Tutorial 2 — Adding FMUs (No SysML)

Overview

This second tutorial will show you how to:

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

Note that a version of this tutorial that includes the extension of the SysML model to include instance and connection creation is available as Tutorial 2 — Adding FMUs.

Requirements

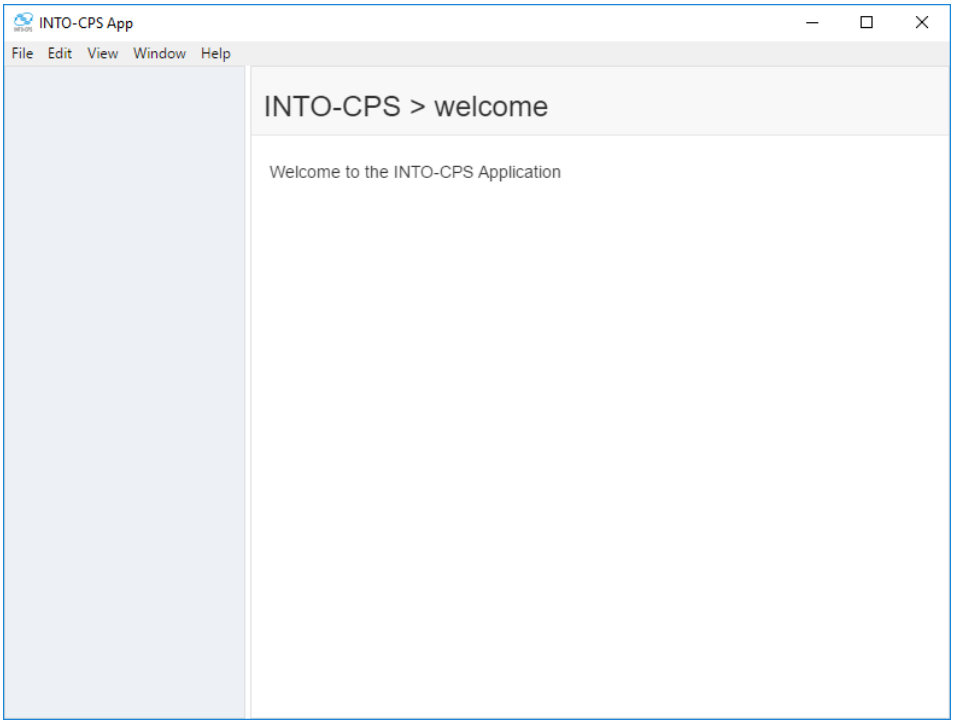
This tutorial requires the following tools from the INTO-CPS tool chain to be installed:

- INTO-CPS Application
- COE (Co-simulation Orchestration Engine) accessible to the Application

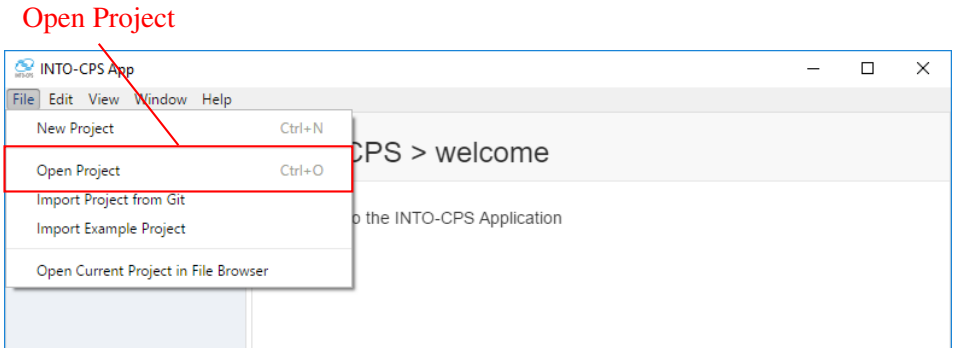
Tools can be downloaded through the Application (*Window* > *Show Download Manager*) or may have been provided on the USB drive at your training session. Please ask if you are unsure.

1 Opening a Project

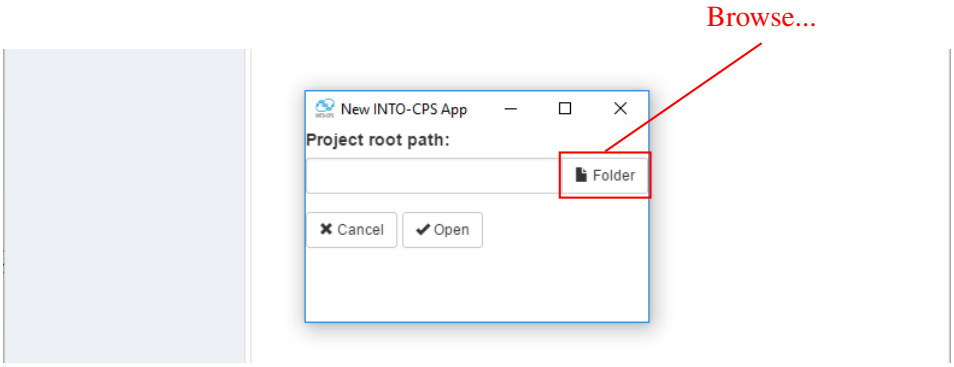
Step 1. Launch the *INTO-CPS Application*. On first loading, it will look like the screenshot below. If you have opened a project previously, that project will be opened automatically.



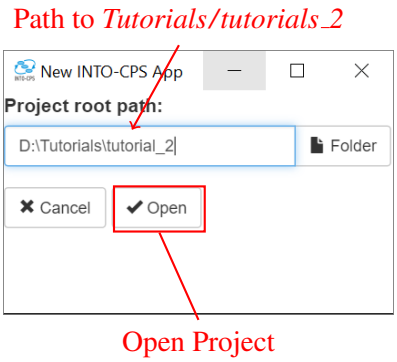
Step 2. To open a project, select *File* > *Open Project*.



Step 3. Set the *Project root path* to the location of *Tutorials/tutorials_2*. You can browse using the *Folder* button.



Step 4. Click *Open Project*.



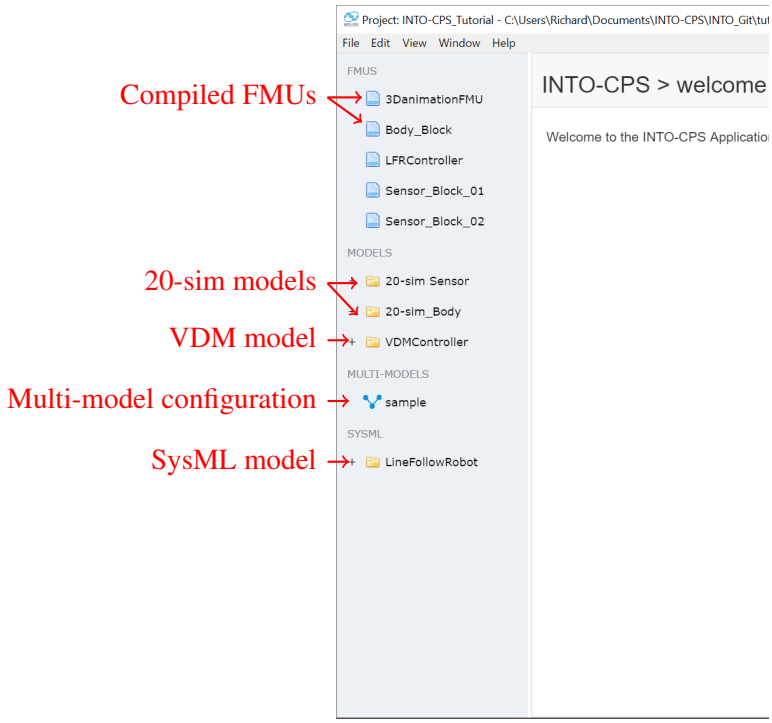
Step 5. Once the project is opened, you will see that project browser on the left of the INTO-CPS Application window is now populated. The entries in the project browser correspond to folders and files in the *Tutorials/tutorials_2* folder. These are:

FMUs Compiled FMUs (with file extension .fmu) that are used in co-simulation.

Models Source models used to generate the FMUs. The icon of each entry shows which tool created the model. In this case Overture and 20-sim.

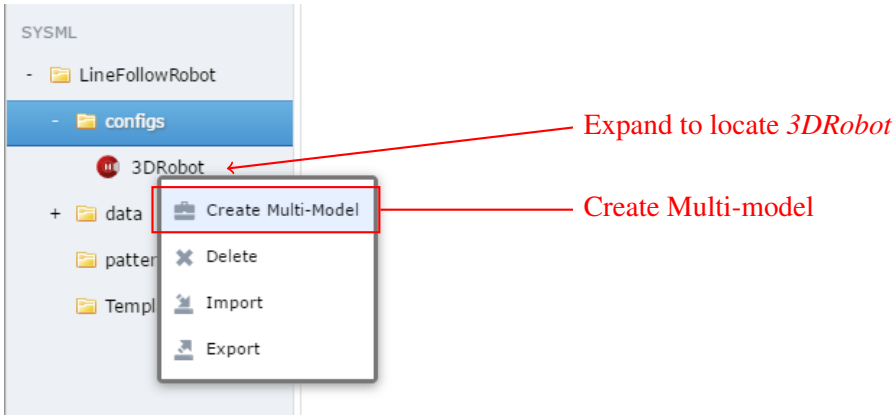
Multi-models Used to configure co-simulations, including which FMUs are used and other co-simulation settings.

SysML Architectural models that are used to create model and multi-model descriptions.



2 Configuring a Multi-model

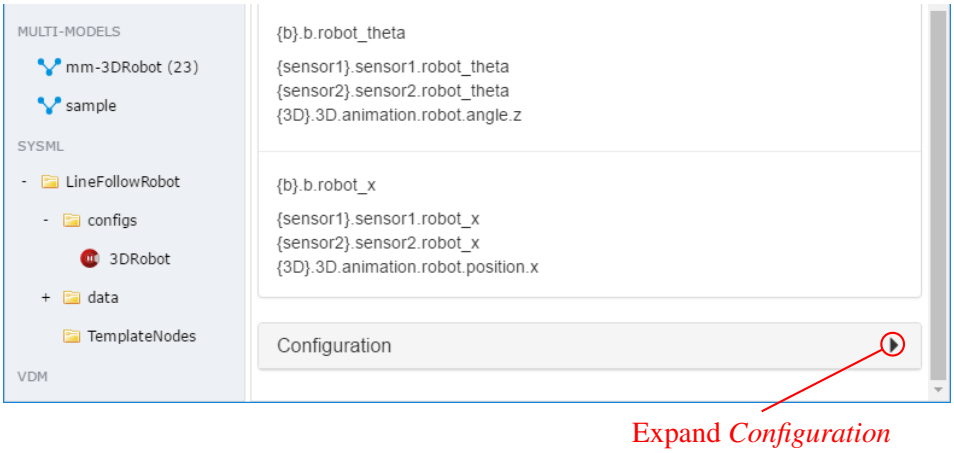
Step 6. In the SysML entry of the project browser, expand the *LineFollowRobot* and then *config* folders. There should be a *3DRobot* icon (as in the Figure below). Right click on *3DRobot* and select *Create Multi-Model*.



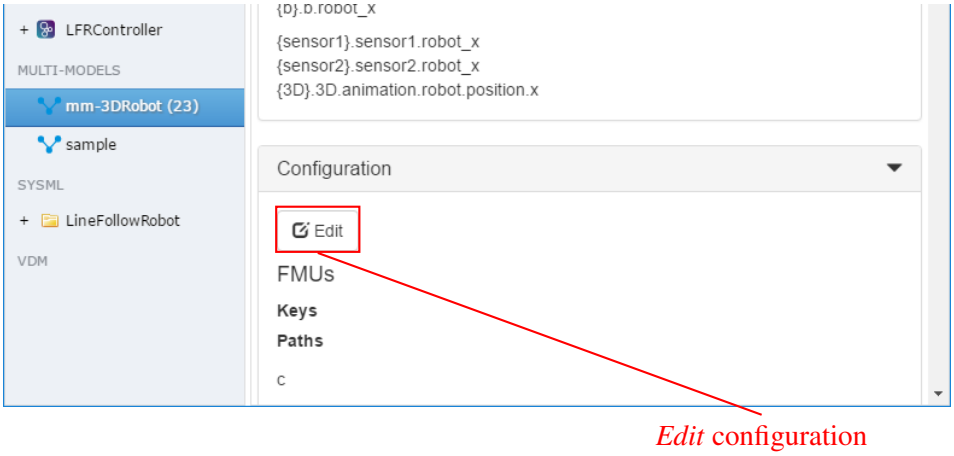
Step 7. A new multi-model configuration has been created and is shown in the multi-model entry of the project browser. Double-click on the new multi-model to open it.



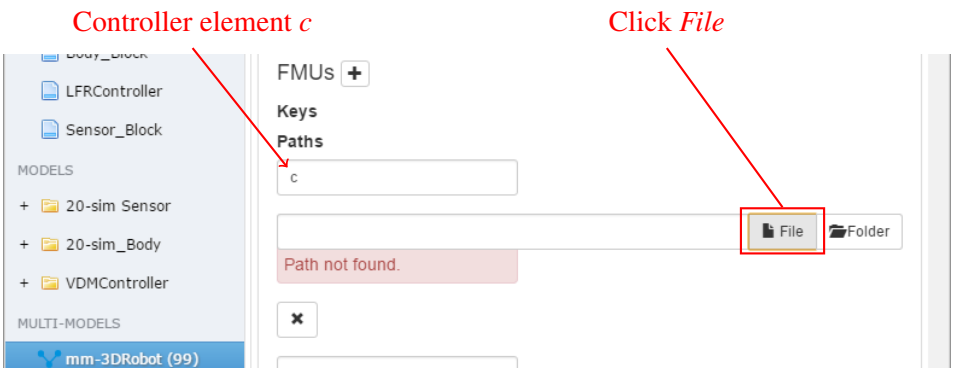
Step 8. We need to associate FMUs with this multi-model and set its parameters. Expand the *Configuration* section of the multi-model by clicking on the triangle.



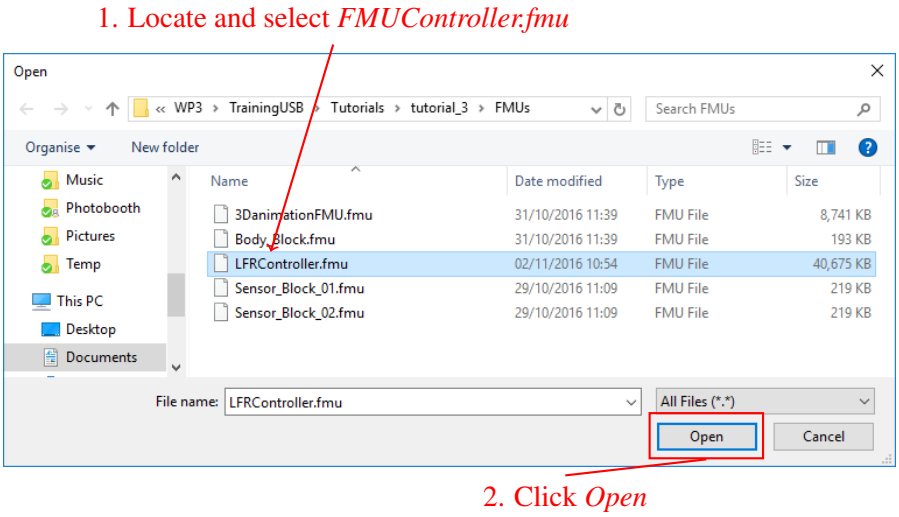
Step 9. Scroll down and click *Edit*.



Step 10. In the *FMUs* section, next to the *Controller* element *c*, click the *File* button.

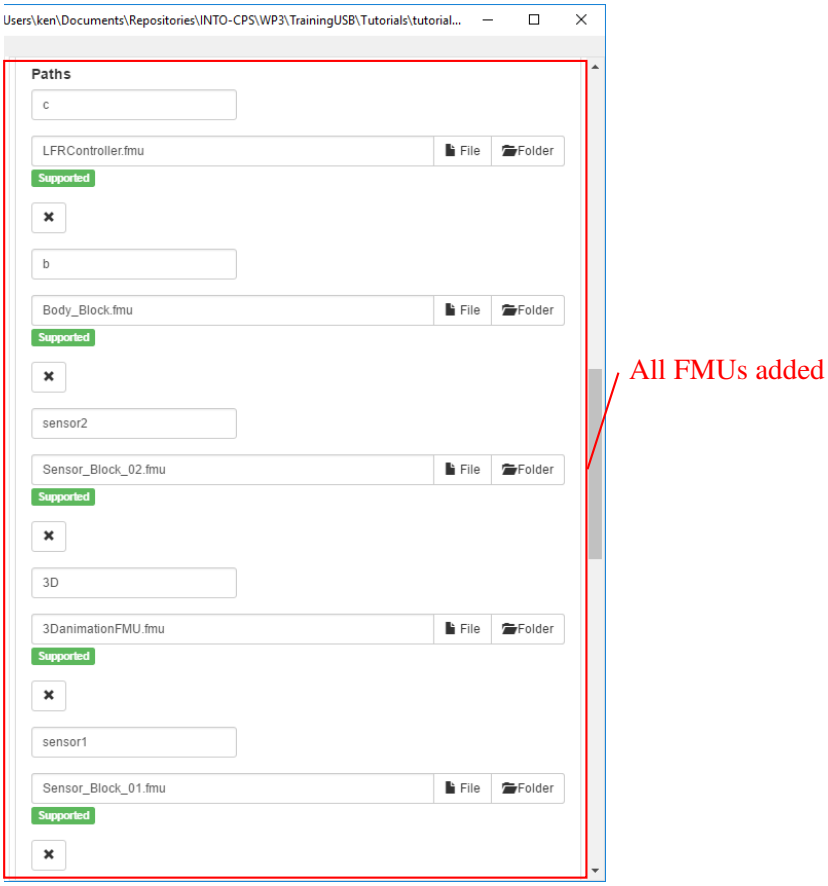


Step 11. A file browser window will open and show five FMUs (if the file browser does not show the FMUs, navigate to *tutorials_3/FMUs*). Select *FMUController.fmu* and click *Open*.



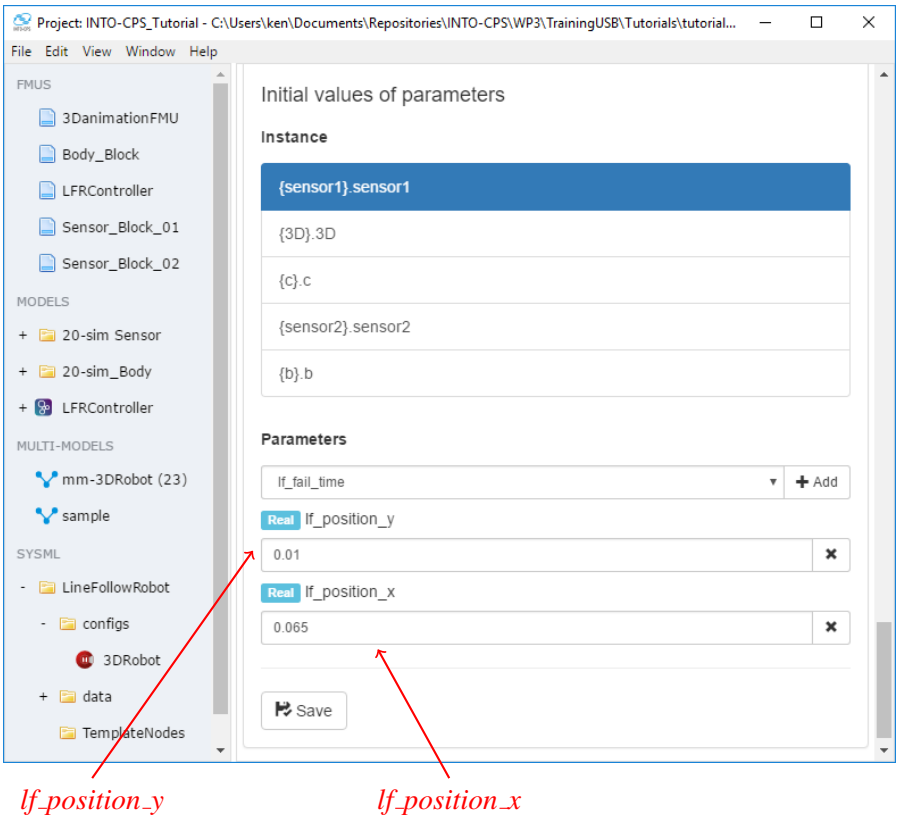
Step 12. The LFRController has been added. Repeat this for the remaining elements:

- *b* : *Body_Block.fmu*
- *3D* : *3DanimationFMU.fmu*
- *sensor1* : *Sensor_Block_01.fmu*
- *sensor2* : *Sensor_Block_02.fmu*



Step 13. Next the sensor positions must be defined. Scroll down to the *Initial values of parameters section*, and click {*sensor1*}.*sensor1*. In the *Parameters* section, enter the following values:

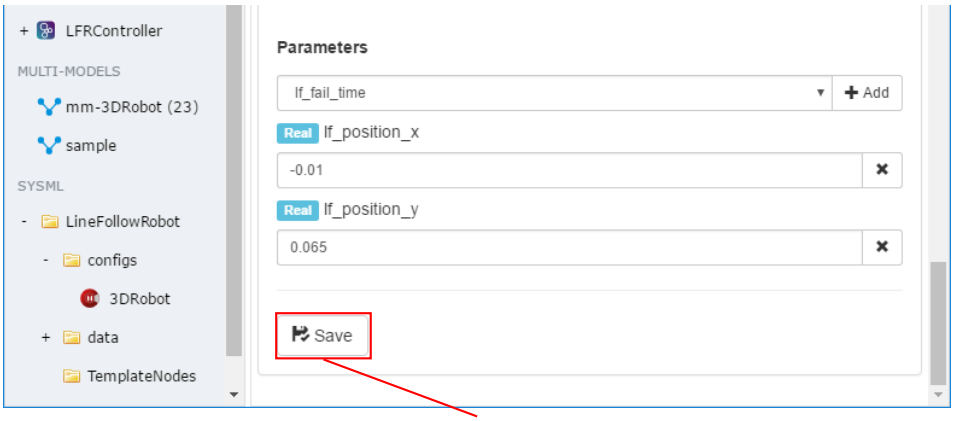
- *lf_position_y* = 0.065
- *lf_position_x* = 0.01



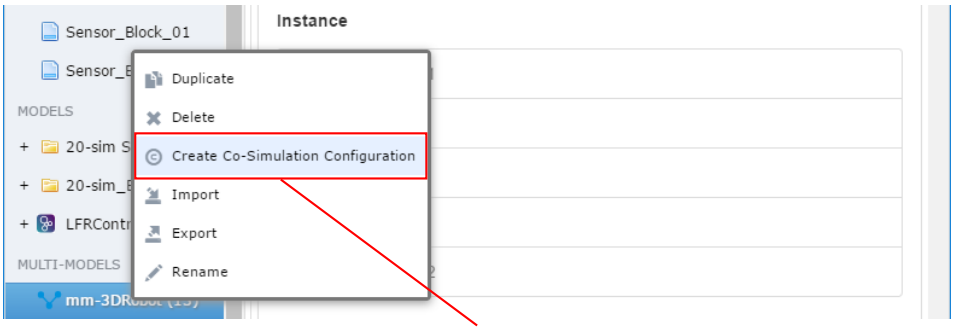
Step 14. Repeat the previous step for the second sensor – {*sensor2*}.*sensor2* with the following values:

- *lf_position_y* = -0.01
- *lf_position_y* = 0.065

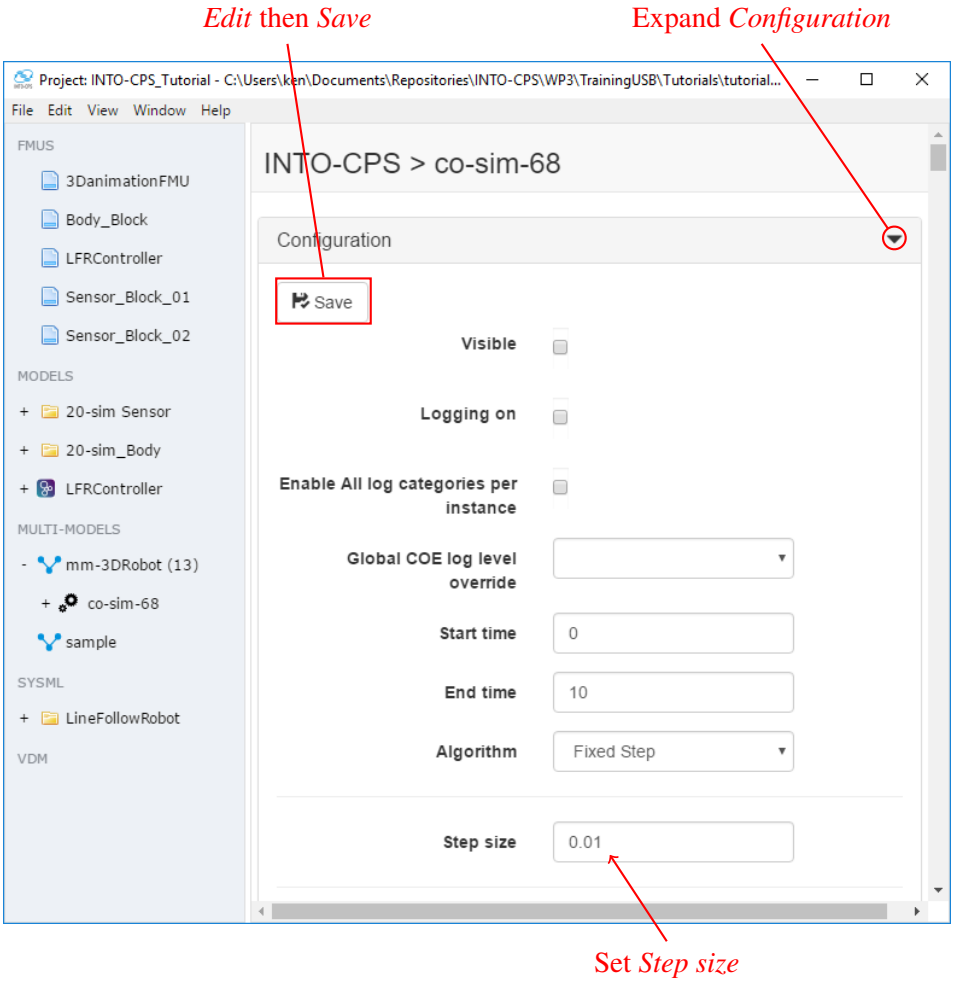
Step 15. *Save* the *Configuration*.



Step 16. The multi-model configuration is complete. Right-click on the multi-model configuration and select *Create Co-simulation Configuration*.



Step 17. Set the *Step size* to 0.01. Don't forget to press *Edit* then *Save*.

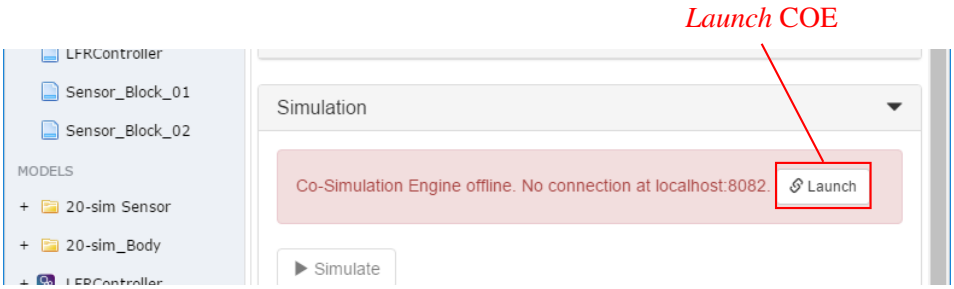


Step 18. You may also *lf_1_sensor_reading* from {*sensor1*}.*sensor1* and {*sensor2*}.*sensor2* to the *Live-stream Configuration* if you want.

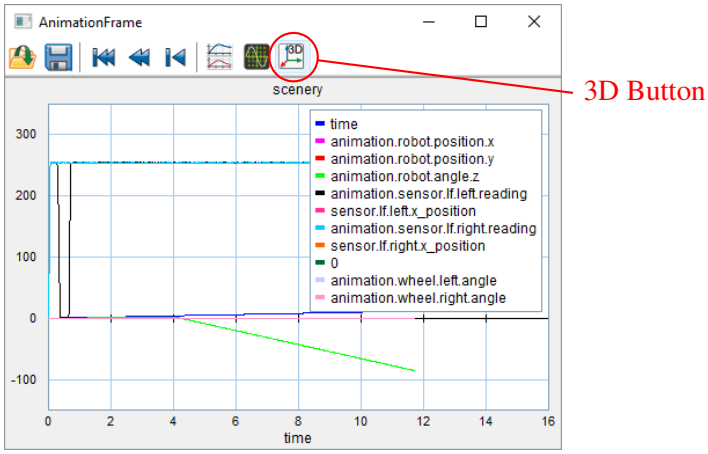


3 Running a Co-simulation

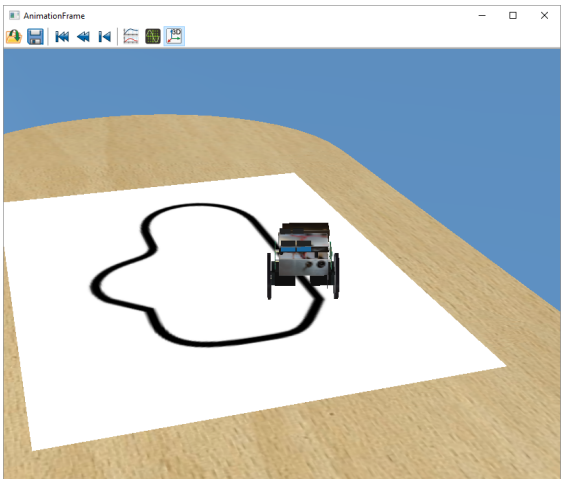
Step 19. Launch the COE if necessary (see *Tutorial 1 — First Co-simulation* for a reminder if needed).



Step 20. When the COE is running (see *Tutorial 1* for more details if you need a reminder), click the *Simulate* button. After a few seconds, a Java window called *Animation Frame* will appear like the one below. It shows a plot of variables from the co-simulation. You can click the *3D* button to see the 3D visualisation of the robot.



Step 21. A 3D model of the line following robot will appear. This view may be changed by clicking and dragging the mouse (note this is currently quite sensitive, so dont make quick movements). When the simulation has finished, this window will close. If everything went well, the robot should follow the line.



4 Additional Exercises

When this tutorial is complete, either move onto Tutorial 3, or try the following additional exercises:

1. Experiment with different sensor positions. Repeat steps 13 and 14 to change the position of the left and right sensors relative the the robot body. How do the different values effect the simulation?

2. Experiment with different robot speeds. Repeat steps 13 and 14 to change the different speed values for the *Controller*. How do the different values effect the simulation?