Tutorial 5 — FMU Export (20-sim)

Overview

This INTO-CPS tutorial will show you how to:

1. Generate a new physical model simulator FMU in 20-sim

   (a) Import a model description into 20-sim
   (b) Complete the skeleton model to produce a working simulator
   (c) Export the simulator FMU

2. Associate the new simulator FMU with a multi-model configuration
3. Execute a co-simulation using the new simulator

In the previous tutorials, you have used a pre-exported *Body_Block* FMU. In this tutorial, you will find out how that FMU was produced, and you will have the opportunity to modify its behaviour.

Requirements

This tutorial requires the following tools from the INTO-CPS tool chain to be installed:

- INTO-CPS Application
- COE (Co-simulation Orchestration Engine) – accessible through the INTO-CPS App Download Manager
- 20-sim - Modeling and simulation tool

1 Download and install the 20-sim tool (Windows only)

Step 1. Launch the *INTO-CPS Application*. Download the 20-sim modeling tool from the *Window > Show Download Manager*. If after the download click *Yes*. If the window does not appear please find the installer inside the *into-cps-projects/install_downloads* folder.
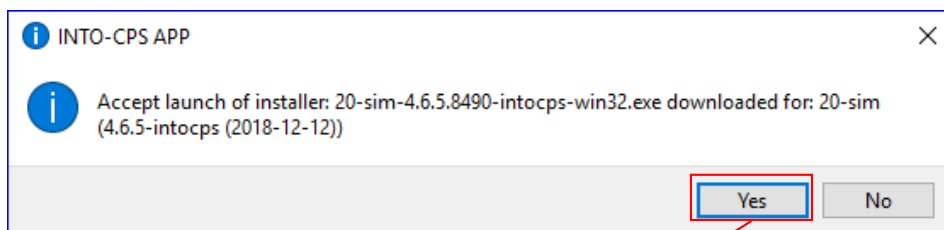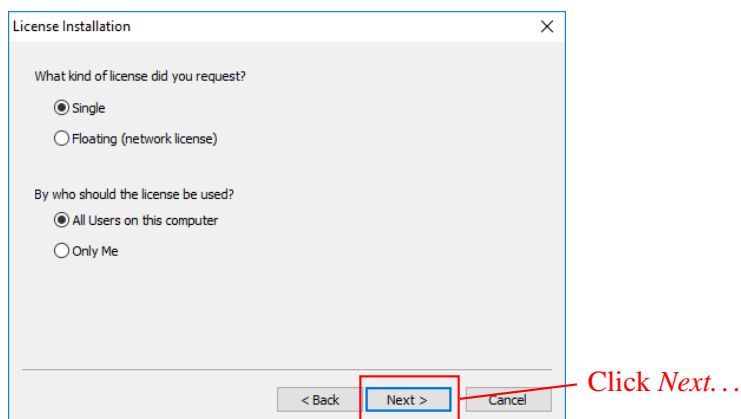

Click *Yes. . .*

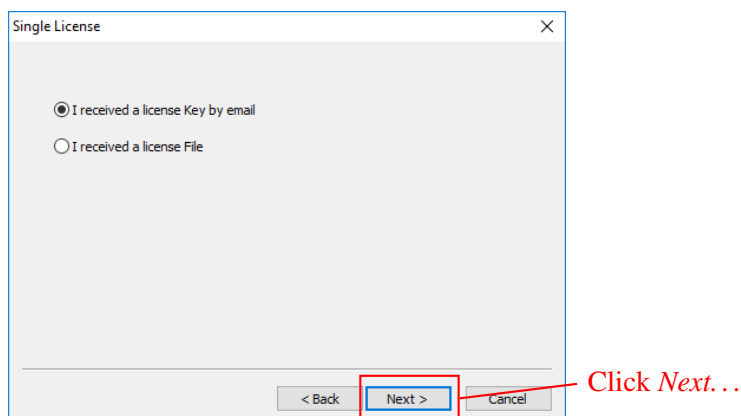Step 2. Follow the installation wizard and make sure to enable Python support.

Step 3. Launch 20-sim, and click *Activation* when the activation screen appears.

Click *Activation...*

Step 4. The following options will appear, click *Next* to continue.
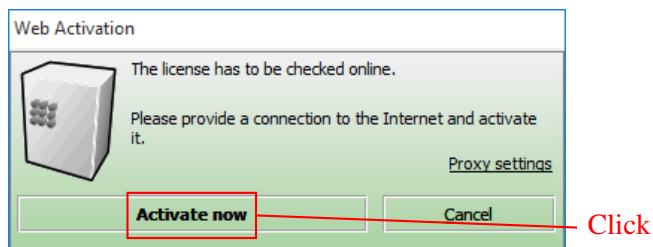

Click *Next...*

Step 5. Select *I received a license Key by email* and click *Next* to continue.

Click *Next...*

Step 6. Please fill in the license key you were given (Available in the blackboard system) and click *Finish*.

Paste *here…*
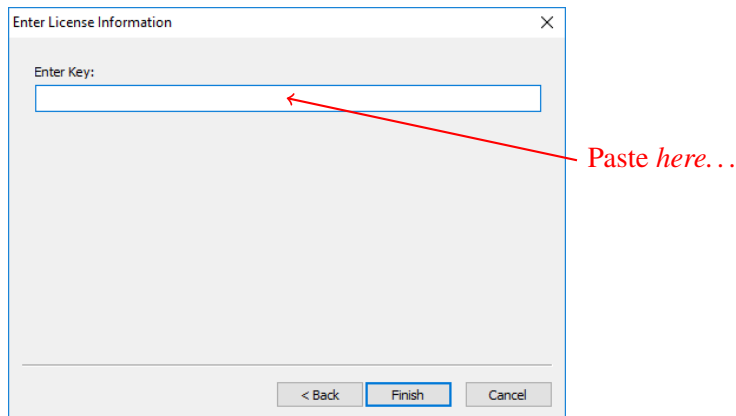
Step 7. You will be asked go online. Click on *Active Now* to finish activation.

Click

## 2   Download and install the 20-sim tool (Linux and macOS only)

It is possible to run 20-sim on Linux and macOS systems after installing Wine.

- For the macOS there is a disk image file (.dmg) ready to install. Please ask us.

- For Linux

  1. Install the Wine package for your Linux distribution or follow the instructions for your Linux distribution from `https://wiki.winehq.org/Download`

  2. Use the winetricks tool to install a few Microsoft DLLs to fix the license activation and some drawing issues:
     - winetricks msxml4
     - winetricks wininet
     - winetricks gdiplus

  3. Now download 20-sim from `http://www.20sim.com/downloads/files/20sim.exe` and install it using: `wine 20sim.exe`
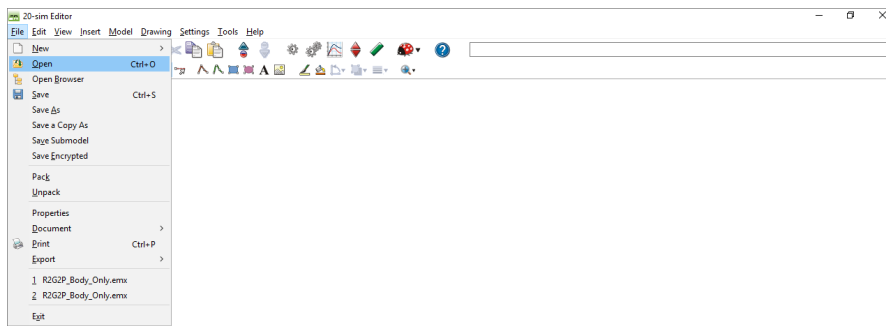
  This should result in a working 20-sim version on Linux. Tested here with various Debian and Ubuntu versions and Wine version 1.5 up to Wine version 3.0.
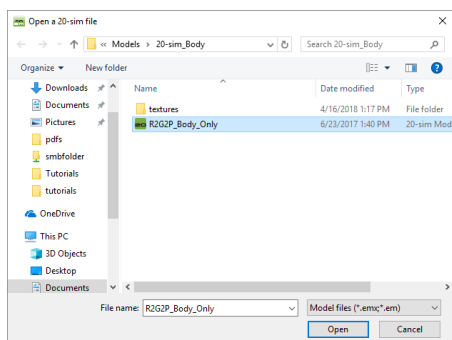
3   Loading a Model in 20-sim

