



OpenIPSL

A Modelica Library for Power Systems Simulation

Assoc. Prof. **Luigi Vanfretti**

luigiv@kth.se

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

Preparatory work – aka *home work!*

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)





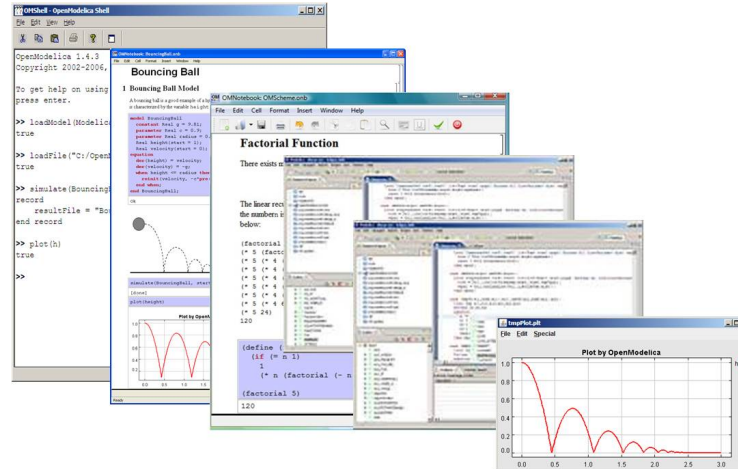
Getting Started with OpenModelica and OpenIPSL

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)

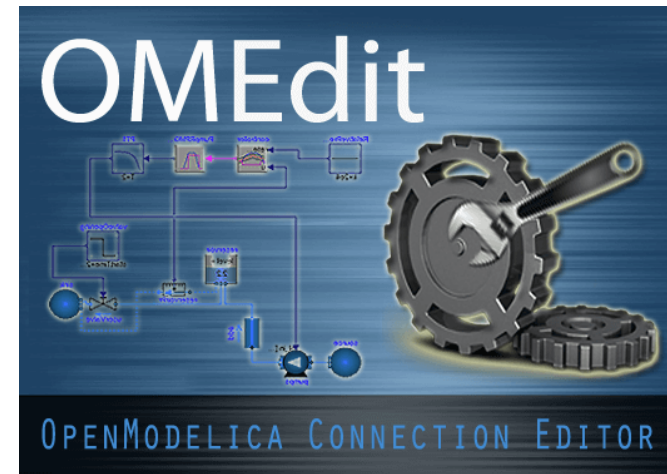
No MAC OSX or Windows OS – No Problem!

- 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



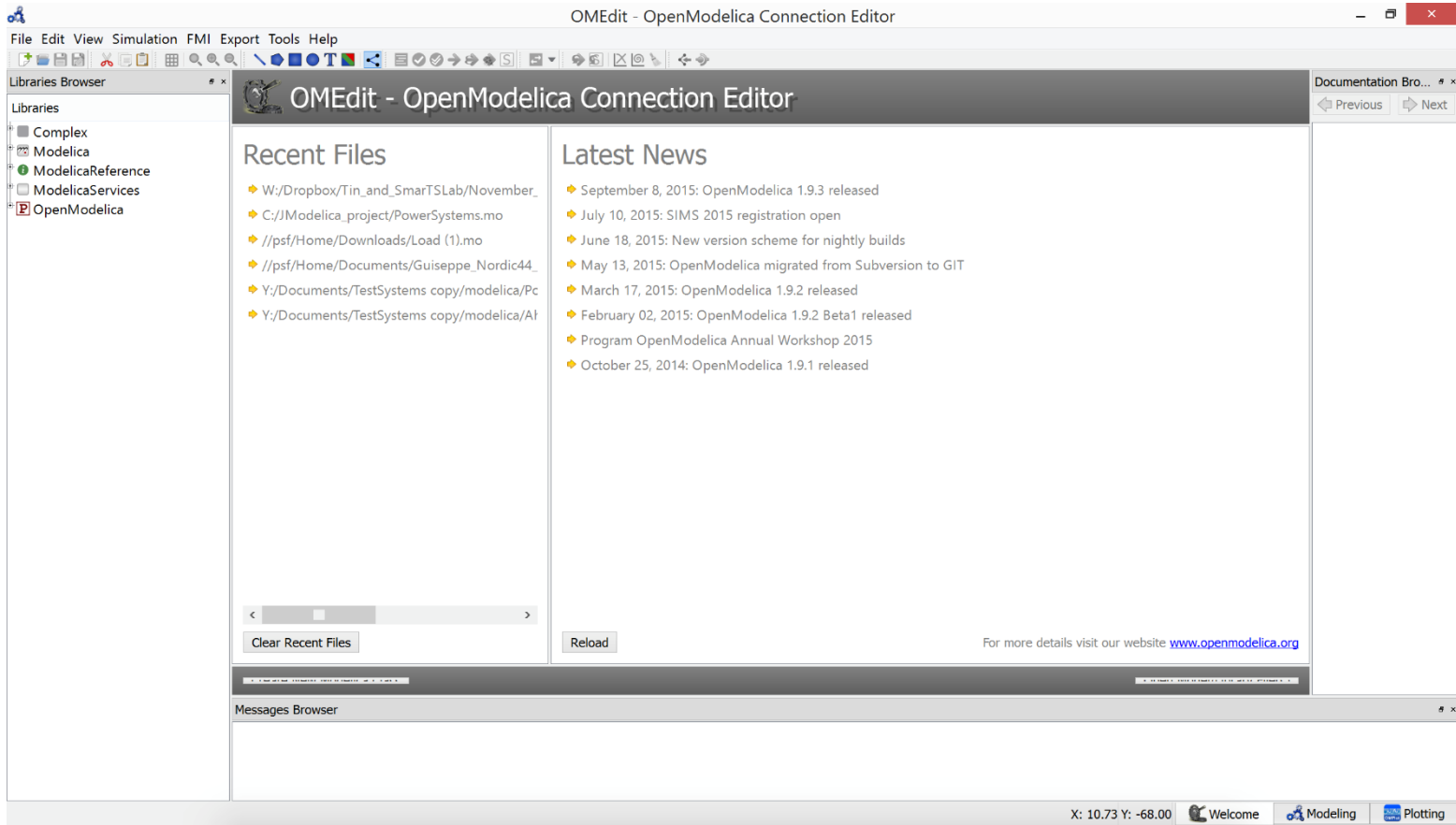
Tasks to check OpenModelica is correctly installed on your computer:

