



OpenIPSL

A Modelica Library for Power Systems Simulation

Assoc. Prof. **Luigi Vanfretti**

luigiv@kth.se

<https://www.kth.se/profile/luigiv/>

Preparatory work – aka *homework!*

Please follow these slides before taking part in the Workshop/Tutorial/Seminar.



Requirements



Requirements for the workshop are:

We have only tested our tutorial for the following configurations.

Windows:

- PC with installed Windows 7 or later
- Installation of OpenModelica

Mac:

- OSX El Capitan
- Installation of OpenModelica (binaries!)
- Xcode (Version 8.0)



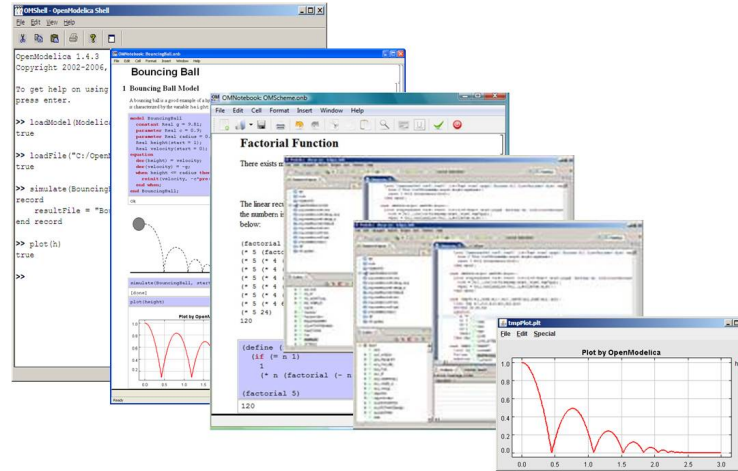
Getting Started with OpenModelica and OpenIPSL

This presentation is a 2-part guide containing the preparatory work needed to carry out the *hands-on examples* of the Modelica and OpenIPSL workshop/tutorial/seminar.

- Part 1: Setting up **OpenModelica**



- Part 2: Setting up OpenIPSL



Part 1

Setting up OpenModelica



Installation of OpenModelica

Instructions:

- Download the installation package
 - Windows:
 - <https://www.openmodelica.org/download/download-windows>
 - 1.9.6: <https://build.openmodelica.org/omc/builds/windows/releases/1.9.6/OpenModelica-v1.9.6.exe>
 - 1.9.11:
 - <https://build.openmodelica.org/omc/builds/windows/releases/1.11.0/>
 - Mac:
 - 1.9.6: <https://build.openmodelica.org/omc/builds/mac/binaries/latest-release-1.9.6.mpkg>
- Launch the Installation package and follow the instructions with default options

Note!

Compatibility with OpenIPSL is checked for OpenModelica versions 1.9.6 (Mac and Windows) and 1.9.11 (on Windows)

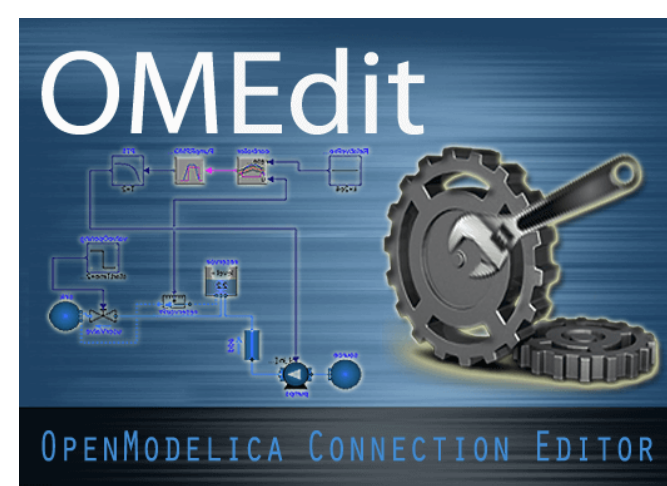
No MAC OSX or Windows OS – No Problem!

