

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.

Name 🗸	Size	File Type
Examples_VehicleDynamics_Braking_Braking_154251.erg	1.2 MB	IPG result file
	14.2 KB	Plain Text Document

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.



Seite 2

urrent Folder				\odot	Command
🗋 Name 🛆	Size	Date	e Modified	Туре	$f_{\underline{x}} >>$
🐴 ACC.mdl	416 KB	02.0	6.2017 20:	Simulin	
🖥 AccelCtrl_ACC.mdl	442 KB	02.0	6.2017 20:	Simulin	
app_tmp.c	5 KB	02.0	6.2017 19:	C Source	
🛓 BodyCtrl.mdl	360 KB	02.0	6.2017 20:	Simulin	
BodyCtrl_params.m	1 KB	02.0	6.2017 19:	Script	
Initialize parameters a	automatica	ally w	hen the mod	el gets I	
CIVI_IVIAIN.C	12 MB	02.0	6.2017 19: 6 2017 19:	C Source	
	1 IZP	b2.0	6.2017 19 6.2017 20:	C Source	
- Add CarMaker direct	tories to tl	ne 1e	Onen		
generic.mdl	361 KB	0	Uide Deteik		
generic_uservehicle	. 569 KB	0		5	
🛿 HydBrakeCU_ESP	1 MB	0	Run		
HydBrakeCU_ESP	5 KB	0	View Help		
 1 ו0 ר	17 KB	0	Create Zip F	ile	
IOVec.h	3 KB	ŏ	Rename		
Makefile	3 KB	o	Delete		
] makefile.defs	1 KB	d —	Belete		
🗋 makefile.targets	1 KB 441 KB	0	Compare S	elected <u>F</u> ile	s/Folders
PTBatteryCU.mdl		0	Compare A	gainst	
PTControl.mdl	702 KB	g —	0.ut		
PTEngineCU.mdl	497 KB	g	Cui		
BTT:::::::::::::::::::::::::::::::::::	501 KB	9	Сору		
	072 I/D	3	Paste		
SingleTrack.mol	1 KB	1	Indianta Fila	a Nation Da	
SoftABS mdl	454 KB	ď –	indicate File	IS NOLON Pa	auri
SoftABS params m	1 KB	ď	Check Code	e Generatio	n Readiness
paramo	· · · · · · · · · · · · · · · · · · ·				

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 <code>a = cmread</code>, 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.

Workspace		
Name ∠	Value	
🔁 a	1x1 struct	

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.



Seite 3

🔏 Variables - a	
: a x	
📃 1x1 struct with 142 fields	
Field ∠	Value
📧 Brake_Hyd_Sys_pMC	1x1 struct
📧 Brake_Hyd_Sys_pWB_FL	1x1 struct
📧 Brake_Hyd_Sys_pWB_FR	1x1 struct
📧 Brake_Hyd_Sys_pWB_RL	1x1 struct
📧 Brake_Hyd_Sys_pWB_RR	1x1 struct
📧 Car_CFL_rx	1x1 struct
📧 Car_CFL_ry	1x1 struct
📧 Car_CFL_rz	1x1 struct
🔚 Car_CFL_tx	1x1 struct
📧 Car_CFL_ty	1x1 struct
🔚 Car_CFL_tz	1x1 struct
🔁 Car_CFR_rx	1x1 struct

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.

🔏 Variables - a.Brake_Hyd_Sys_pMC					
: a 🗙 a.Brake_Hyd_Sys_pMC 🔉	<				
a.Brake_Hyd_Sys_pMC					
Field ∠	Value				
🔤 name	'Brake.Hyd.Sys.p				
🔤 unit	'bar'				
Η nstates	0				
Η firststate	0				
금 data	1x2183 double				

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

1	Zvariables - a.Brake_Hyd_Sys_pMC.data							
:	a x a.Brake_Hyd_Sys_pMC x a.Brake_Hyd_Sys_pMC.data x							
	a.Brake_Hyd_Sys_pMC.data							
	1	2	3	4	5	6	7	8
1	0	3.0577e-08	2.4749e-07	6.4521e-07	1.1705e-06	1.8699e-06	0	0
2								

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!