Tutorial 2 — Adding FMUs

Overview

This tutorial will show you how to:

1. Edit a multi-model configuration
2. Add a new FMU to a multi-model configuration
3. Execute a co-simulation using the new multi-model configuration

Requirements

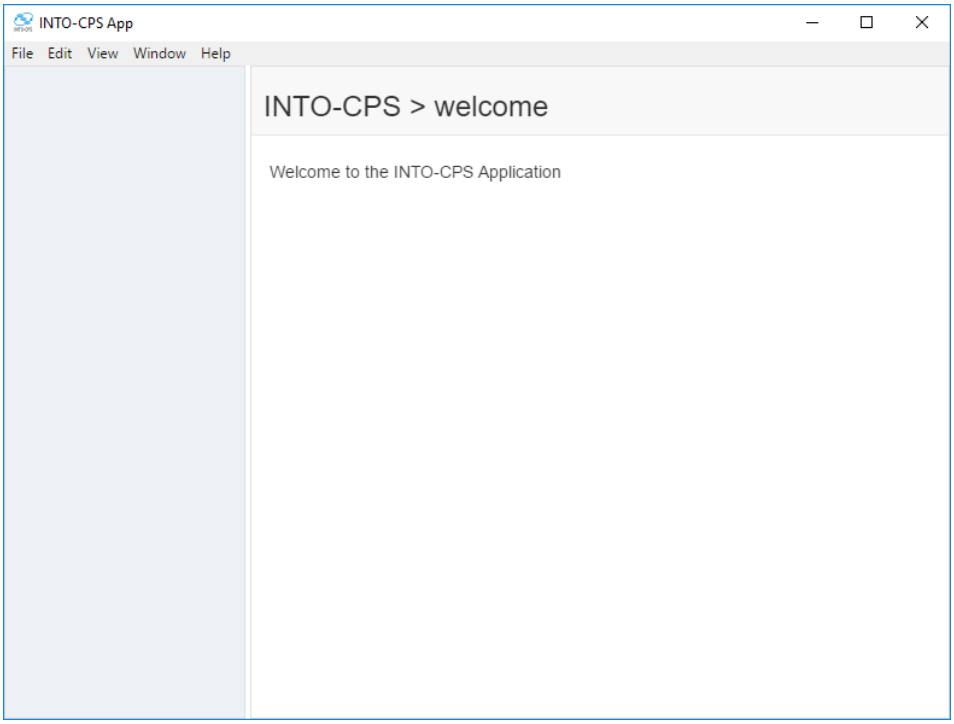
This tutorial requires the following tools from the INTO-CPS tool chain to be installed:

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

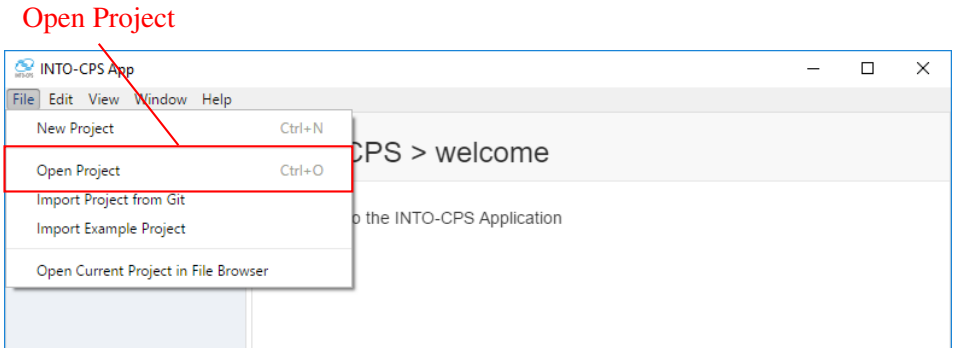
You may have been provided with tools on a USB drive at your training session. Otherwise the INTO-CPS Application can be downloaded from `https://into-cps.github.io/download/` and tools can be downloaded from there through *Window* > *Show Download Manager* to your *into-cps-projects* install downloads directory. Please ask if you are unsure.

1 Opening a Project

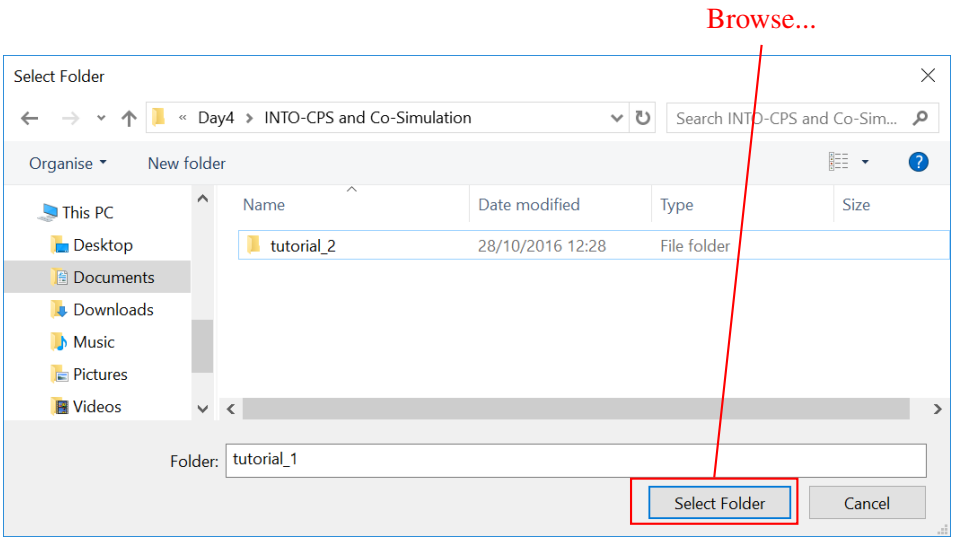
Step 1. Launch the *INTO-CPS Application*. On first loading, it will look like the screenshot below. If you have opened a project previously, that project will be opened automatically.



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



Step 3. Find *Tutorials/tutorials_2*, select it and press *Select Folder*.



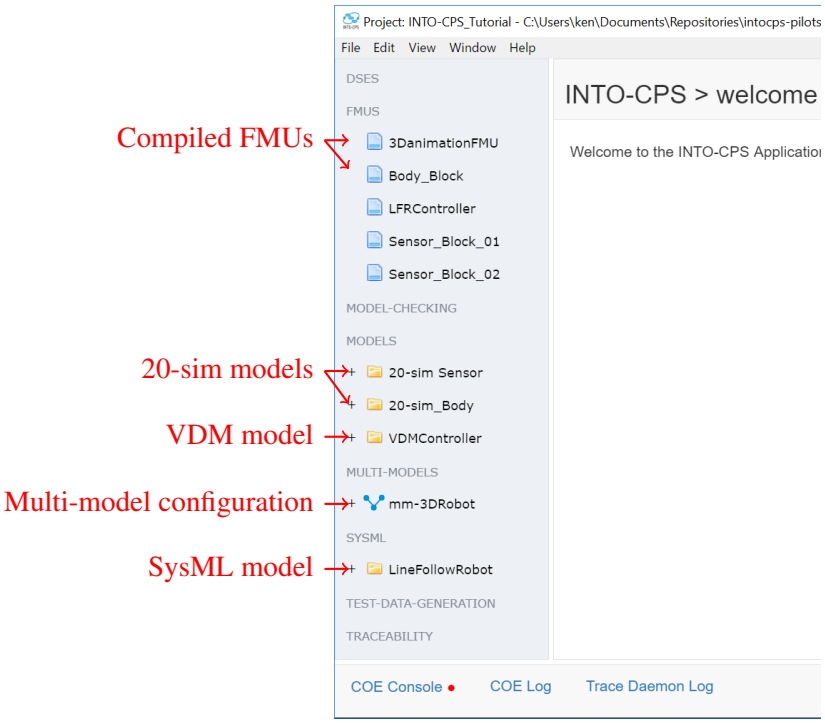
Step 4. Once the project is opened, you will see that project browser on the left of the INTO-CPS Application window is now populated. The entries in the project browser correspond to folders and files in the *Tutorials/tutorials_2* folder. These are:

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.

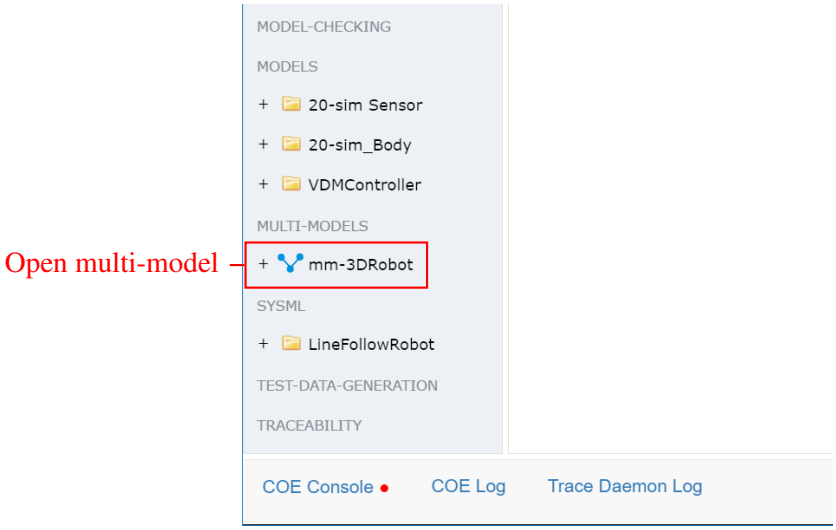


This multi-model is a line-following robot, which contains a model of the body (including wheels and motors), a model of the sensors, and a controller which reads the sensors and controls the motors to make the robot follow the line.

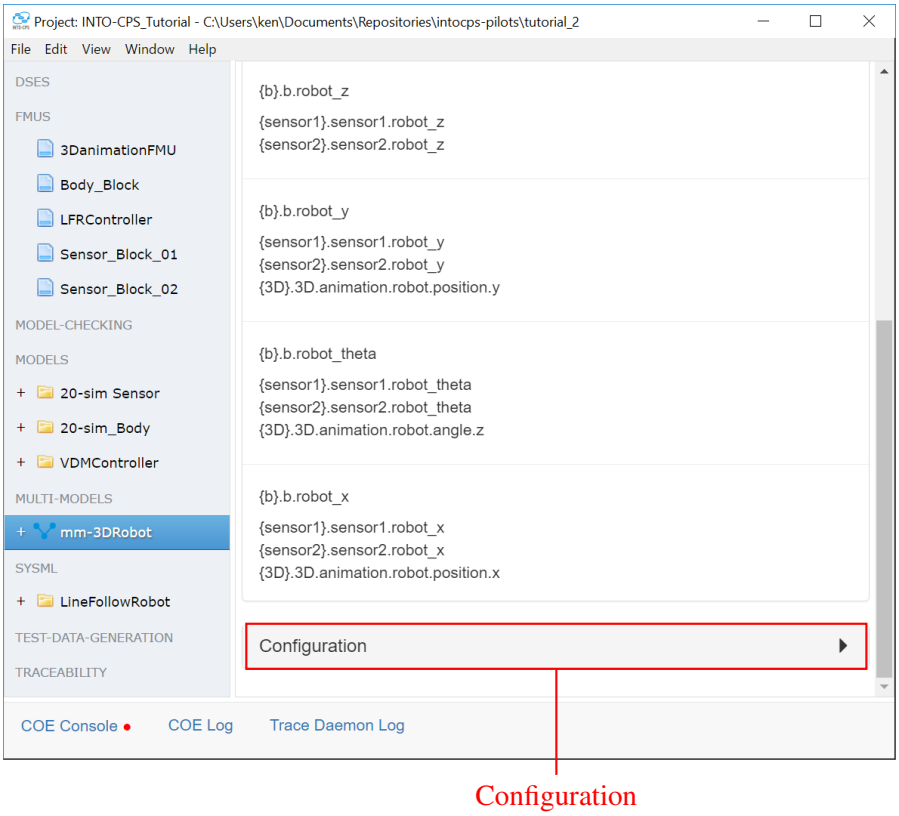
2 Editing a Multi-model

In this tutorial, the controller FMU has not been added to the multi-model, and multi-model parameters are missing. We will add the FMU, set the parameters, then run a co-simulation.

Step 5. Double-click the *mm-3DRobot* to open it.



Step 6. Click on *Configuration* to expand it.



In the *Overview* part an overview of the already configured connections is provided. This corresponds to the blue arrows in Figure 1.

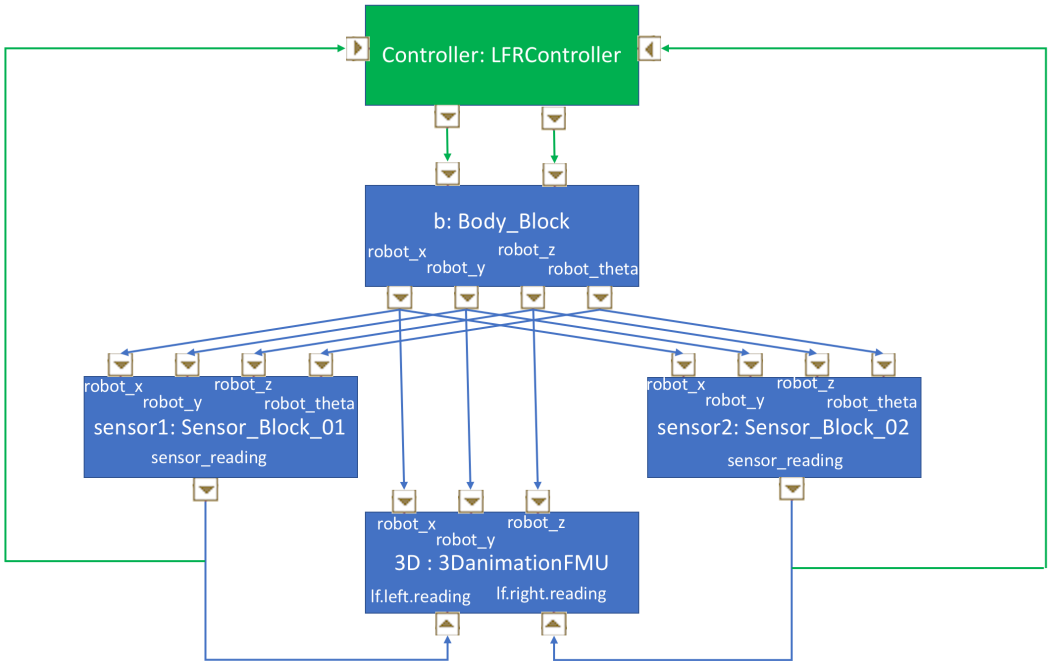
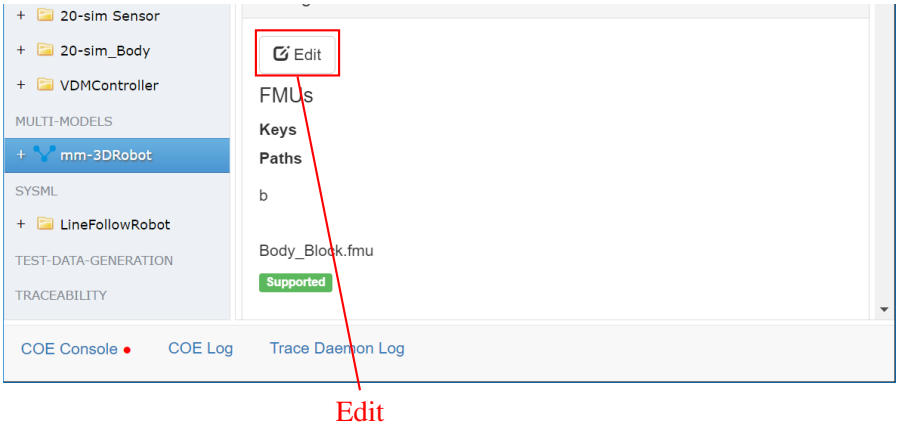


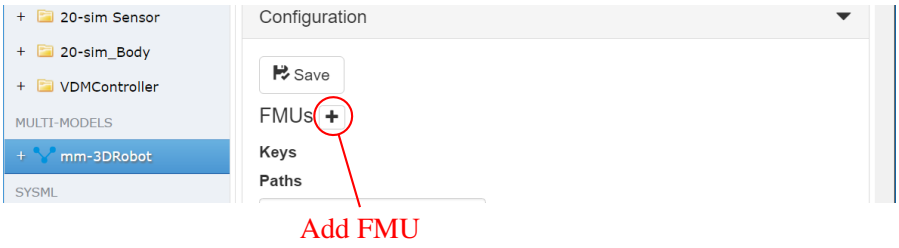
Figure 1: Initial connections with blue, new ones to configure with green

The green box and arrows in Figure 1 are what you need to configure in the steps below.

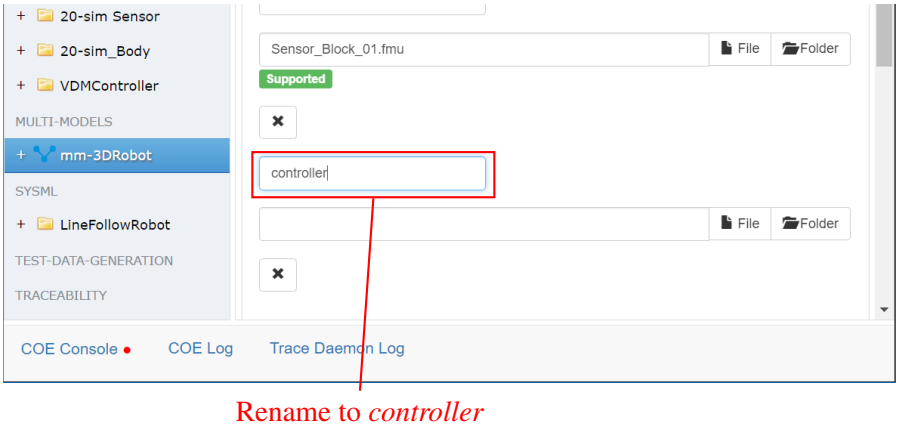
Step 7. Click *Edit*.



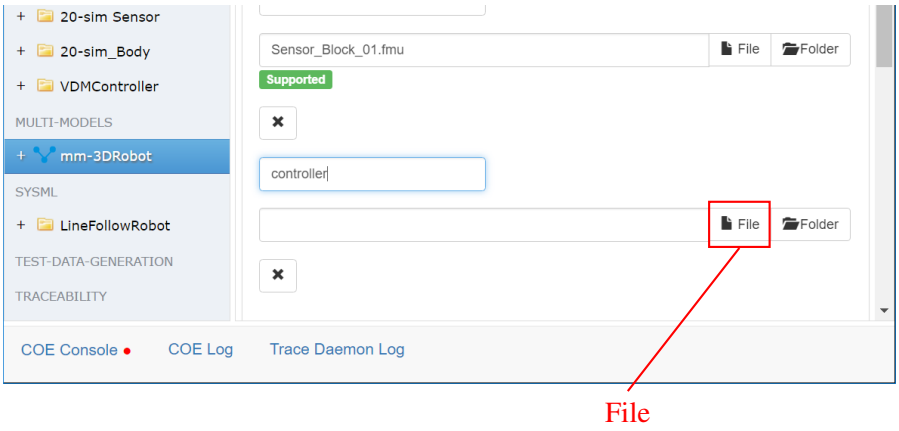
Step 8. Click the + icon to add a new FMU entry.



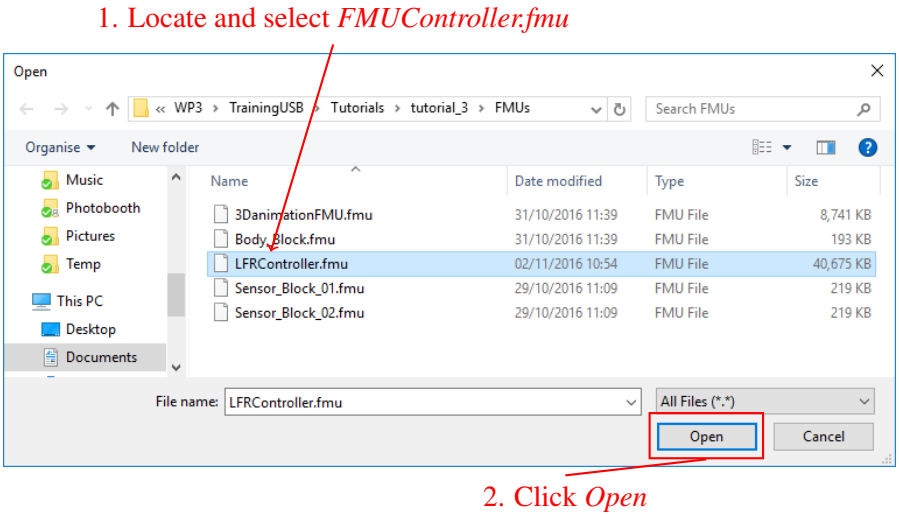
Step 9. The new entry is named *FMU*. Rename it to *controller*.



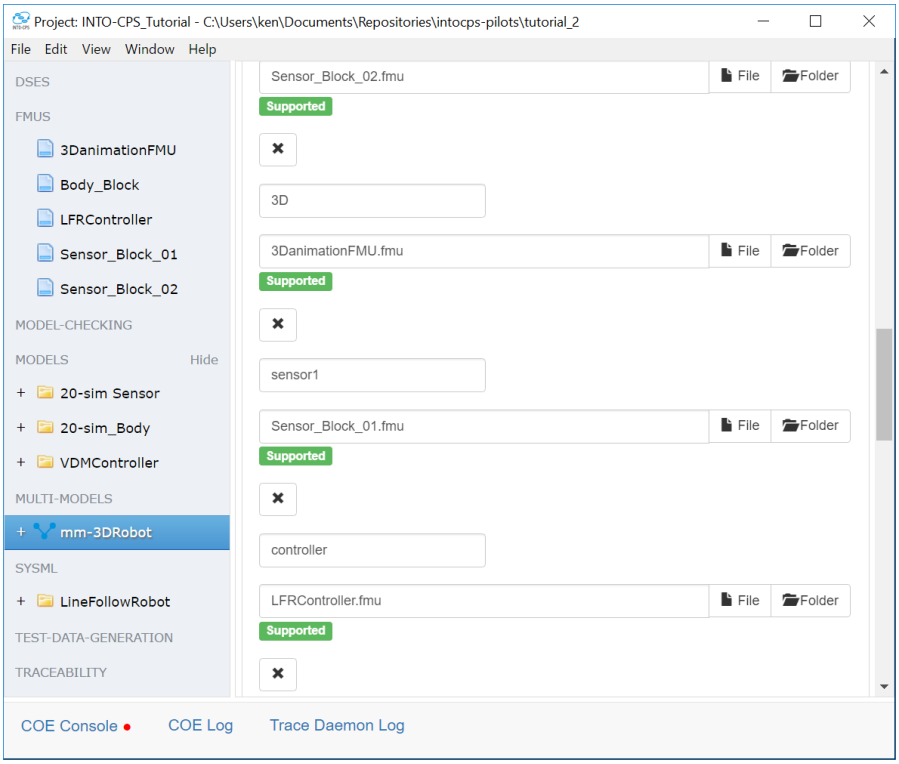
Step 10. We need to associate an FMU with this entry. To do this, click the *File* button.