- OpenModelica is available for GNU/Linux distributions here:
 - <https://www.openmodelica.org/download/download-linux>
 - **Note:** the compatibility of OpenIPSL has not been tested under these OS distributions
- Virtual Machine:
 - OpenModelica can be installed through pre-built Virtual Machines containing all the libraries and clients that come with OpenModelica.
 - See instructions here:
 - <https://www.openmodelica.org/download/virtual-machine>
 - **Note:** the compatibility of OpenIPSL has not been tested under these VM configurations.





Check of OpenModelica



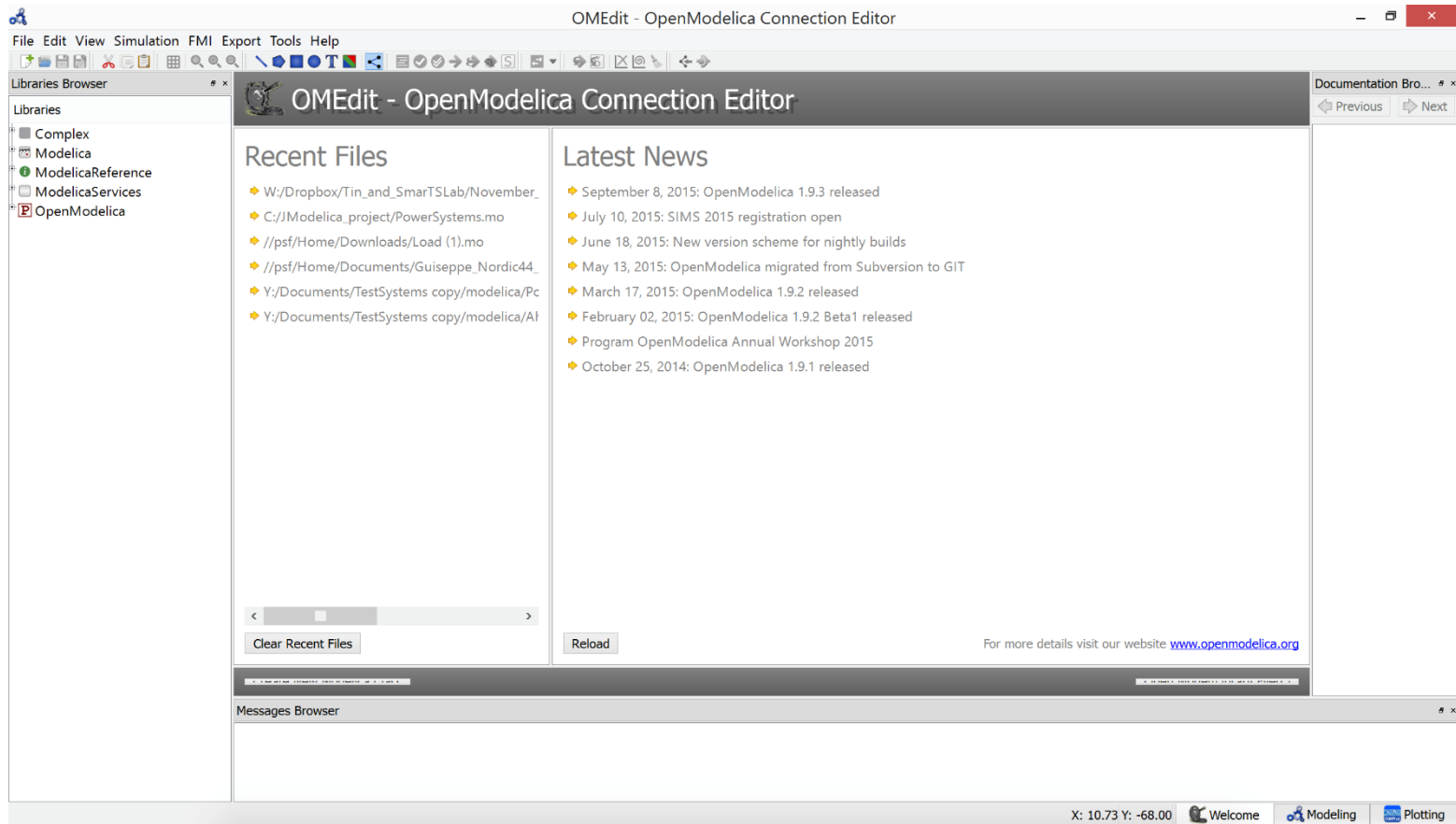
Tasks to check OpenModelica is correctly installed on your computer:

- Start OpenModelica Connection Editor (OMEdit)
- In the Libraries Browser navigate to Modelica.Blocks.Examples.PIDController
- Select Runge Kutta as a solver and simulate the model
- In the “Plotting” view, plot variable speedSensor.w



Check of OpenModelica – Step 1

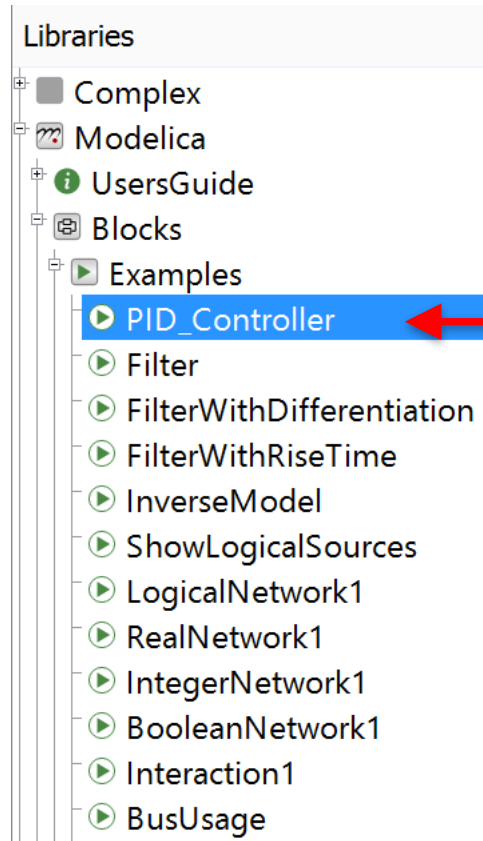
Upon launch, the Connection Editor will present the following window





Check of OpenModelica – Step 2

Browse the Modelica library to find the PID_Controller and open it



Double-click to
open

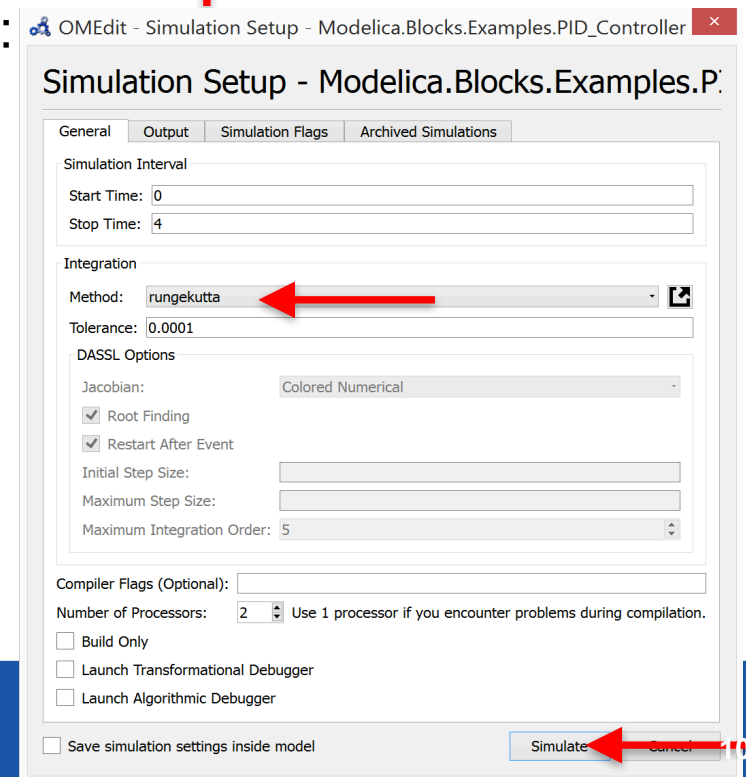
Check of OpenModelica – Step 3

Simulation settings are accessed on the toolbar:



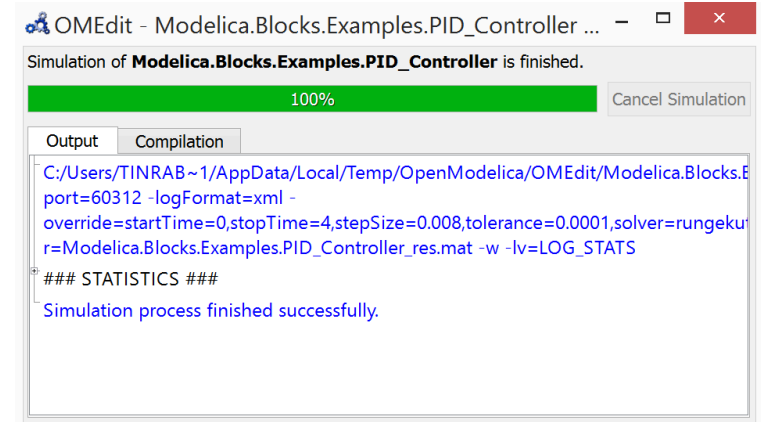
Adjust the settings to match the followings :

Click on **Simulate** to launch the simulation

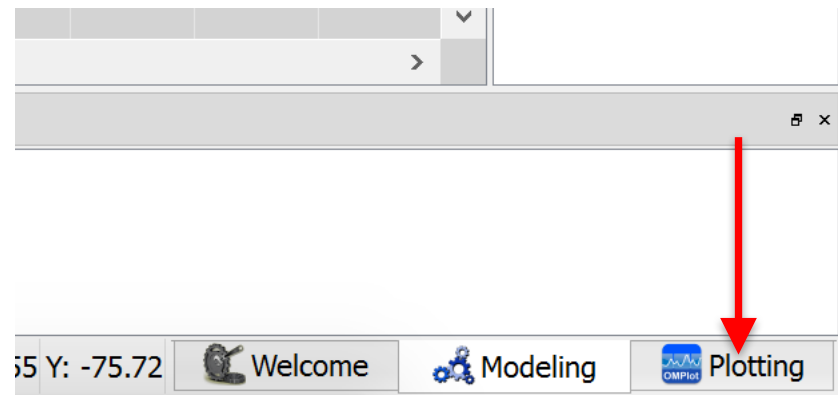


Check of OpenModelica – Step 4a

Once the simulation is completed (100 %):



- Access the plotting facility by clicking on the **Plotting** tab in the lower right corner of the screen





Check of OpenModelica – Step 4b

In the plotting facility, browse the variable to find the rotational speed w

Variables Browser

Find Variables

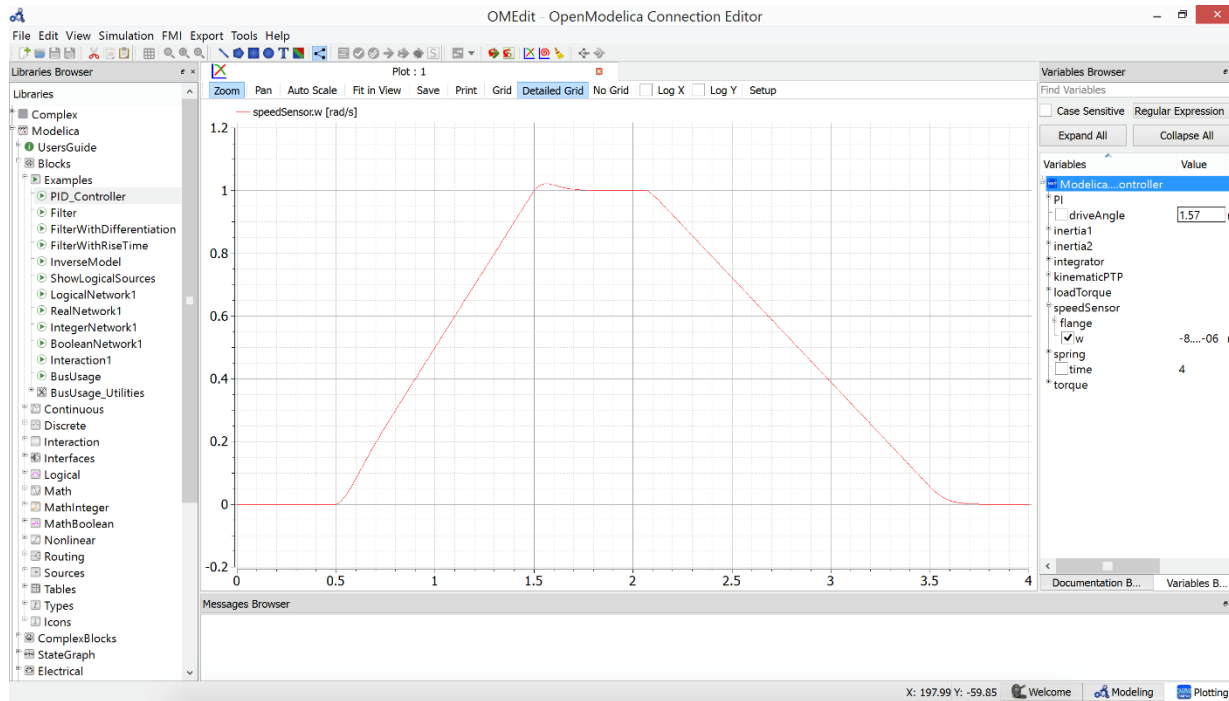
Case Sensitive Regular Expression

Expand All Collapse All

Variables	Value	Unit
Modelica...ontroller		
PI		
<input type="checkbox"/> driveAngle	1.57	rad
inertia1		
inertia2		
integrator		
kinematicPTP		
loadTorque		
speedSensor		
flange		
<input checked="" type="checkbox"/> w	-8.00	rad/s
spring		
<input type="checkbox"/> time	4	
torque		



Check of OpenModelica – Final Result



If your screen looks like this, you're ready to go!



Part 2

Setting up OpenIPSL



Download OpenIPSL!



Go to our Github repo:

https://github.com/SmarTS-Lab/OpenIPSL/releases/tag/Tuto_UCD_2017

The screenshot shows a GitHub release page for the repository 'Workshop on Dynamic Systems Modeling @UCD'. The release is labeled 'Pre-release' and is titled 'Tuto_UCD_2017' with commit hash 'af38070'. The release was made by 'MaximeBaudette' 13 days ago, with 8 commits since. A pull request #103 from 'tinrabuzin/OpenCPSD5d3B' is mentioned, with the description: 'Adding resynchronisation models developed by Tin Rabuzin.' Below this, it states: 'These are the new models for distribution network re-synch simulation built by Tin Rabuzin as part of the OpenCPS project, and reported in deliverable D5.3B.' Under the 'Downloads' section, there are two buttons: 'Source code (zip)' and 'Source code (tar.gz)'. A large blue button with the text 'Click Here!' is overlaid on the 'Source code (zip)' button.

Note: A dedicated package will be prepared for the tutorial and uploaded soon. Please download (again!) the package on the day of the tutorial so that you have the most up to date files.

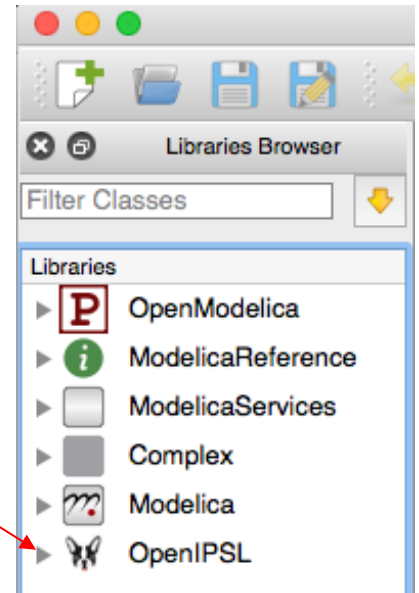
The dedicated package will also be available on a USB stick that we can circulate on the day of the tutorial.



Load the OpenIPSL to OMEdit

External libraries, e.g. OpenIPSL, must be loaded in OMEdit to be used:

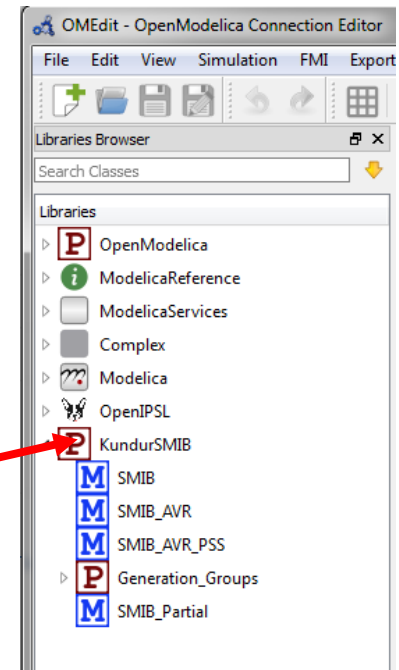
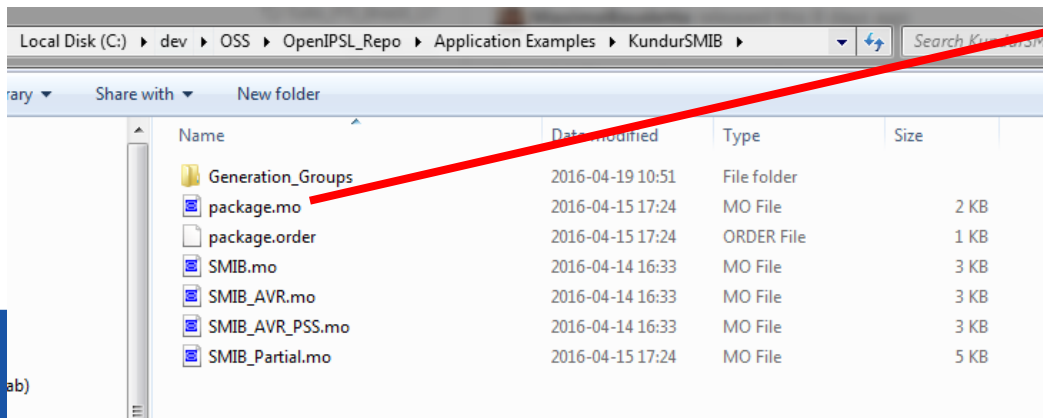
- Unzip the package downloaded at the previous step
- Open OpenModelica Connection Editor (OMEdit)
- Go to **File/Load Library**
- Browse to the location of the unzipped folder
- Choose the **/OpenIPSL** folder
- The icon with the OpenIPSL puppy should appear
- *Alternatively:*
- Drag & drop the **package.mo** file to the **Library Browser** in OMEdit.



Load an Application Example to OMEdit

Once the OpenIPSL is loaded (see previous slide) in OMEdit, you can load an “Application Example”:

- Go to **Open Model/Library File(s)**
- Browse to the location of the unzipped folder
- Go to the **/Application Examples/KundurSMIB** folder, and select **package.mo**
- Alternative:
- Drag & drop the **package.mo** file to the **Library Browser** in OMEdit.





Check that it simulates

Click on “SMIB”, “Simulation Setup” and “Simulate”

OMEdit - OpenModelica Connection Editor

Search Classes

Libraries

- OpenModelica
- ModelicaReference
- ModelicaServices
- Complex
- Modelica
- OpenIPSL
- KundurSMIB
 - SMIB
 - SMIB_AVR
 - SMIB_AVR_PSS
 - Genera...Groups
 - SMIB_Partial

Example 1: Single-machine infinite bus model*
(Constant EIM)

Messages Browser

X: -118.67 Y: 38.52

OMEdit - Simulation Setup - KundurSMIB

Simulation Setup - KundurSMIB.SMIB

General Output Simulation Flags

Simulation Interval

Start Time: 0

Stop Time: 10

Number of Intervals: 100000

Interval: 0.0001

Integration

Method: dassl

Tolerance: 1e-06

DASSL Options

Jacobian: Colored Numerical

Root Finding

Restart After Event

Initial Step Size:

Maximum Step Size:

Maximum Integration Order: 5

Compiler Flags (Optional):

Number of Processors: 4 Use 1 processor if you encounter

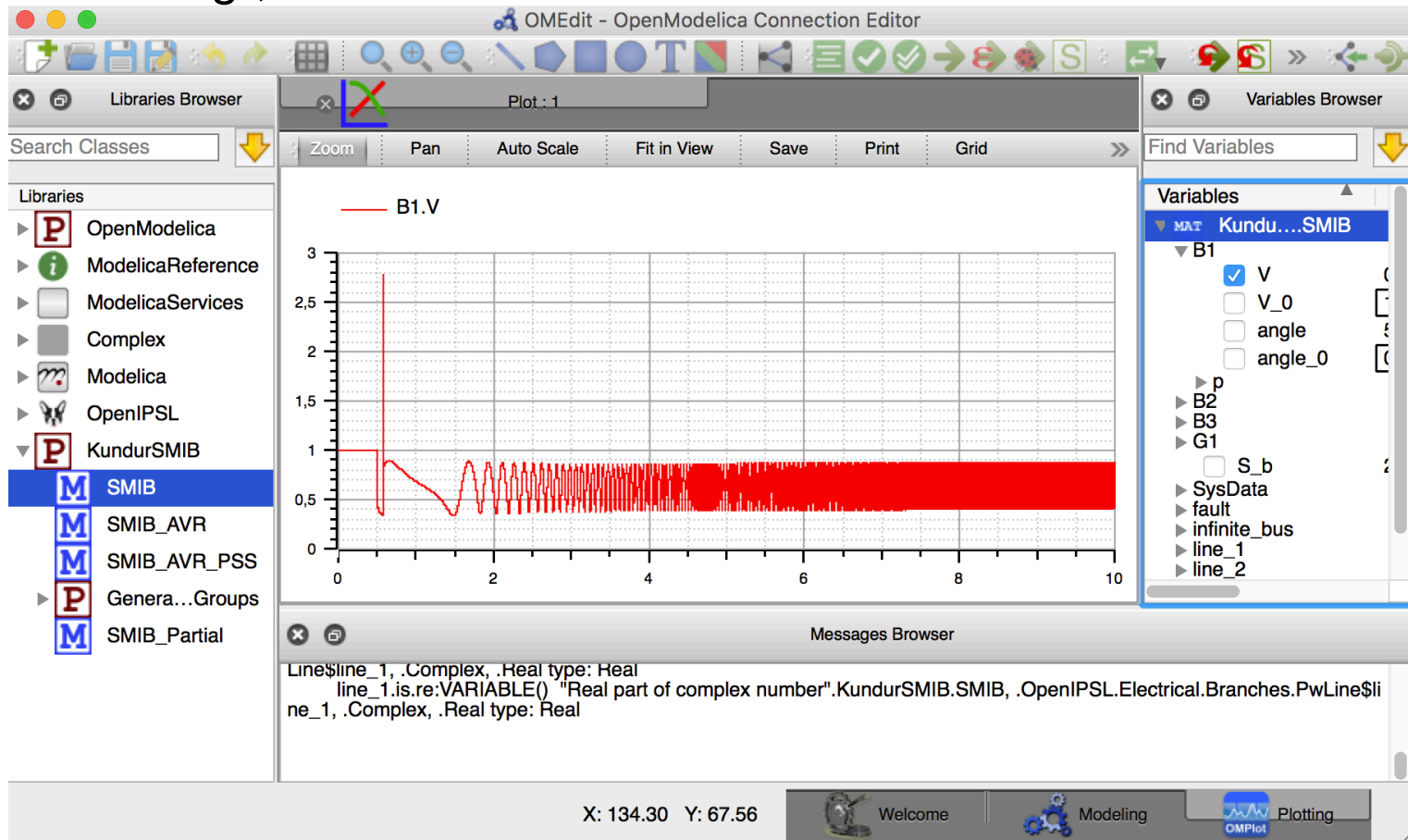
Launch Algorithmic Debugger

on settings inside model

Simulate Cancel

Plot the results

Click on “Plotting”, scroll to “B1” and select “V”



The screenshot shows the OMEdit - OpenModelica Connection Editor interface. The main window displays a plot titled "Plot : 1" showing a red line graph labeled "B1.V". The y-axis ranges from 0 to 3, and the x-axis ranges from 0 to 10. The plot shows a sharp initial spike followed by a damped oscillation that settles around a value of 0.5. The interface includes a Libraries Browser on the left, a Variables Browser on the right, and a Messages Browser at the bottom. The Variables Browser shows the hierarchy: MAT Kundu...SMIB > B1 > V (checked). The Messages Browser shows error messages related to the plot.

Libraries

- OpenModelica
- ModelicaReference
- ModelicaServices
- Complex
- Modelica
- OpenIPSL
- KundurSMIB
 - SMIB
 - SMIB_AVR
 - SMIB_AVR_PSS
 - Genera...Groups
 - SMIB_Partial

Variables

- MAT Kundu...SMIB
 - B1
 - V
 - V_0
 - angle
 - angle_0
 - p
 - B2
 - B3
 - G1
 - S_b
 - SysData
 - fault
 - infinite_bus
 - line_1
 - line_2

Messages Browser

```
Line$line_1, .Complex, .Real type: Real  
line_1.is.re:VARIABLE() "Real part of complex number".KundurSMIB.SMIB, .OpenIPSL.Electrical.Branches.PwLine$li  
ne_1, .Complex, .Real type: Real
```

X: 134.30 Y: 67.56

Welcome Modeling Plotting



Finally, take a look at our repository and documentation!

Repository: <https://github.com/SmarTS-Lab/OpenIPSL>

Go to: <http://openipsl.readthedocs.io/en/latest/index.html>

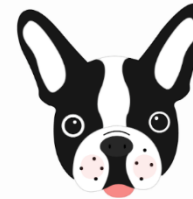
[Docs](#) » OpenIPSL's documentation!

[Edit on GitHub](#)

OpenIPSL's documentation!

Welcome to **OpenIPSL** - The Open-Instance Power System Library.

This documentation is the main source of information for **users** and **developers** working with (or contributing to) the **OpenIPSL** project.



OpenIPSL in short

The OpenIPSL or Open-Instance Power System Library is a [Modelica](#) library, fork of of the [iTesla Power System Library](#) developed and maintained by the [SmarTS Lab](#) research group, collaborators and friends (contributions are welcome!).

The library contains a set of power system component models and test power system networks adopting the "phasor" modeling approach. [Time domain simulations](#) can be carried out using a

You are ready!

See you for the workshop/tutorial/seminar!