Step 8. Launch the *20-sim* tool. You should see the following 20-sim window.
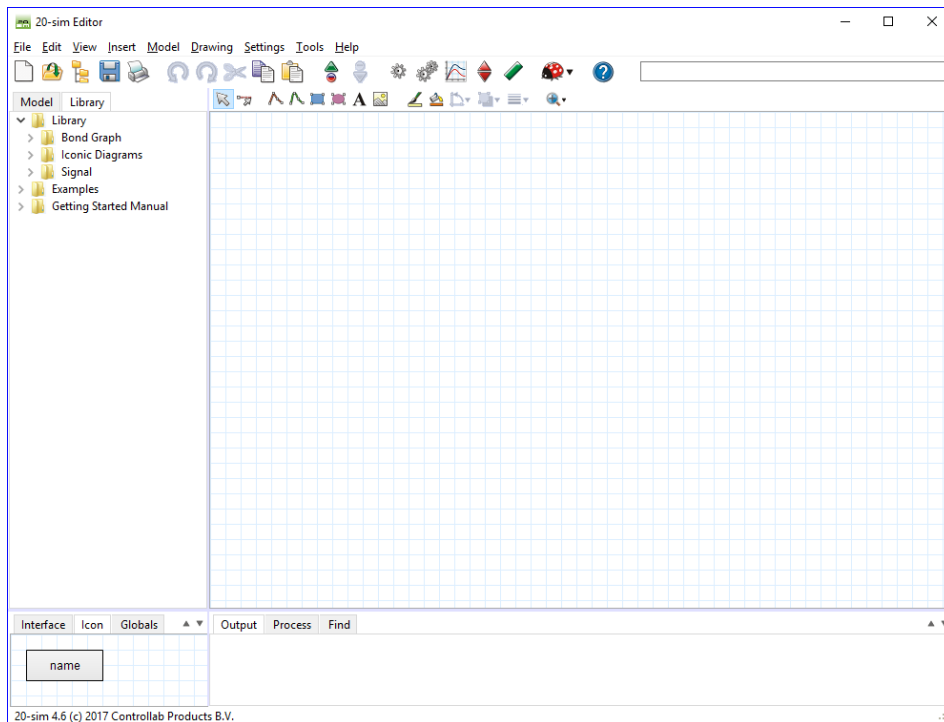
Step 9. Open the 20-sim physical model of the line follower robot by selecting *File > Open*.


Step 10. Navigate to the location of *tutorial_5/Models/20-sim_Body* and *Open* the *R2G2P_Body_Only* model.

Step 11. You should see the block diagram for the model. The diagram is composed of two blocks: The *Environment* and the *Body_Block*. The Body_Block corresponds to the *Body_Block* FMU, and the Environment provides default inputs to enable the simulation of the Body_Block. Once the Body_Block FMU is exported, the inputs to the Body_Block will be provided by the co-simulation. Click on the Body_Block to visualize its contents.
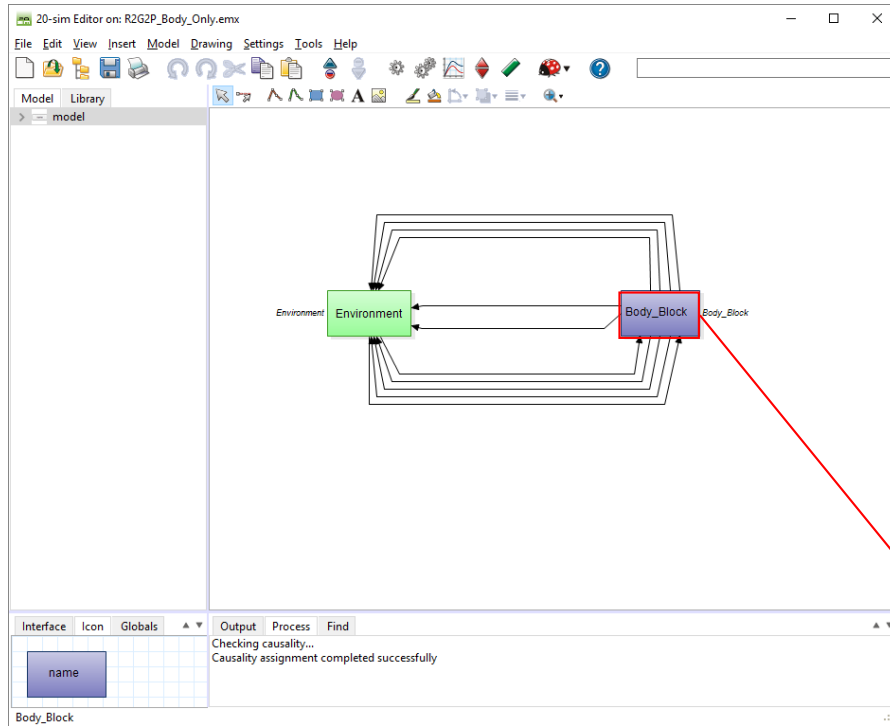
Step 12. At this step you can explore the model further by clicking on blocks or by clicking on the Model view. To continue our tutorial open the *wheel_right* submodel.
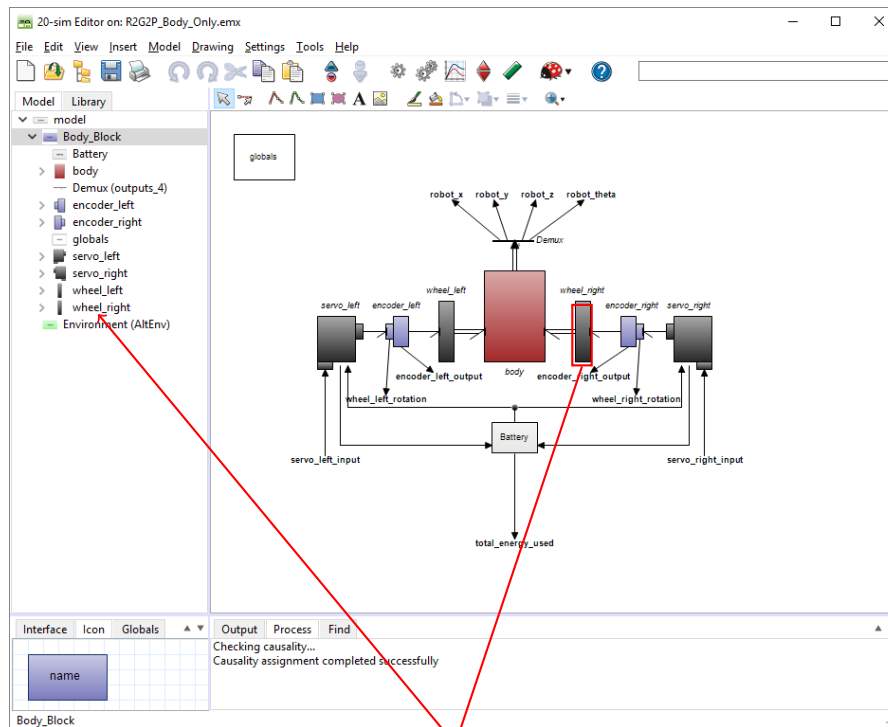