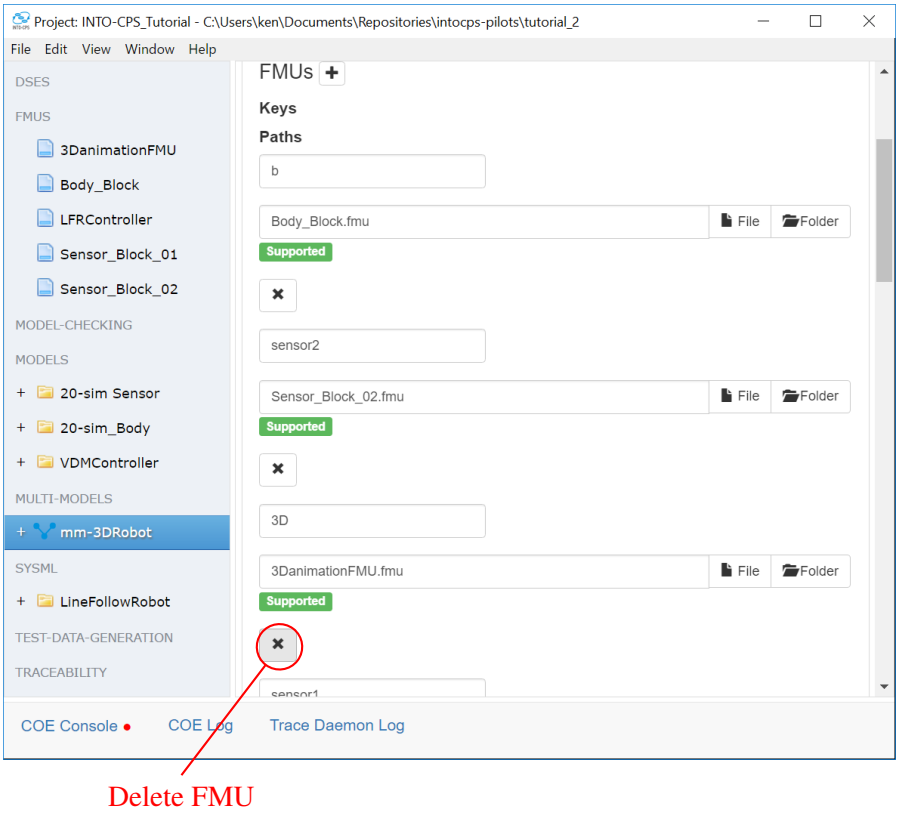
Step 11. A file browser window will open and show five FMUs (if the file browser does not show the FMUs, navigate to *tutorials_2/FMUs*). Select *FMUController.fmu* and click *Open*.



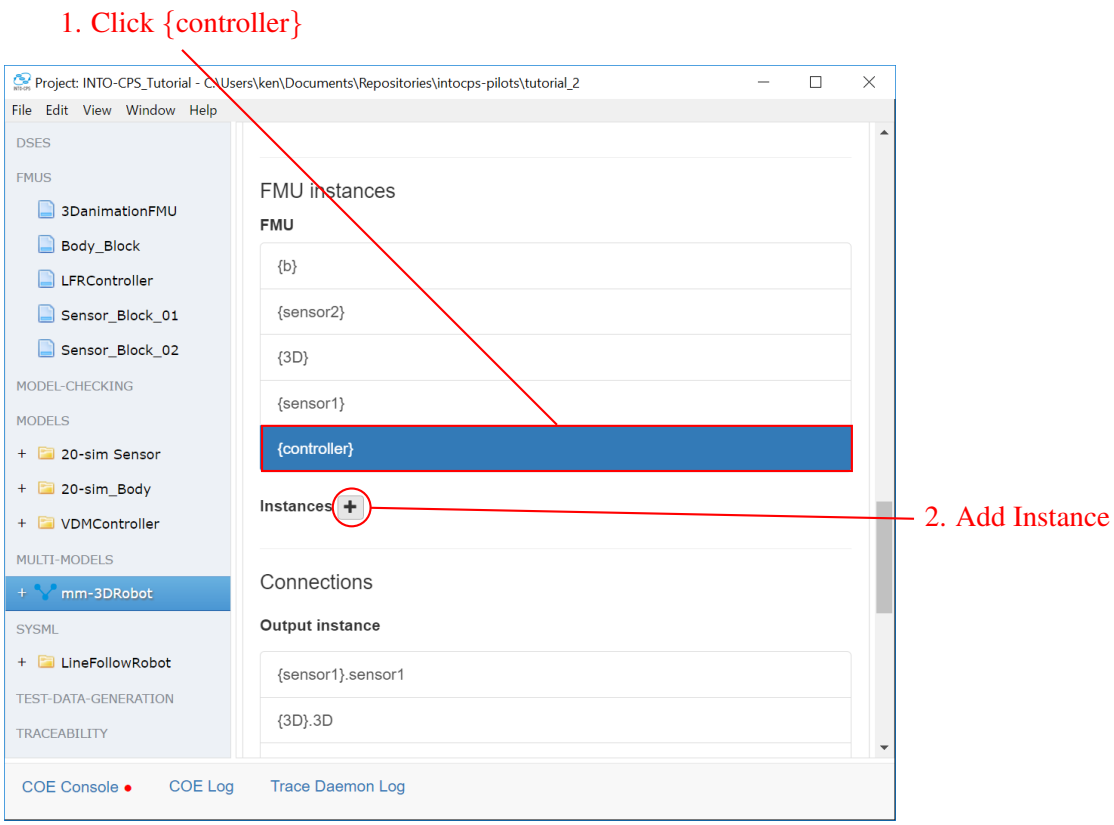
Step 12. The controller FMU is now associated.



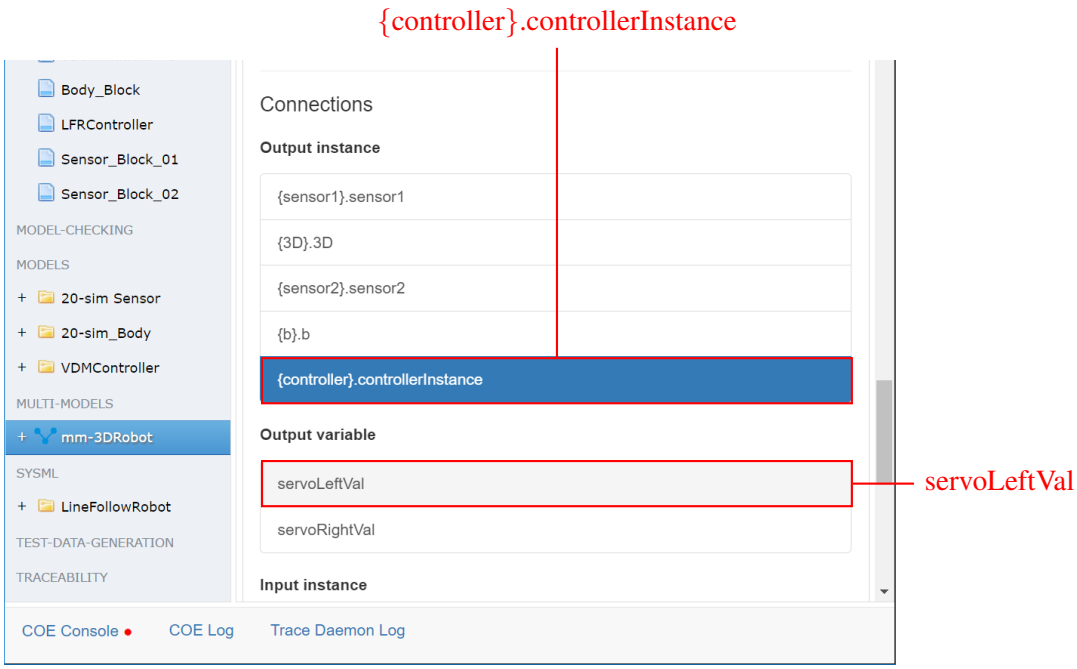
Step 13. **MacOS X / Linux Only:** As in the first tutorial, the 3D visualisation FMU from this multi-model is only supported on Windows. Therefore delete it using the *X* button.



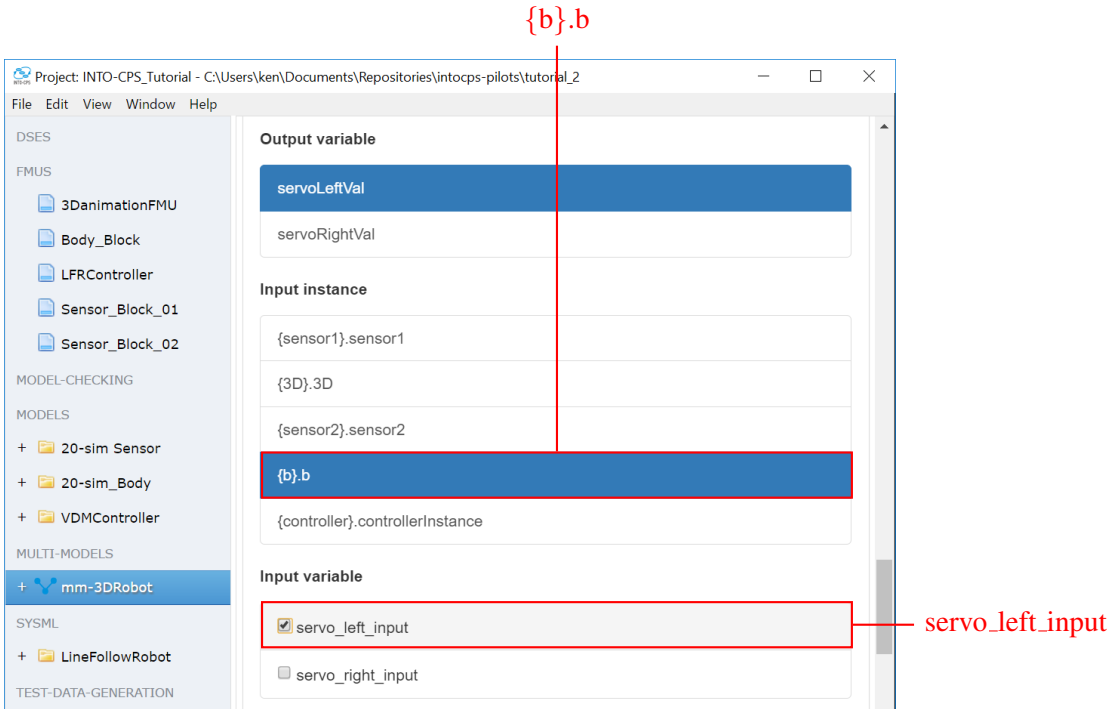
Step 14. Next we must add an instance of the controller. Under *FMU Instances* click {*controller*} then click the + icon.



Step 15. Next the controller outputs must be connected to the body inputs (to control the motors). Scroll down to *Connections*. Under **Output instance** click on {*controller*}.*controllerInstance* and under **Output variable** select *servoLeftVal*.

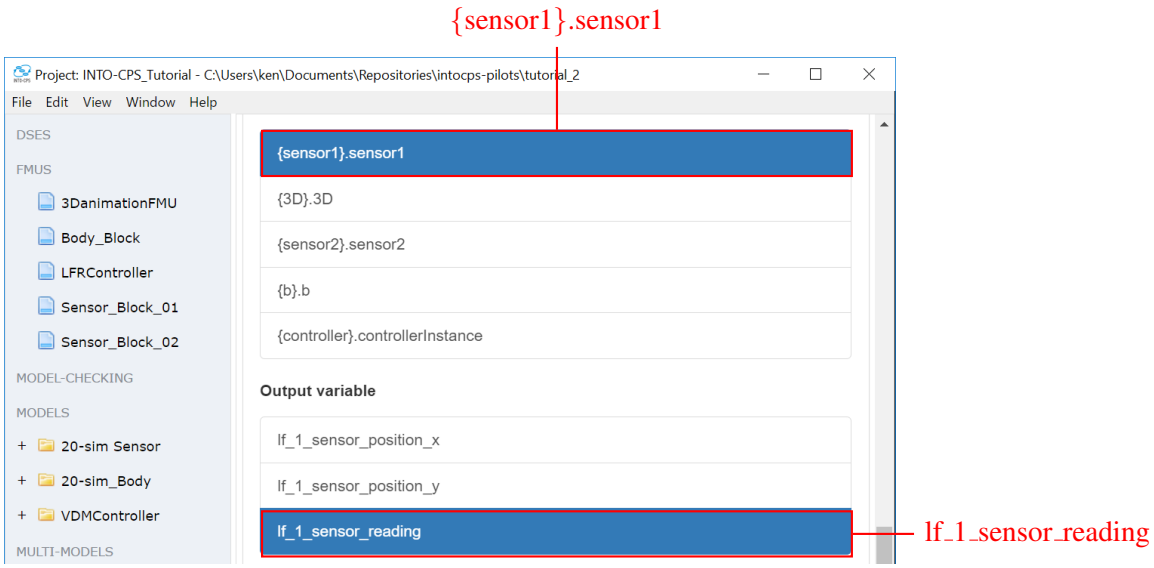


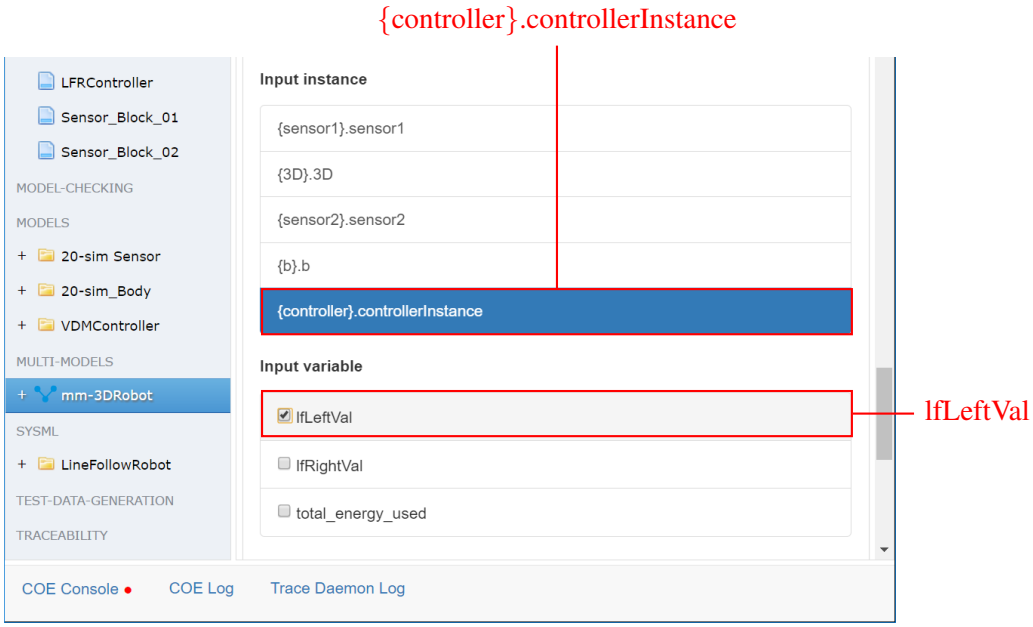
Step 16. Then under **Input instance** select {*b*}.*b* and under **Input variable** check *servo_left_input*.



Step 17. Repeat the previous step to connect {*controller*}.*controllerInstance / servoRightVal* to {*b*}.*b / servo_right_input*.

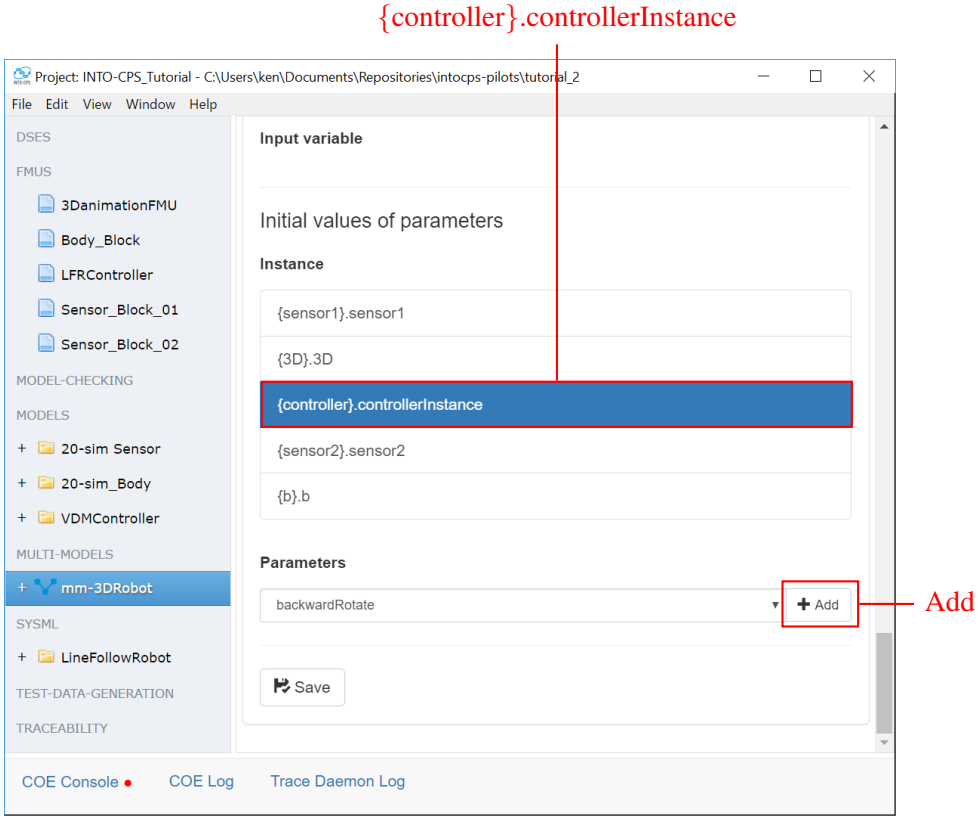
Step 18. Then repeat to connect {*sensor1*}.*sensor1 / lf_1__sensor_reading* to {*controller*}.*controllerInstance / lfLeftVal*.



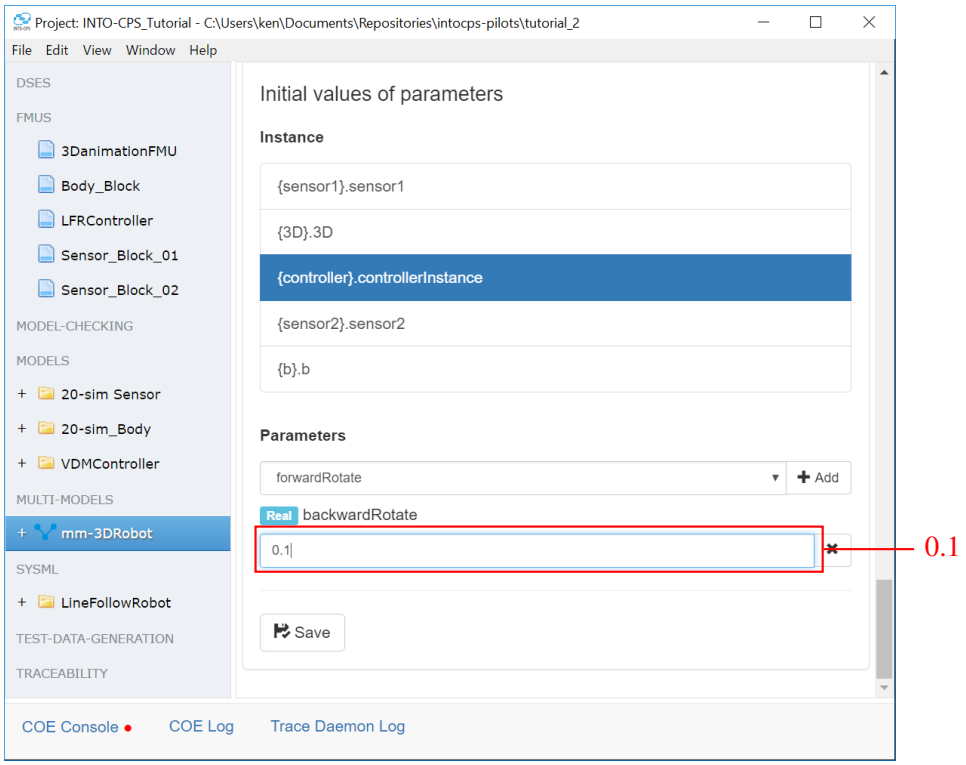


Step 19. Finally, repeat to connect {*sensor2*}.*sensor2 / lf_1__sensor_reading* to {*controller*}.*controllerInstance / lfRightVal*.

Step 20. Next we must set the parameters of the controller, which determines how it responds to sensor input. Scroll down to *Initial values of parameters* and select {*controller*}.*controllerInstance*. Then click + *Add* for *backwardRotate*.

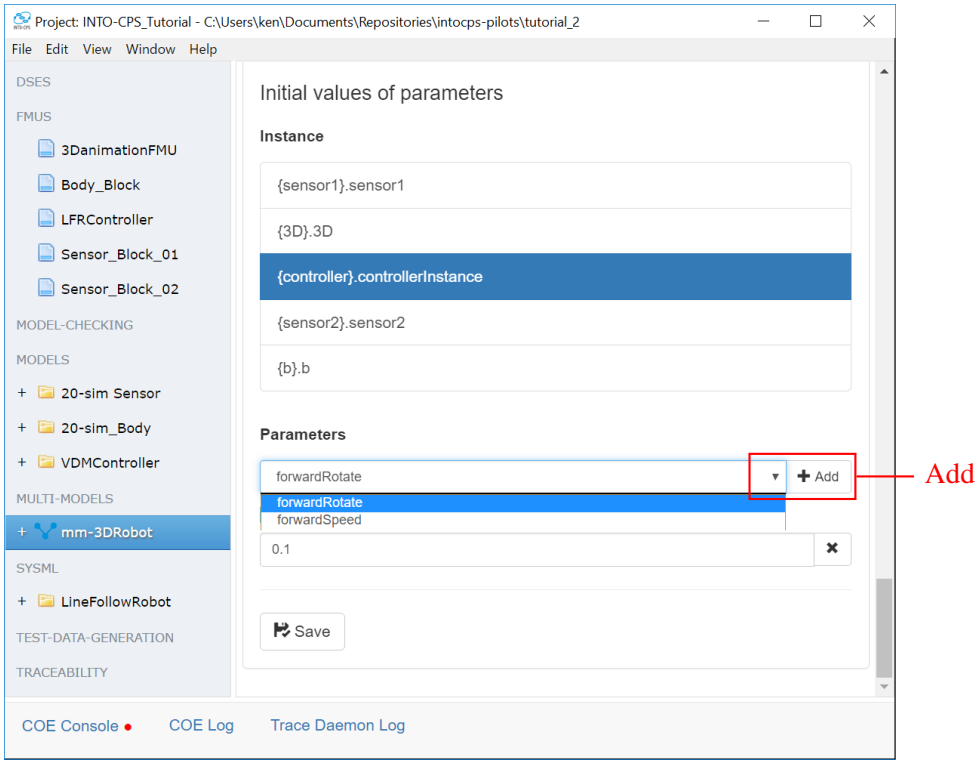


Step 21. Set *backwardRotate* to *0.1*.

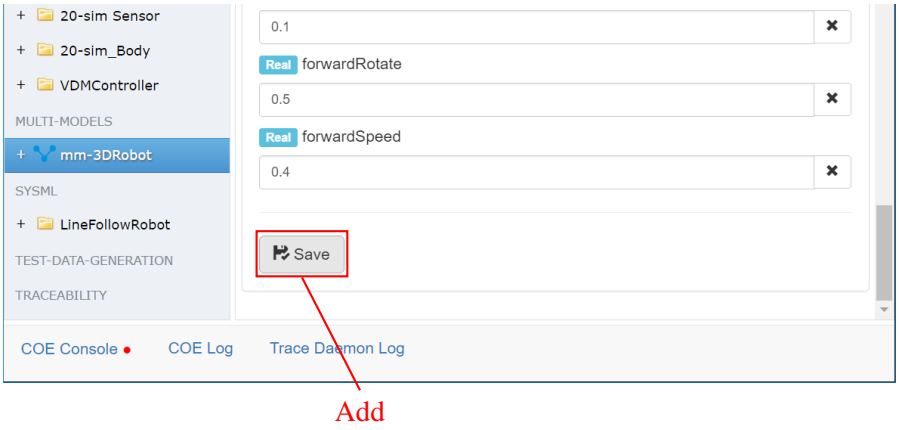


Step 22. Use the drop-down box and + *Add* button to add:

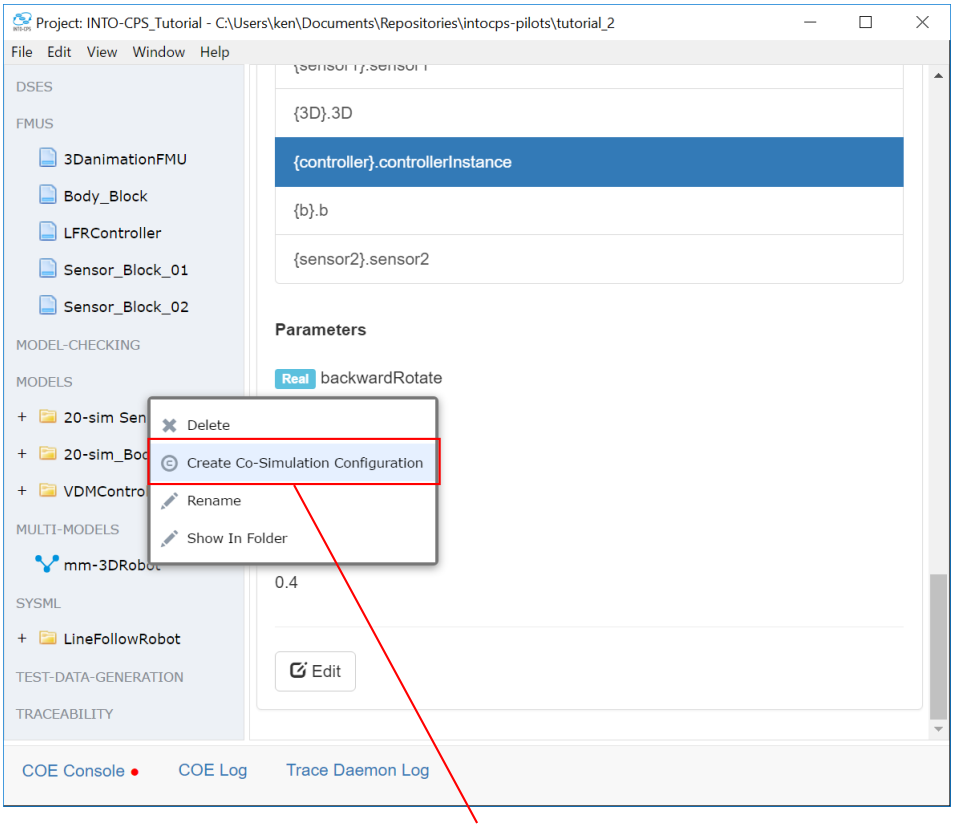
- *forwardRotate* with a value of *0.5*.
- *forwardSpeed* with a value of *0.4*.



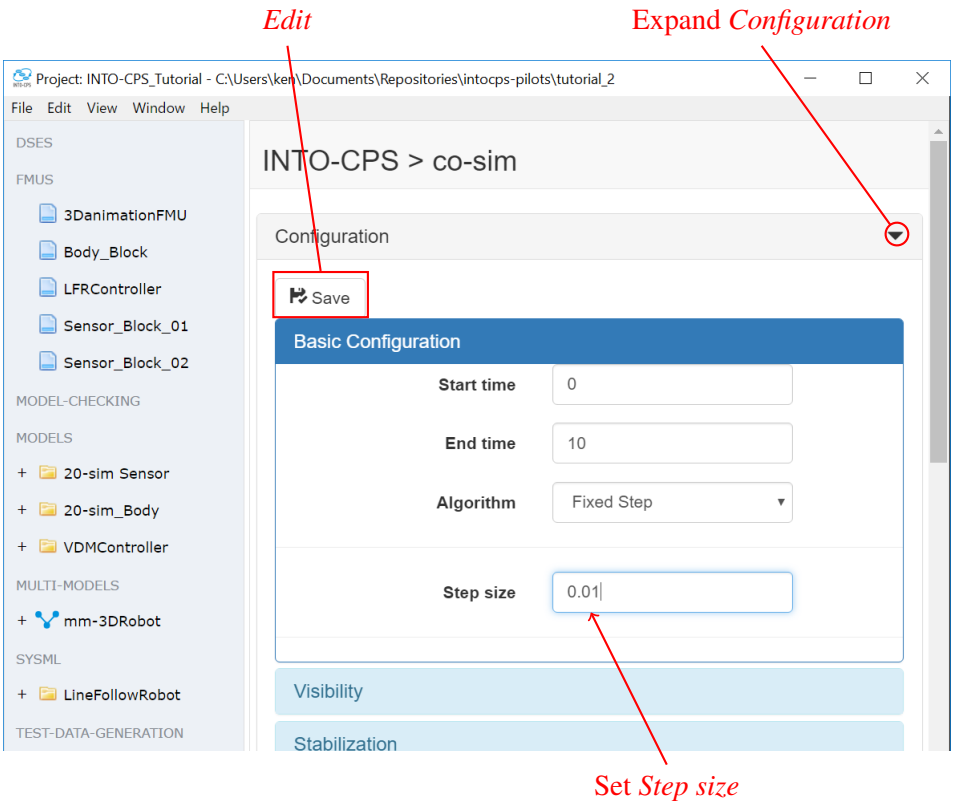
Step 23. *Save* the *Configuration*.



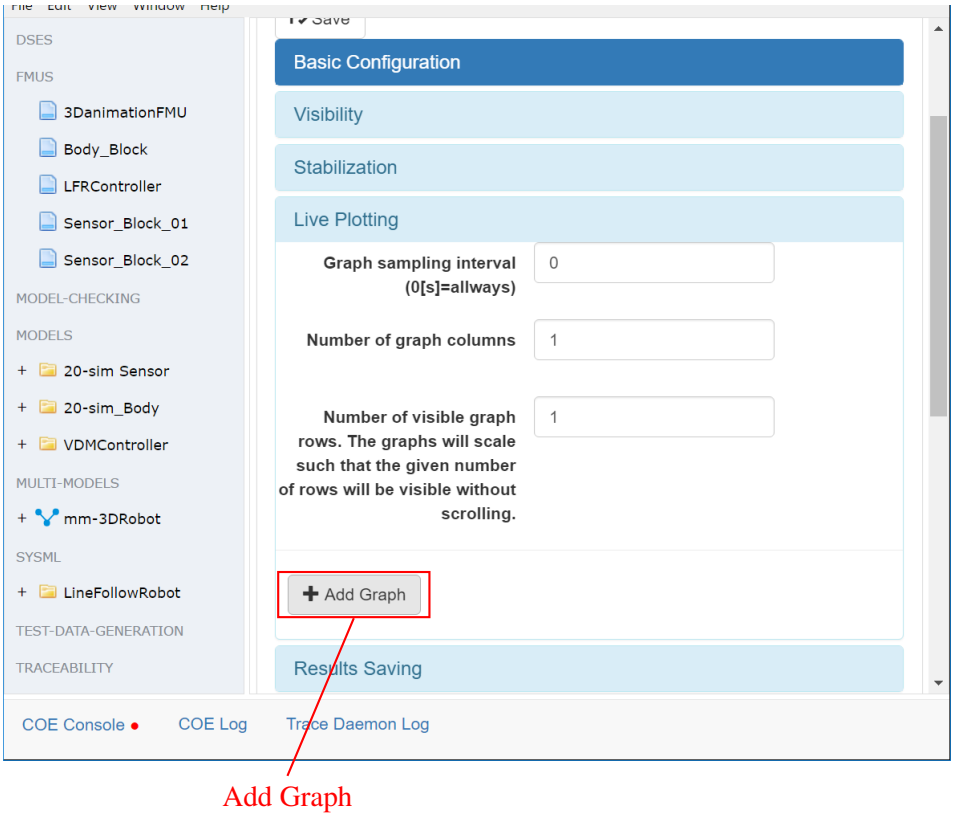
Step 24. The multi-model configuration is complete. Right-click on the multi-model configuration and select *Create Co-simulation Configuration*. You can use the default name, *co-sim*, or choose your own name.



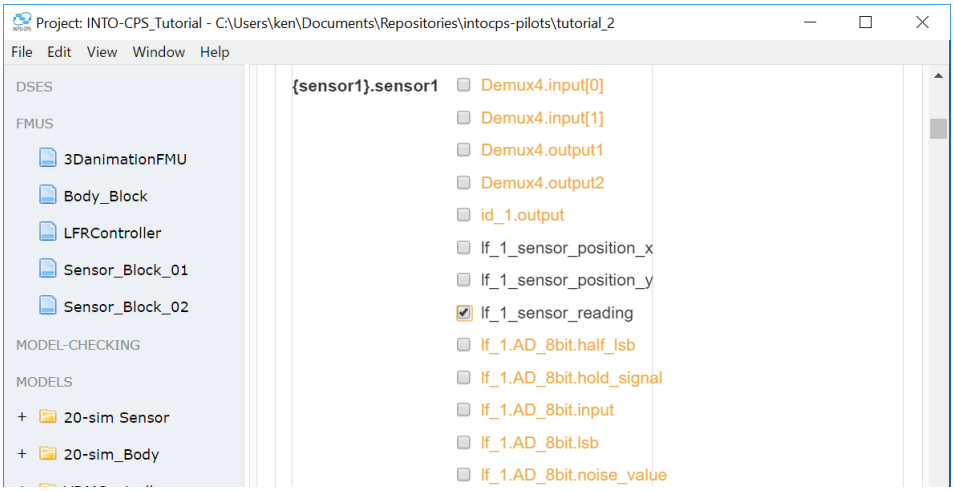
Step 25. Under *Basic Configuration* set the the *Step size* to 0.01. Don't forget to press *Edit*.



Step 26. Under *Live Plotting* click *Add Graph*.

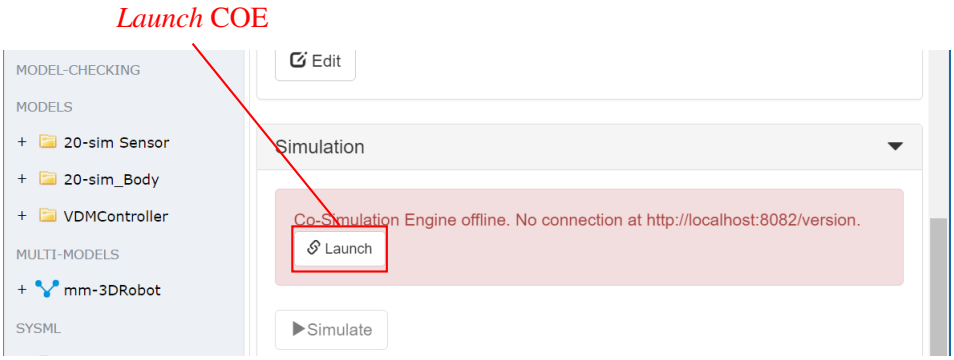


Step 27. Check *lf_1_sensor_reading* from {*sensor1*}.*sensor1* and {*sensor2*}.*sensor2* to see the sensor values appear in the *Livestream Configuration*.

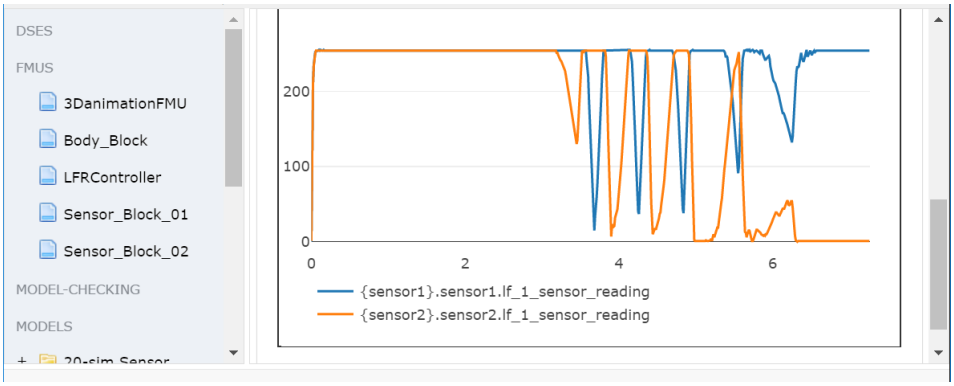


3 Running a Co-simulation

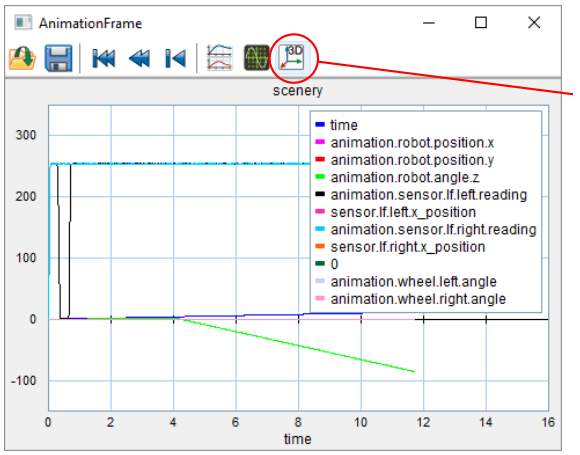
Step 28. Launch the COE if necessary (see *Tutorial 1 — First Co-simulation* for a reminder if needed).



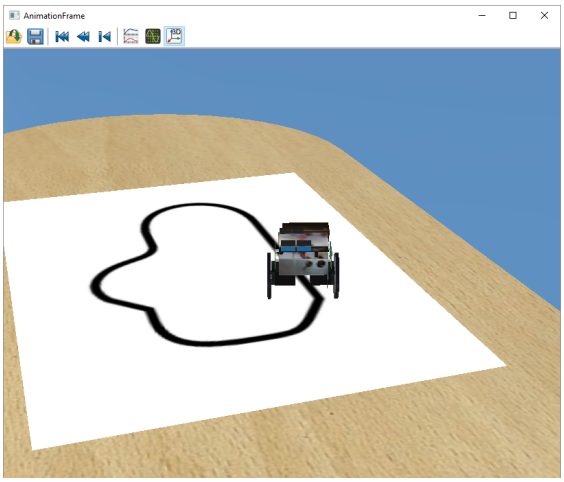
Step 29. When the COE is running (see *Tutorial 1* if you need a reminder), click the *Simulate* button. The graph should show blue and orange lines, and if these should move up and down, indicating that the robot is sweeping left and right, following the line.



Step 30. **Windows only:** During co-simulation, 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 31. **Windows only:** 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 don't 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?