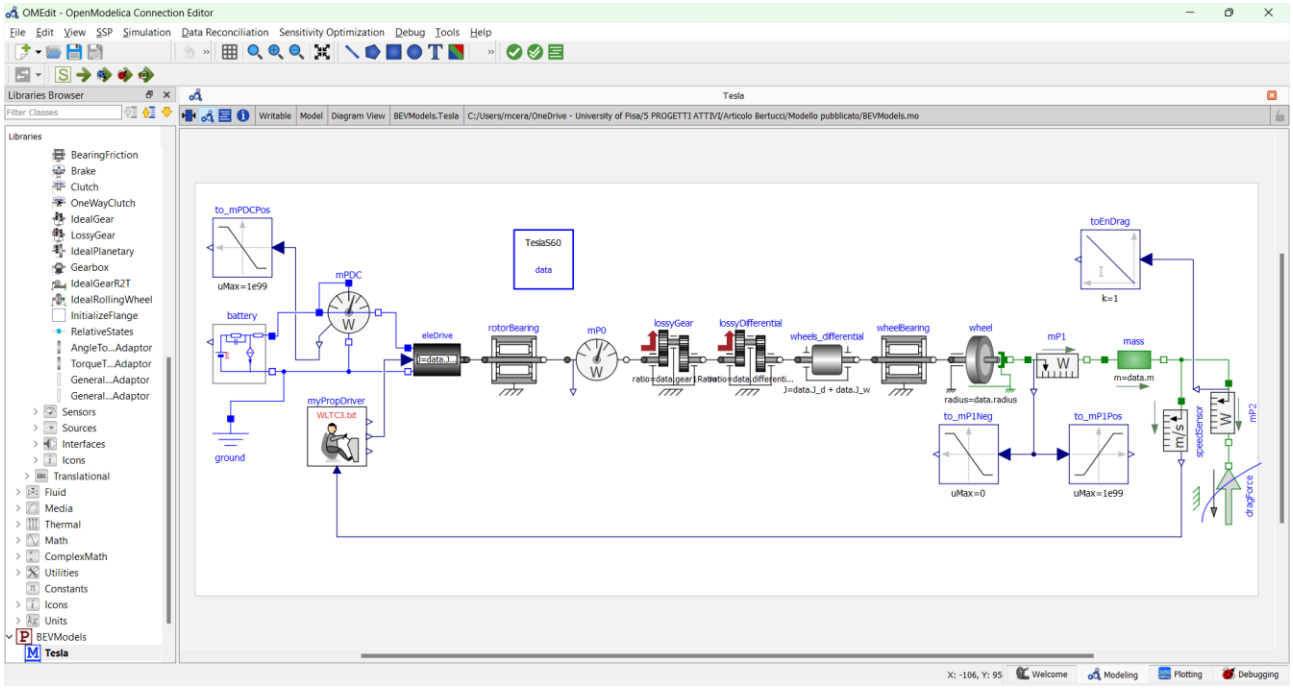
We have two such files.

- File "WLTC3.txt", containing the standard testing cycle WLTC, linked to driver mentioning its name in the drover's dialog box (can be open double-clicking on the driver's icon) and putting the file "WLTC3.txt" in the working directory of the simulation tool.
- "TeslaMaps,txt" and NissanMaps.txt" containing the efficiency maps of the two considered vehicles.

To facilitate users' activities, the directory contains also the NEDC cycle, in a file named "NEDC.TXT".

Running simulations and plotting results depend on the chosen simulation tool.

For instance, the OpenModelica appearance with Tesla model is as follows:



To run the model, one just clicks on the tool “simulate” on the toolbar (or choose from the menu Simulation “Simulate”). After a few seconds (around 20-30 seconds on a common desktop PC) the plotting perspective appears where one can plot results. An example is as follows: