

Date:	2019-06-17
Author:	Pooja Hegde, Test Systems and Engineering, Germany
Release No.:	CM-7.1.1

Data Files

The CarMaker_CANoe_CoSim.zip file following contains the files.

- 1) CarMaker Project file: Created using the CarMaker GUI containing the general folder structure
- 2) CANoe Files: Contains the config file to be used, the database file that is needed, CAPL script used for controller logic and the final resulting FMU that is exported

Connecting CarMaker with CANoe through FMU

This is an example to connect CarMaker with CANoe for a Rest Bus Simulation of a generic ACC system. The connection is established to communicate signals between CarMaker and CANoe when both the tools are open and simulating simultaneously.

Technical Background

FMUs are used to exchange models conveniently between different tools directly as black box (or grey box) models. This example is based on FMU standard 1.0. The aim here is to obtain input signals to the FMU from CarMaker, then use these signals in CANoe where the ACC function is implemented and finally to give the output from the FMU ack to CarMaker. The controller is a very basic open loop proportional gain speed control since the aim of the example is to establish the connection between the tools and not develop the function.

Note: The officially supported FMU mode by CarMaker is of the type Co-simulation Standalone (Refer <u>FMI for Co-Simulation</u>). The default mode of the FMU from CANoe is set at Co-simulation Tool. A workaround is used to manually set the FMU mode to standalone even when both tools are open and running and hence this particular example is not officially supported.

CarMaker version: 7.1.1 CANoe version: 11.0





Setting the Simulation Mode

- Go to folder C:\ProgramData\Vector\CANoe\<version> and open file CAN.ini in a text editor
- Change the value of *CSv1ForceStandalone* to 1. This will set the configuration to be of type Cosimulation standalone

```
[FMI]
; Default length of FMI model variables to be exported
DefaultExportStringLength=128
; implementation type of exported FMI v1 co-simulation FMU
; 0 - CoSimulation_Tool (default)
; 1 - CoSimulation_StandAlone
CSv1ForceStandalone=1
```

Setup: CANoe Components

All the required settings are in the Configuration file that is in this example. Below is a brief explanation of the components.

- Node: Nodes are added to the CAN network to simulate different ECUs
- Dbc file: Database file contains the raw signals that are used for data transfer
 - Drag and drop the dbc file from a folder into CANoe GUI. It will be automatically added to the network



- A database file needs to be added to the Node to access different signals
 - o Right click on node and click on configure
 - o Select the dbc file along with the required network node
 - In the components tab add the dll files required by the CAPL script that should be linked to the node. The filepath for the dlls is C:\Program Files\Vector CANoe 11.0\Exec32



Seite 3

Common CANope	an Components Buses
Title:	ECU 1
Network node:	TrainingNetwork::CCU ~
State:	● simulated O off
Execution:	Standard
	Extended
Hint The execu function range o hardware featur Node specificatio	tion mode affects the real time behavior as well as the supported if the simulation node. Some execution modes require special es. For details, please refer to the online help. on File Edit Compile C\oe_CoSim\CANoe_files\CarMAker_CANoe_CoSim.can
	OK Cancel Help
nfiguration	
n CANopen C	omponents Buses
J (Type: Network I	Node):

N

- CAPL: Vector specific programming language that can be used to manipulate and calculate signals. This can be accessed by doing a right click on the node and selecting edit. Each node can have a separate CAPL script for different logics.
- System Variables: User defined variables to store and calculate data. Different datatypes can be assigned to the variables.
 - These variables can be either mapped to raw signals to store data or the user can initialize his/her own variables for internal usage in a script (as a trigger for Example)

ile Home Analysis	Simulation Test	Diagnostics E	nvironment	Hardware	Tools L
🦞 💊 🛁		35.25	N	(Çn	
mbol System Symbol olorer Variables Mapping	Start Communicati Values Setup •	on Compile D All Nodes	ocuments	Tool Couplings -	Licensing
Symbols			Mor	e	
User-Defined System-Def	ined Name Displa S 🔊 • 🔊 • 🥼	y 🍋 <search th="" va<=""><th>riable></th><th>• //</th><th>i 酸 ₽-</th></search>	riable>	• //	i 酸 ₽-
User-Defined System-Def Variable	fined Name Display	y <search va<br="">Initial Value</search>	riable> Min	• ∰ Max	Location
User-Defined System-Def	ined Name Displa S S • S • S •	y Search Va Initial Value	riable> Min	- M Max	Location
User-Defined System-Def Variable Q control Variable	ined Name Displa Name Displa Image: Second state Image: Double	y Search Va	riable> Min	- Max	Location Configurati
User-Defined System-Def Variable Q control Q sensordata	ined Name Display	y Search Va Initial Value	riable> Min -	- Max	Location Configurati
User-Defined System-Def Variable Q control Q sensordata - > input_brake	ined Name Display	y Search Va Initial Value	riable> Min -	- Max	Location Configurati
User-Defined System-Def Variable Q control Q sensordata Q sensordata Q sensordata Q sensordata	ined Name Display	y Search Va Initial Value	riable> Min - -	- H Max - - - -	Location Configurati



- Simulated bus: Select the Simulated bus option from the Home menu

File	Home	Analysis	Simulatio	n	Test	Diagnostics	
<u> </u>		🛼 Step	100	~	4	Online Mode	*
7	Church	🔍 Break	-		٩	Simulated Bus	*
start *	Stop	🕼 Animate				Standalone Mo	de
		Mea	surement				

FMU Export settings

- Go to Environment Tab-> Tool Couplings-> Functional Mockup Interface
 Settings:
 - Interface Version:1
 - FMU type: Co-Simulation
 - Stepsize: 1ms (or other depending on simulation and datatypes)
- Be sure to assign the correct Input/Output status to each signal/variable
- Click on the export symbol on the top-left corner to export the FMU

mbol Syste	m Symbol	Start C	ommunicati	ion (Compile	Document	ts Tool	Licensin
plorer Variab	les Mapping Symbols	Values	Setup -	A	ll Nodes		Couplings - More	
Functional Mock-up U port Export	nits (FMUs)							
Export settings Identifier: Description: Interface version: Exported FMU:	CANoeFMIv1	 FMU type Demo\Demo_pot 	Co-Simulat	ion n\CANoe.fm	S F S	itep size [s]: MI port: lecoupled time b reparate FDX file	0.00200 € 4660 € ase: ✓	
Variable / Signal Variable / Signal S				Category Input Input Input Output	Data Type Real Real Real	Byte Size 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8		M / Tx Node

CANoe signal mapping

- Data inside CANoe has to be mapped to appropriate channels
 - Raw Signal data/data stored in variables should be mapped from a source on to a destination.
 - Input_brake, speed and objectdistance are coming in from CarMaker. Let us consider input_brake
 - It is first written into the raw signal cno_sigBrake from the dbc file
 - Some calculations are performed and this data is modified
 - This modified data is stored in the variable brake



This is the signal that will be sent back to CarMaker as a control input
 Note that it is important that the data types of the mapped signals match for the simulation to run without errors.

Home	Analysis	Sim	ulation	Te	est	Diagnostics	Environment	Hardwa	are Tools
	-	E.	T)			33	1	Çn	1
ol System er Variables	Symbol Mapping	Sta Valu	rt Cor	nmun Setup	nicatio p =	on Compile All Nodes	Documents	Tool Couplings	Licensing
	Symbols						M	ore	
× =					- 4	ga 24 <u>5</u>			
≫ 🔤 Destination			Factor		- 4	source		Offset	Mapping
X		=	Factor 1	x	- 4	Source	+	Offset 0	Mapping OnChange
X Destination cno_sigBrake brake		=	Factor 1	x x	• 4 ~	Source input_brake cno_sigBrake	+	Offset 0	Mapping OnChange OnChange
Destination cno_sigBrake brake cno_sigVehSp	eed	=	Factor 1 1	x x x	• •	Source input_brake cno_sigBrake speed	+	Offset 0 0 0	Mapping OnChange OnChange OnChange
	ol System er Variables	el System Symbol Variables Symbol Symbols	ol System Symbol Status variables Symbols	el Correction Symbol Start Correction Symbols	ol System Symbol Start Commun Variables Mapping Symbols	er System Symbol Variables Mapping Symbols	er Variables Symbol Symbols	er Variables Symbol Symbols Symbol Start Communication Symbols Symbol Setup - All Nodes Market Mapping Symbols Market Symbols Market Symbols Market Symbol Symbol Setup - S	er Variables Symbol Symbols Symbols Start Communication Setup - All Nodes Ocuments All Nodes More



Signals

- Input_brake: current brake pedal position of the driver (DM.Brake)
- Objectdistance: distance to object nearest point (Sensor.Object.OB00.relvTgt.NearPnt.ds_p)
- Speed: Current vehicle speed (Car.v)
- Brake: Control signal for vehicle control (VC.Brake)





CarMaker signal mapping

Follow the usual FMU integration procedure

- Place exported FMU in plugins folder of the project directory
- Go to FMI GUI from Applications>FMU Plug Ins and update list
- Select the appropriate model class (Vehicle control in this example) and connect all the required FMU signals
- Replace the required subsystem model with FMU in the vehicle data set
- Start simulation

CarMaker - FMU Plug-ins						-		×
MU Plug-ins							Close	
Verview FMU Details								
Μ	lodel class: V	ehicle Control	~ Settin	gs 🔻				
	FM	IU: CarMaker_CANoe_C	oSim					
From CarMaker: Signals: 32	FMU Inputs: Signals:	3 FMU Params: 3 Signals: 7	FMU Outputs: Signals:	:		To CarMa Signals:	aker: 32	
Connected: 0	Connected:	3 Connected: 0	Connected:	1		Connecte Data Dic	ed: 1 tionary	
Connected: 0	Link type	3 Connected: 0	Connected:	1 Choose sign		Connecte Data Dic	ed: 1	
Connected: 0 Signal name FMU Inputs	Link type	3 Connected: 0 Signal source / destination	Connected:	1 Choose sign	nal source	Connecte Data Dic e / destin	ed: 1 tionary ation	
Connected: 0	Link type	3 Connected: 0 Signal source / destination DM.Brake	Connected:	1 Choose sign	nal source	Connecte Data Dict e / destin	tionary	
Connected: 0 Signal name FMU Inputs →•• sensordata.input_brake →• sensordata.objectdistance	Link type DDict DDict	3 Connected: 0 Signal source / destination DM.Brake Sensor Object OB00.relvTgt.Nea	Connected:	Choose sign	nal source	Connecte Data Dict	tionary	
Connected: 0 Signal name FMU Inputs +• sensordata.objectdistance +• sensordata.objecdi	Connected: Link type DDict DDict DDict	3 Connected: 0 Signal source / destination DM.Brake Sensor.Object.OB00.relvTgt.Nea Car.v	Connected:	1 Choose sign	nal source	Connecte Data Dict	tionary	
Connected: 0 Signal name FMU inputs Growsensordata.objectdistance Growsensordata.speed FMU Parameters	Connected: Link type DDict DDict DDict	3 Connected: 0 Signal source / destination DM.Brake Sensor Object OB00.relvTgt Nea Car.v	connected:	1 Choose sign	nal source	Connecte Data Dict	ation	
Connected: 0 Signal name FMU inputs FMU inputs FMU sensordata.objectdistance FMU Parameters FMU Outputs FMU Outputs	Connected: Link type DDict DDict DDict	3 Connected: 0 Signal source / destination DM.Brake Sensor.Object.OB00.relvTgt.Nea Car.v	rPnt.ds_	1 Choose sign	al source	Data Dic	iionary	

Note:

- All signals may not appear in the data dictionary of FMI GUI in the beginning. Run the testrun briefly once to update the list of CarMaker signals.
- The Driver setting has to be set so that the driver does not consider traffic. Otherwise the driver model will interfere with the controller built in the example.

Simulation Visualization



Comparing signals with IPGControl and CANoe-Graphics tools

CANoe-Graphics tool

CarMaker-IPGControl