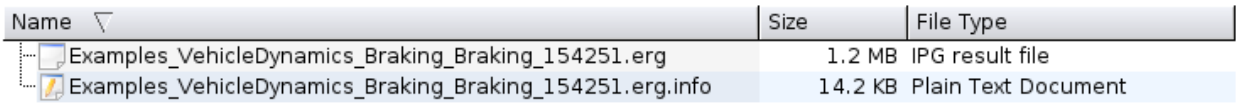
Date:	2017-07-20
Author:	Sophie Schäferle, CarMaker Service Team Germany
Release No.:	<= CarMaker 6.x (example for CarMaker 6.x)

How can I use Matlab for postprocessing of CarMaker simulation results?

Sometimes I speak to CarMaker users that have managed to model very thorough, complex TestRuns, but aren't sure how to go from there. Successfully completing a variety of virtual experiments may be a crucial element of today's development process, but not very effective if the produced data isn't evaluated.
There are several methods for evaluating simulation results. Along with using IPGControl or Concerto, CarMaker offers an excellent interface to MATLAB with which data can be exported to the MATLAB workspace. Since MATLAB is a very popular option, I wanted to show you how to upload your data using an example.

So first I need a fully parametrized TestRun that I can simulate and generate results with. I'm going to use the CarMaker example TestRun “Braking”, which can be found in the product examples via *Examples/VehicleDynamics/Braking/Braking.* In this example I want to save the data of the entire TestRun, so in the field labelled “Storage of Results”, I'm going to set the mode to “Save all”. If you want to know how the other storage modes work, you can have a look in the User's Guide.

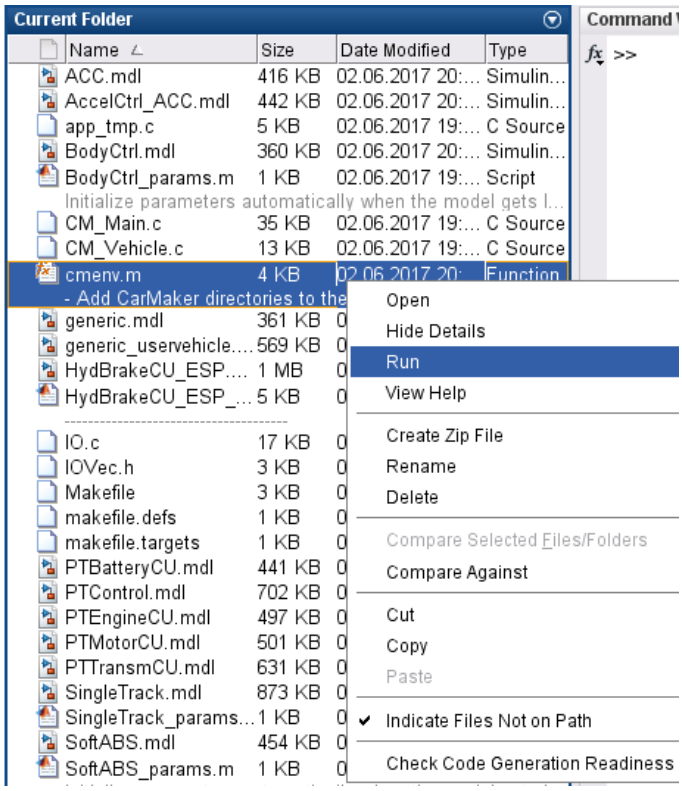
Now I can run the simulation! After the simulation has come to an end, I can go to the “SimOutput” folder of my project directory *<ProjectFolder>/SimOutput/<ClientID>/<Date>* and I'll see that a result file with an “.erg” ending and its corresponding infofile have been generated.



So as off now, I'm going to work with MATLAB.

To find out which versions of MATLAB are compatible with your current CarMaker version, just check the Release Notes.

After opening MATLAB, I need to make sure that my working directory is the “src_cm4sl” folder of my project directory. In this folder, there's a file called “cmenv.m” that you need to run. After it's run successfully, the MATLAB search path is configured and CarMaker is connected.



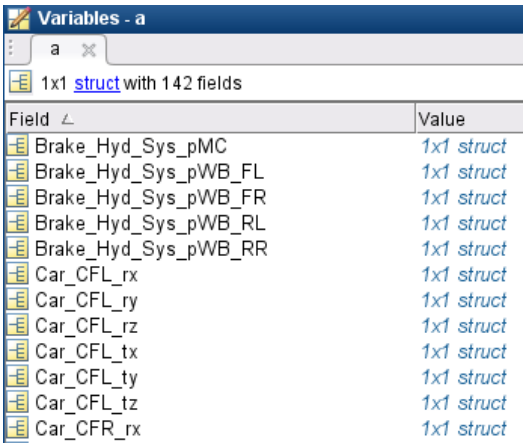
Now it's time to upload the result file to the MATLAB workspace. First, select a variable name that will represent your data - I'm going to use “a”. Using the following command, I'm going to assign my result file to my workspace variable “a” using the “cmread” command:

```
1:  a=cmread('../SimOutput/<ClientID>/<Date>/<ResultFileName>.erg')
```

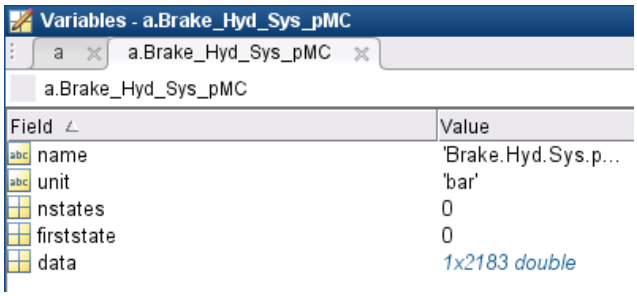
After */SimOutput/* be sure to specify the correct path to your result file and don't forget to include the file ending. In case you aren't sure what the absolute path is, you can also just type `a = cmread`, press “enter” and a browser will open within which you can navigate to your result file.

In the command window you will see that a list with all of the output quantities appears and in the workspace, on the right side, your variable has been assigned a matrix.

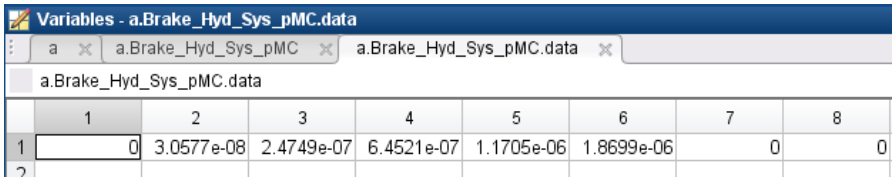
Clicking on the variable (in this case “a”) shows you the interior of the struct. This is a list of all the output quantities that you could also see in the command window.



When you click on one specific quantity, there are several fields: name, unit, nstates (number of different states that the quantity can have), firststate (start value of the quantity) and data.



To see the exact values of a quantity, select “data”.



Now that you've loaded the results into MATLAB, there are endless ways of working with them. Here are just two examples:

- Displays the contents of a quantity

```
1:  disp(<Variable>.<QuantityName>)
```

- Plots two quantities against one another

```
2:  plot(<Variable>.<QuantityName>.data , <Variable>.<QuantityName>.data):
```

You can find further information regarding this topic either in the CarMaker User's Guide or online within the MATLAB help page.

I hope you learned something out of today’s entry and feel confident in using MATLAB for postprocessing!