Click the wheel_right

Step 13. The *wheel_right* models the physical behaviour of the line following robot wheel. You can explore the model but we will focus on the *rotation_to_translation* element. This element is a transformer which as it implied by its name transforms the rotation of the wheel into the translation of the line following robot body structure.
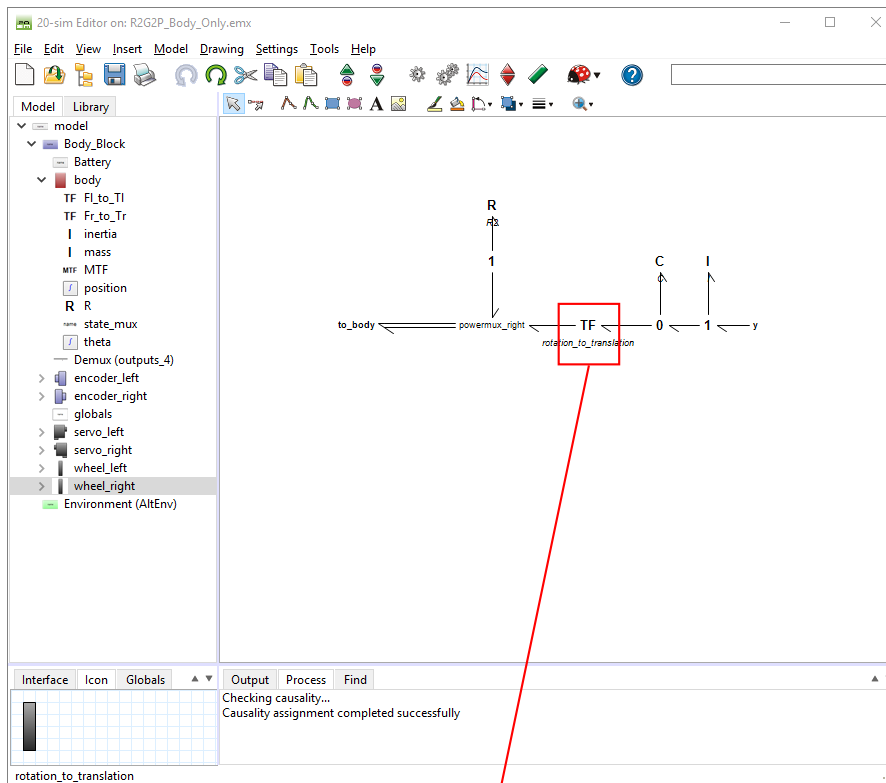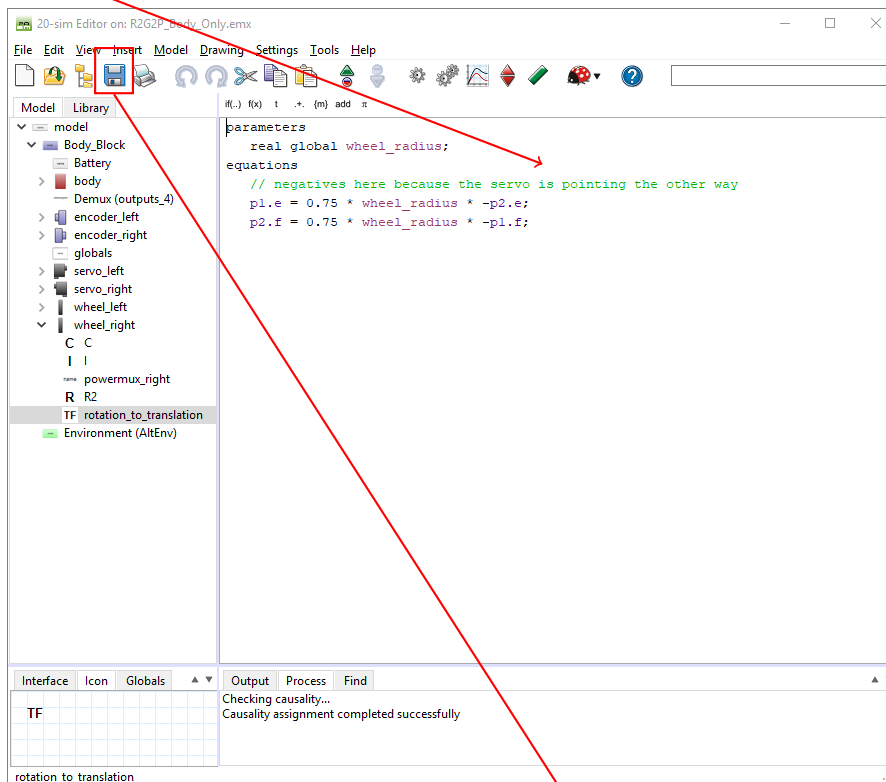

rotation_to_translation

Step 14. After double clicking the transformer you should find its contents to be empty.
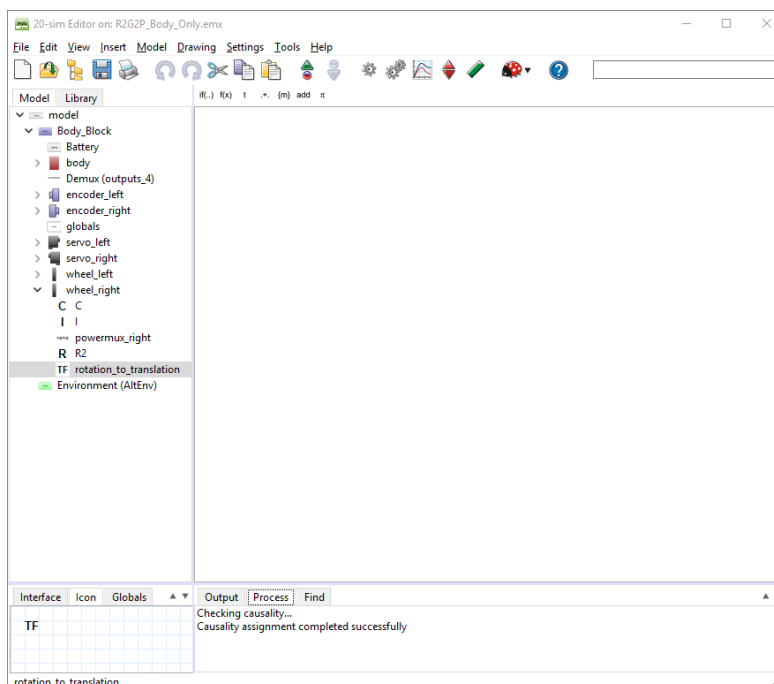
Step 15. Fill in the *rotation_to_translation* with the content shown on the figure, and save afterwards.

1. Fill in

2. Click *Save*

Congratulations!

You edited your first 20-sim physical simulator.

In the following sections, you'll need either a Windows system or an emulated Windows platform, to be able to export an FMU for the physical model of the Line Following Robot. The simulator FMU can be provided to you and you can continue in Section 6. If you cannot follow along in your computer, join one of your colleagues.

4   Installing Microsoft Build Tools (Windows Only)

20-sim is able to export new FMUs however *this feature depends on Microsoft Build Tools being available*, because 20-sim exports the source code of the FMU and then builds it using Microsoft Build Tools. If you suspect you already have Microsoft Build Tools installed, try to follow the instructions in Section 5. If 20-sim fails to build the FMU, then refer to the following step as one possible way to install Microsoft Build Tools on your system.

Step 16. Install (or modify the existing installation of) Visual Studio Community 2019, and include *Desktop Development with C++*.
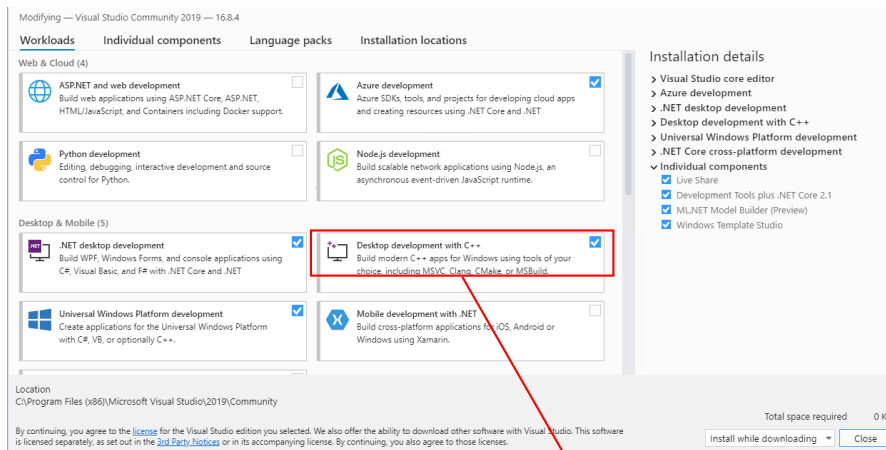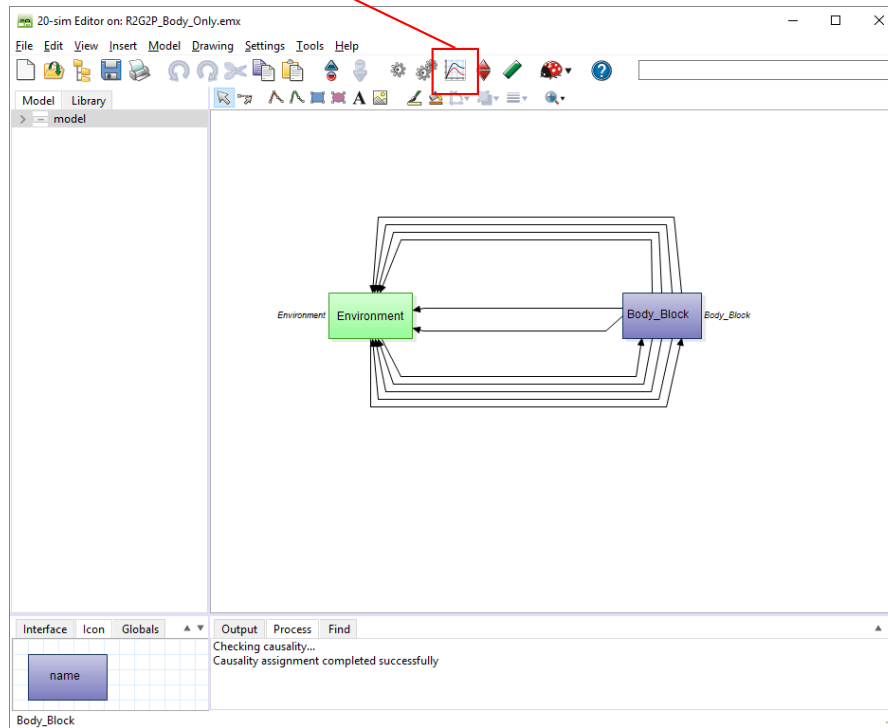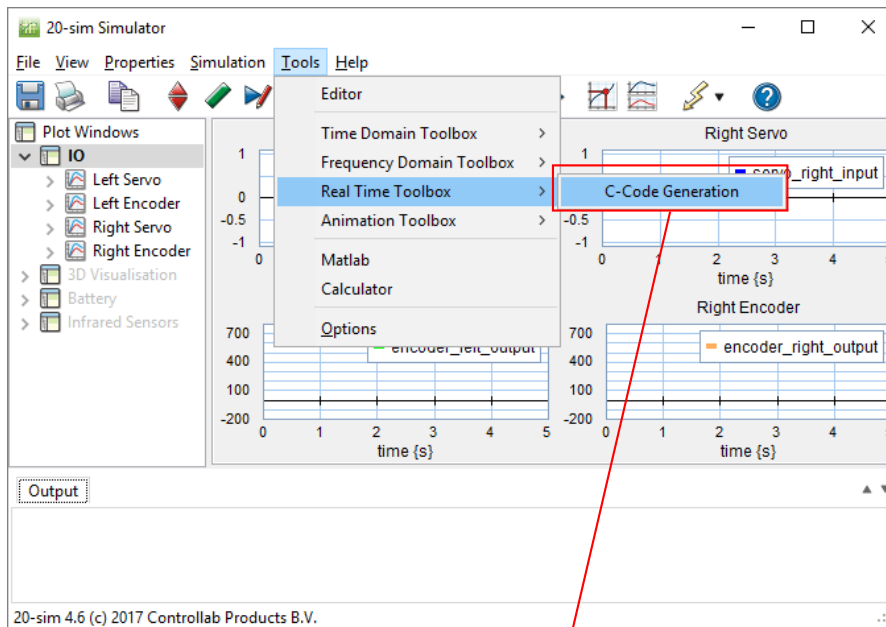

Check

5   Exporting an FMU and Adding it to a Multi-model

Step 17. Reload the 20-sim model again by repeating Step 8. to Step 10. and click *Start Simulator* to open the Simulator Window.
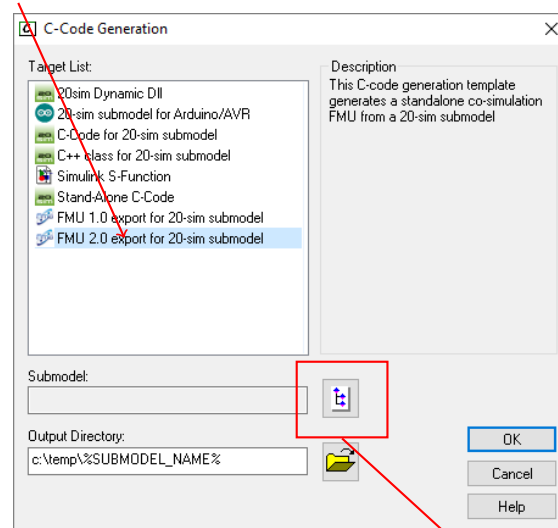
Step 18. After the *Simulator Window* appears, navigate to *Tools> Real Time Toolbox> C-Code Generation* and click *C-Code Generation*.
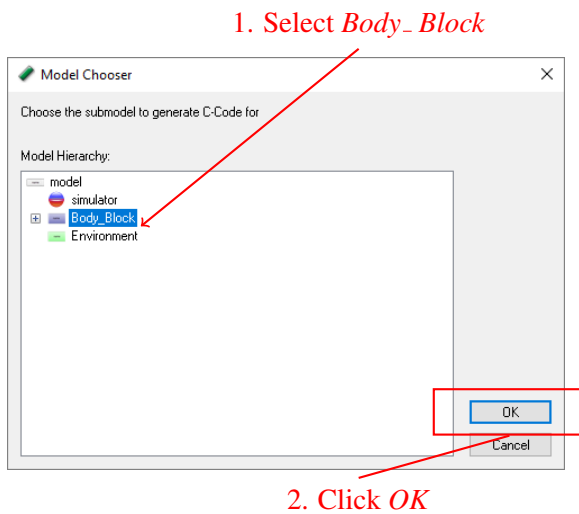

Click *C-Code Generation*

Step 19. Select *FMU 2.0 export for 20-sim submodel* from the *Target List* and click on the *Submodel* icon.
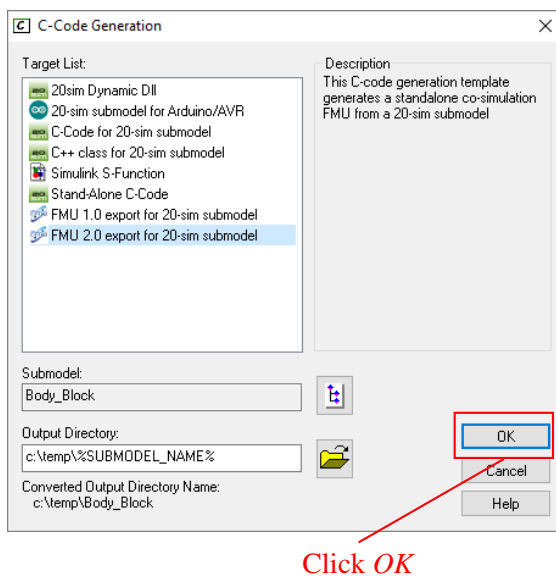
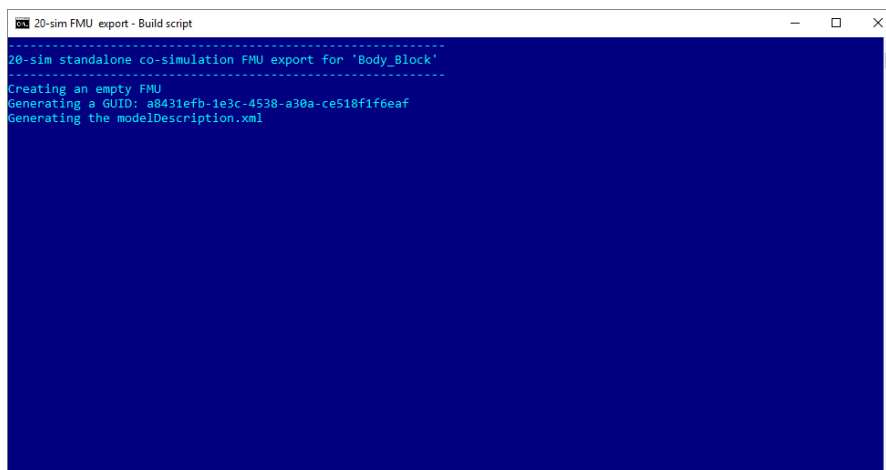1. Click on *FMU 2.0 export...*

2. Click on the *Submodel* icon

Step 20. A model chooser appears. Select the *Body_block* in the Model Hierarchy.

1. Select *Body_ Block*

2. Click *OK*

Step 21. You may change the location for the FMU output by changing the *Output Directory* or leave it to its default location and click *OK* to generate the FMU.
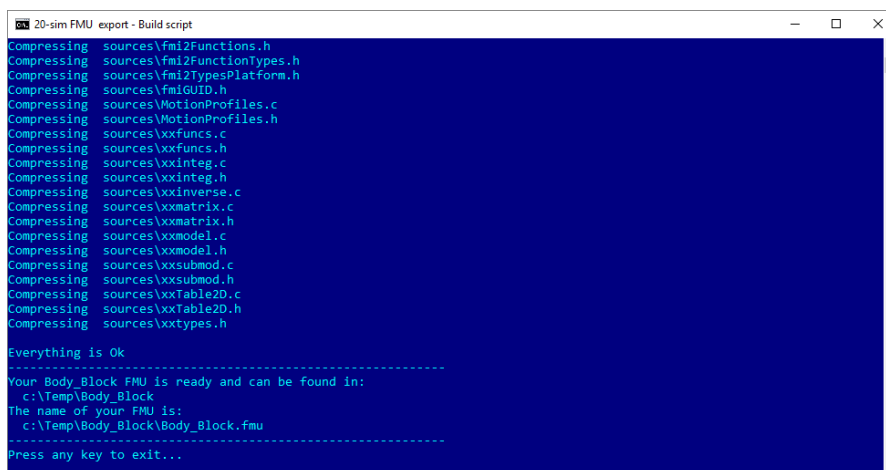

Click *OK*

Step 22. The following shell should pop up and the script to be run may take a while to complete.

Step 23. When the script is finished and the FMU generation succeeded you should see the following message.

Common issues at this step:
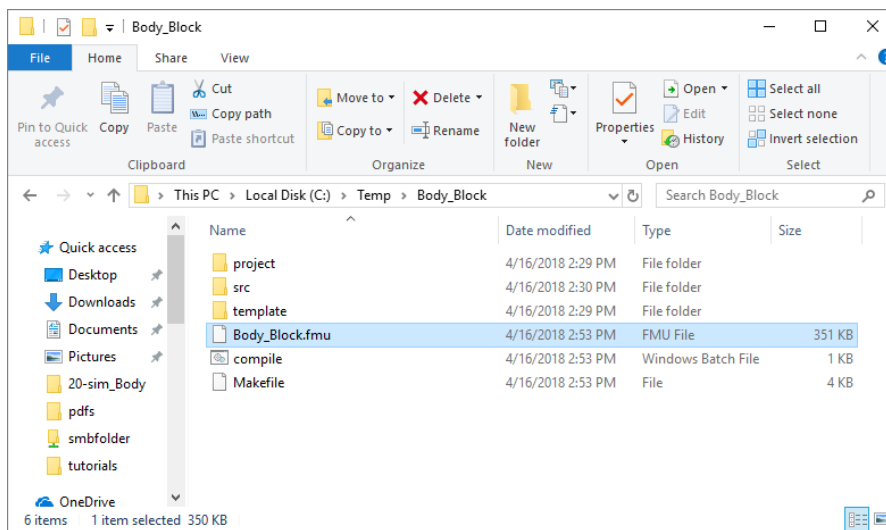
- 20-sim is unable to collect resources – This might be caused by python not having been installed when you ran the 20-sim setup. Try reinstalling 20-sim with that support enabled.

- 20-sim cannot determine the location of the VS Common Tools folder – There are multiple possible causes for this error:

    1. You don't have Microsoft Build Tools installed, and/or
    2. The script *vsvars32.bat* expects to find a registry entry that contains the path to the Common tools folder (e.g., *C:\Program Files (x86)\Microsoft Visual Studio 14.0\Common7 \Tools*). This issue is described in [1]. The easiest solution is to uninstall Microsoft Build Tools 2015 follow the instructions in Section 4.

---

[1] https://stackoverflow.com/questions/3461275/vs2010-command-prompt-gives-error-cannot-determine

Step 24. Navigate to the chosen folder and copy the *Body_Block.fmu*.

Step 25. Paste the *Body_Block.fmu* into the FMUs folder inside tutorial 5..
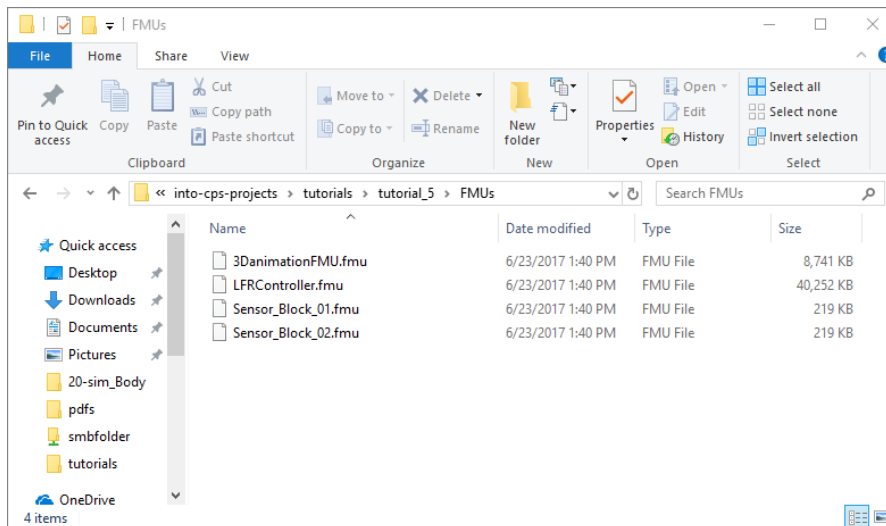
Congratulations! You are now ready to use the generated FMU in a co-simulation.

6 Co-simulating with the New Simulator

Step 26. Launch the *INTO-CPS Application* and select *File > Open Project*. Set the *Project root path* to the location of *Tutorials/tutorials_5* and click *Open*. You can browse using the *Folder* button.

Path to *Tutorials/tutorials_5*
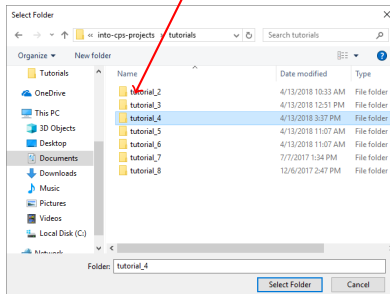
You should see the newly export *Body_Block* FMU in the list.

Newly created FMU →
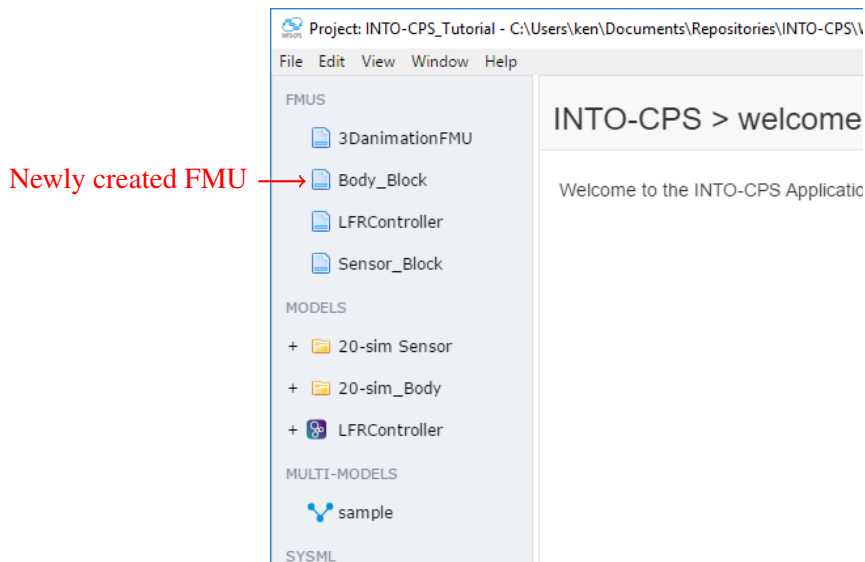
Step 27. In the SysML entry of the project browser, expand the *LineFollowRobot* folder, then *config* folders. Right-click on *3DRobot* and select *Create Multi-Model*.

Expand to locate *3DRobot*

Create Multi-model

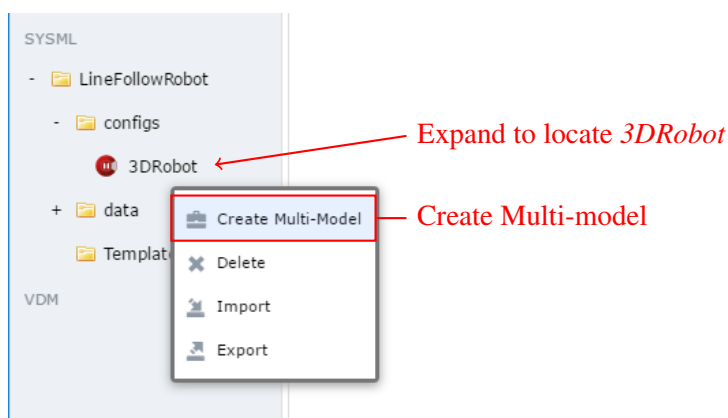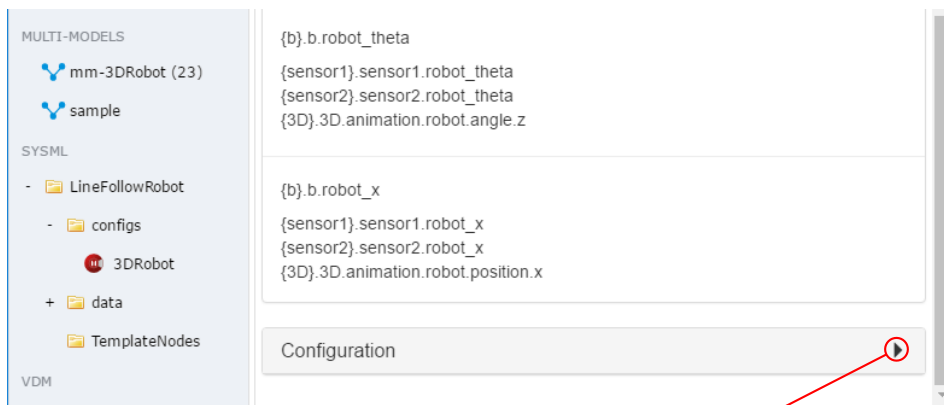Step 28. We now need to associate FMUs to the multi-model as we did in *Tutorial 2*. Scroll down to find the *Configuration* panel and expand it by clicking the arrow.

Expand *Configuration*

Step 29. Scroll down and click *Edit*.
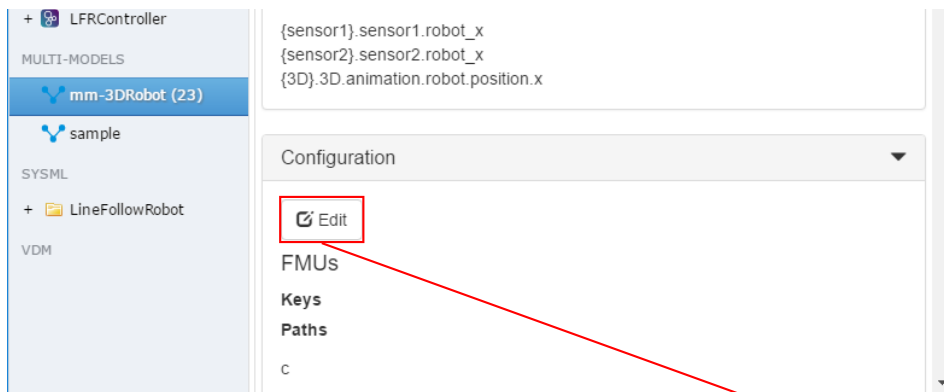
Edit configuration

Step 30. As in *Tutorial 2*, in the FMUs section press *File* next to the Body_ Block element, *b*. A file browser window will open and show five FMUs (if the file browser does not show the FMUs, navigate to *tutorials_5/FMUs*). Select *Body_Block.fmu* and click *Open*.

1. Locate and select *Body_Block.fmu*
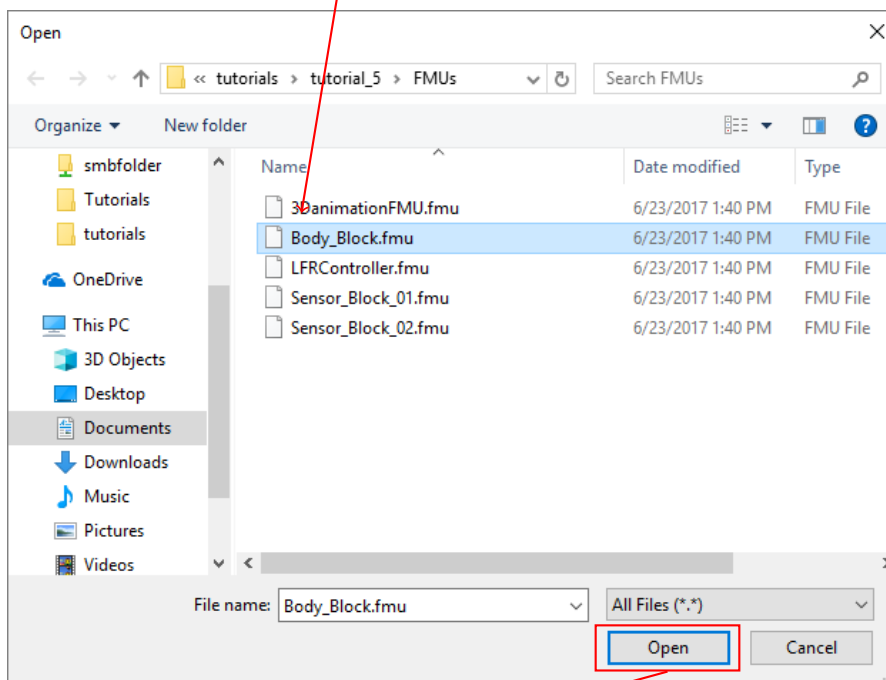
2. Click *Open*

Step 31. Repeat this for the remaining elements:

- *c* : *LFRController.fmu*
- *3D* : *3DanimationFMU.fmu*
- *sensor1* : *Sensor_Block_01.fmu*
- *sensor2* : *Sensor_Block_02.fmu*

The complete set of FMUs will look like this:

Project: INTO-CPS_Tutorial - C:\Users\ken\Documents\Repositories\INTO-CPS\WP3\TrainingUSB\Tutorials\tutorial... — □ ×

File Edit View Window Help
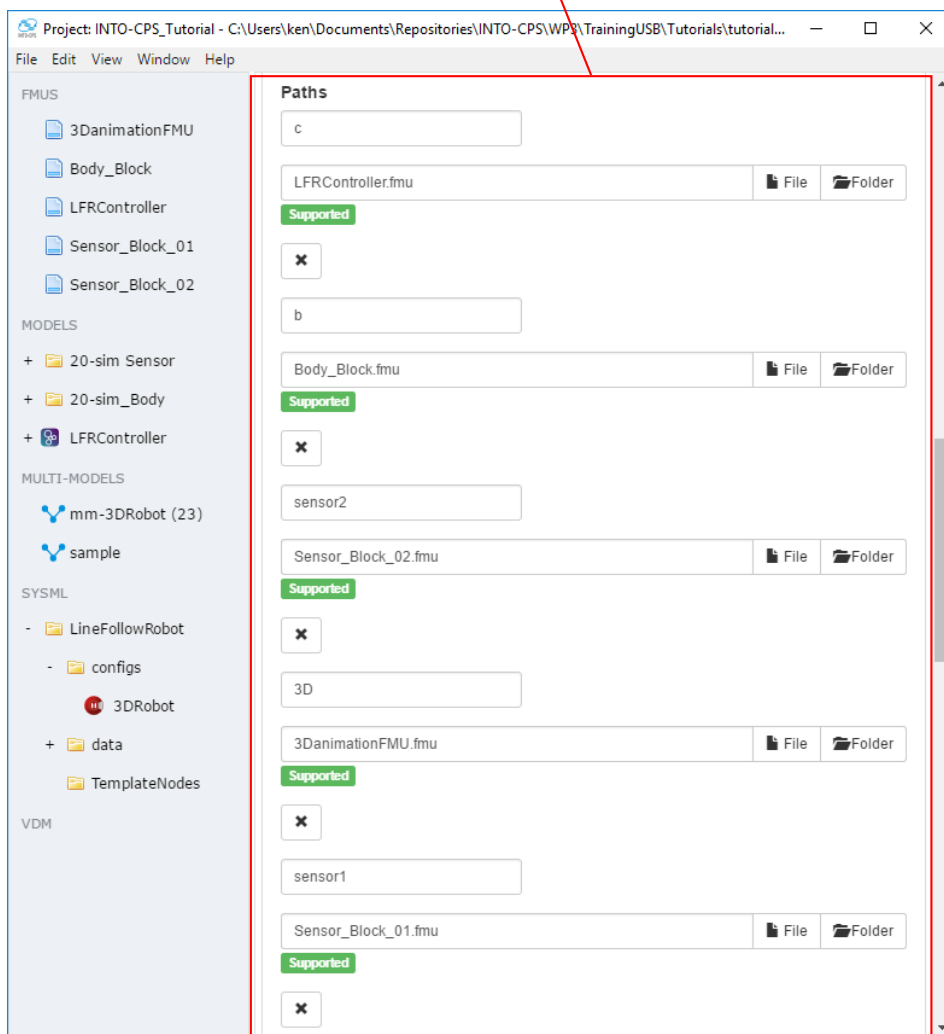
FMUS
- 3DanimationFMU
- Body_Block
- LFRController
- Sensor_Block_01
- Sensor_Block_02

MODELS
- + 20-sim Sensor
- + 20-sim_Body
- + LFRController

MULTI-MODELS
- mm-3DRobot (23)
- sample

SYSML
- LineFollowRobot
  - configs
    - 3DRobot
  - + data
  - TemplateNodes

VDM

Paths

c

LFRController.fmu   📄 File   📁 Folder
Supported

✖

b

Body_Block.fmu   📄 File   📁 Folder
Supported

✖

sensor2

Sensor_Block_02.fmu   📄 File   📁 Folder
Supported

✖

3D

3DanimationFMU.fmu   📄 File   📁 Folder
Supported

✖

sensor1

Sensor_Block_01.fmu   📄 File   📁 Folder
Supported

✖

Step 32. Scroll down to the *Initial values of parameters section*, and click {*sensor1*}.*sensor1*. In the *Parameters* section, enter the following values:
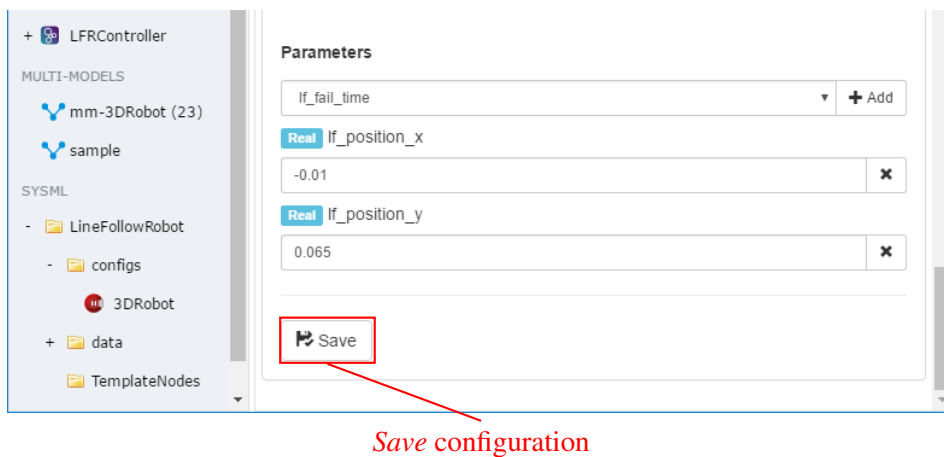
- *lf_position_y* = 0.065
- *lf_position_x* = 0.01

lf_position_y   *lf_position_x*
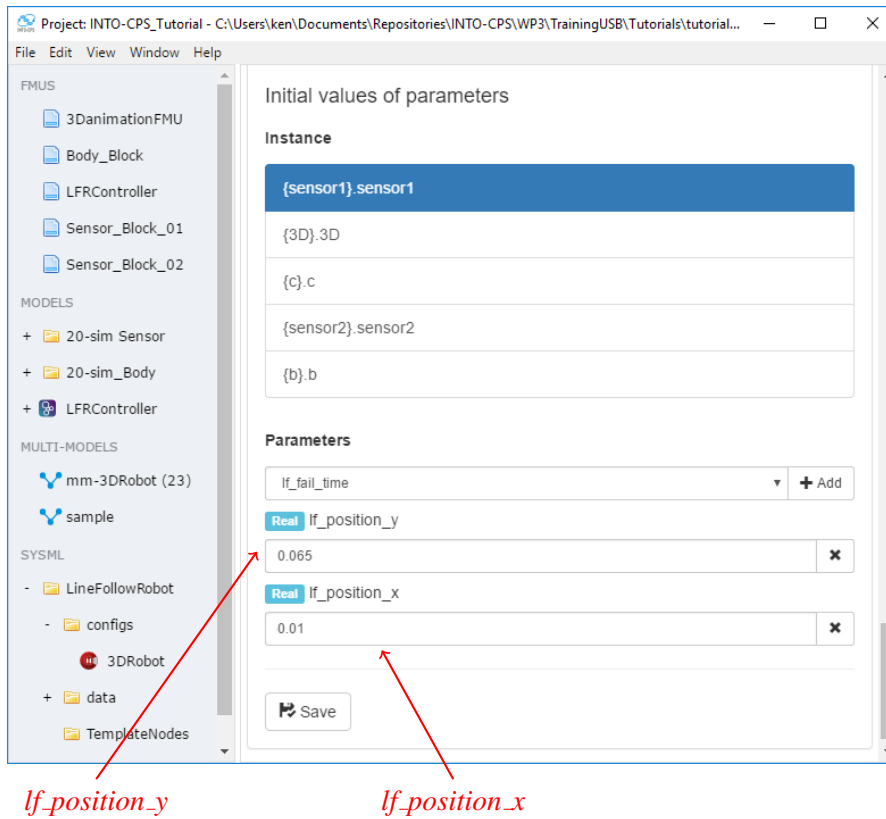
Step 33. Repeat the previous step for the second sensor, {*sensor2*}.*sensor2*, with the following values:
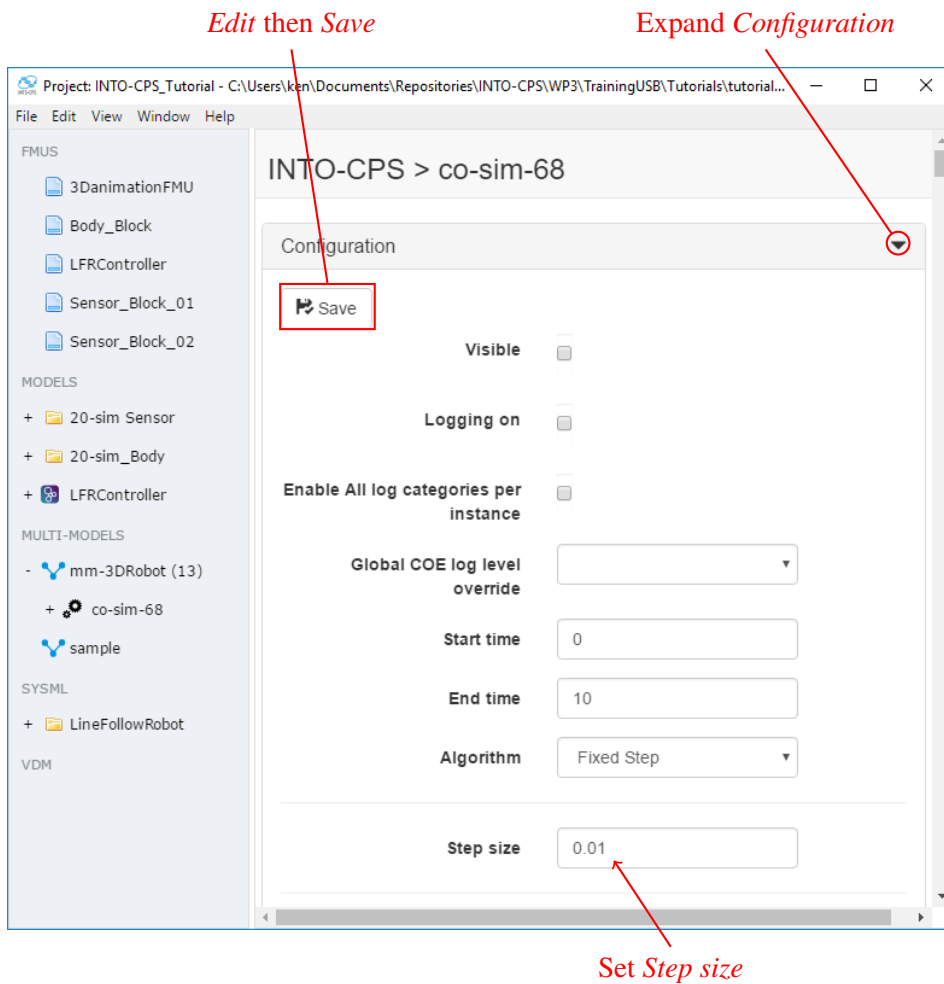
- *lf_position_x* = -0.01
- *lf_position_y* = 0.065

Step 34. *Save* the *Configuration*.

Save configuration

Step 35. Right-click on the new multi-model configuration and select *Create Co-simulation Configuration*.

Create Co-Simulation Configuration
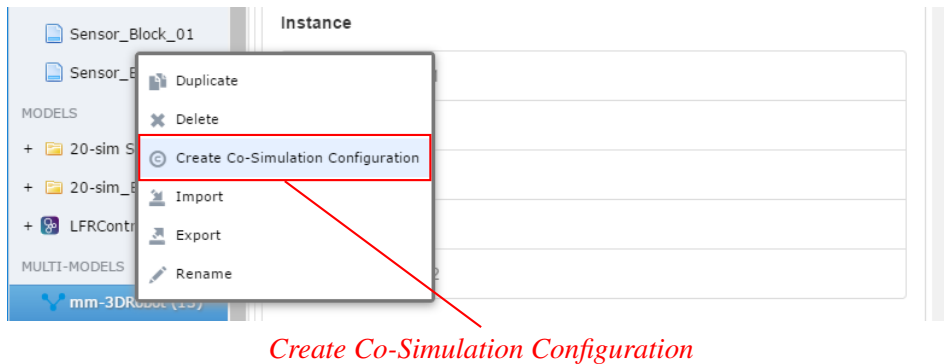
Step 36. Set the *Step size* to 0.01. Don't forget to press *Edit* then *Save*.

*Set Step size*

Step 37. Check *lf_1_sensor_reading* from {*sensor1*}.*sensor1* and {*sensor2*}.*sensor2* to see the sensor values appear in the *Live Plotting*.

Step 38. Launch the COE if necessary (see *Tutorial 1 — First Co-simulation* for a reminder if needed).

Launch COE
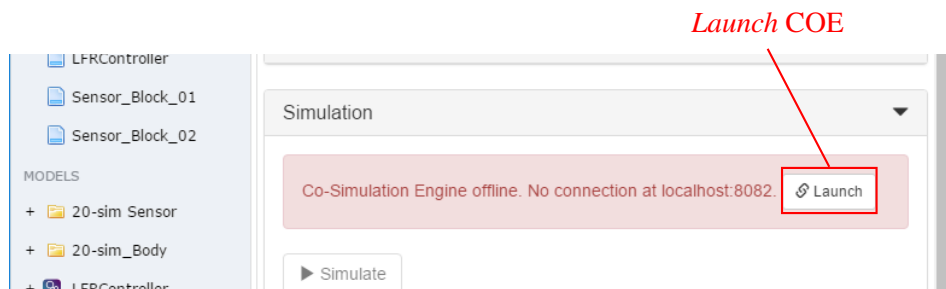
Step 39. When the COE is running, click the *Simulate* button. If you are on Windows, the *Animation Frame* should appear. You can click the *3D* button to see the 3D visualisation of the robot.

3D Button
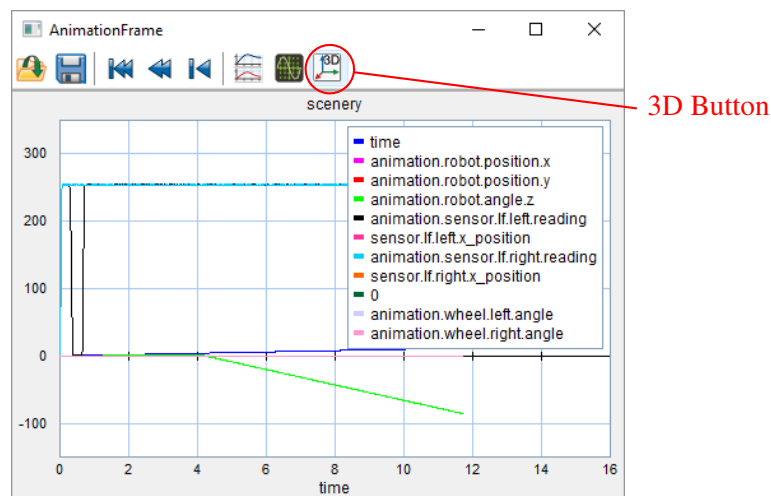
Step 40. If everything went well, the robot should follow the line as in *Tutorial 2 — Adding FMUs*. Although with a slight *difference...*
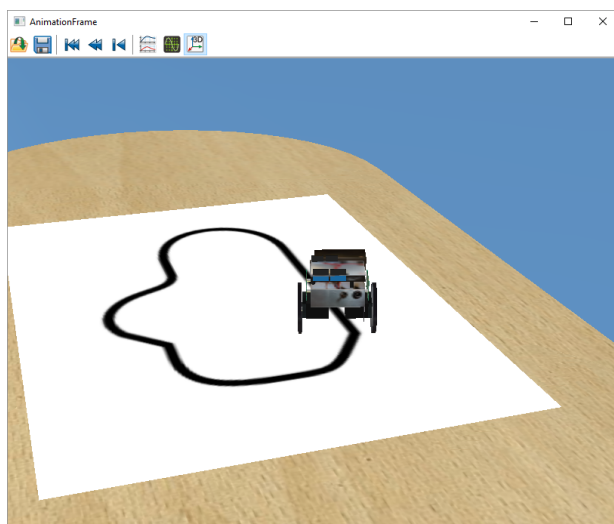
7 Additional Exercises

Can you spot the *difference* between the co-simulation you run in this tutorial and the co-simulation you run in the previous tutorial? You can go back to *20-sim* and explore[2] the physical model. Try to make some changes to obtain the correct behaviour. Just repeat Step 17. to Step 25. to regenerate and copy the FMU, then press *Simulate*.

---

[2]Ask for a hint in case you feel lost...