- 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

The screenshot shows a tree view of libraries in OpenModelica. The 'Libraries' folder is expanded to show 'Complex', 'Modelica', 'UsersGuide', 'Blocks', and 'Examples'. Under 'Examples', the 'PID_Controller' block is highlighted in blue. A red arrow points from the text 'Double-click to open' to the 'PID_Controller' entry.

- Libraries
 - Complex
 - Modelica
 - UsersGuide
 - Blocks
 - Examples
 - PID_Controller**
 - Filter
 - FilterWithDifferentiation
 - FilterWithRiseTime
 - InverseModel
 - ShowLogicalSources
 - LogicalNetwork1
 - RealNetwork1
 - IntegerNetwork1
 - BooleanNetwork1
 - Interaction1
 - BusUsage

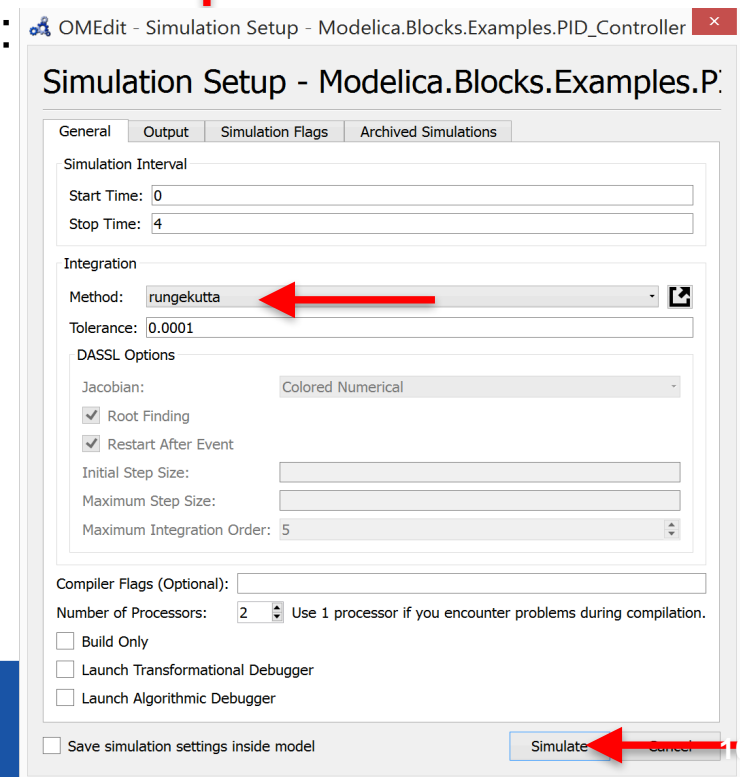
Check of OpenModelica – Step 3

Simulation settings are accessed on the toolbar:



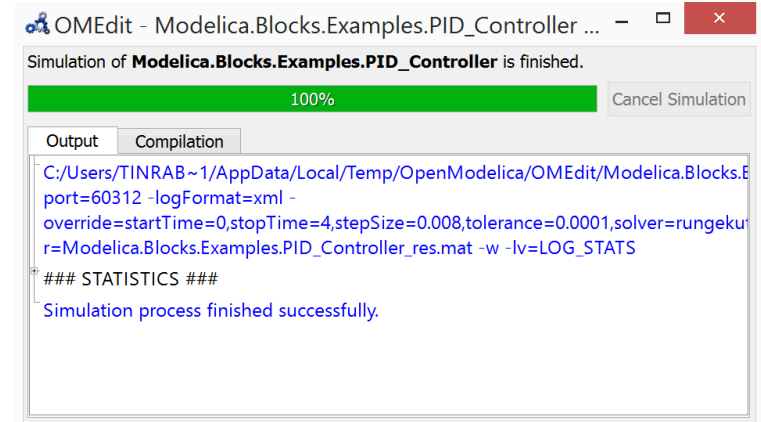
Adjust the settings to match the followings :

Click on **Simulate** to launch the simulation

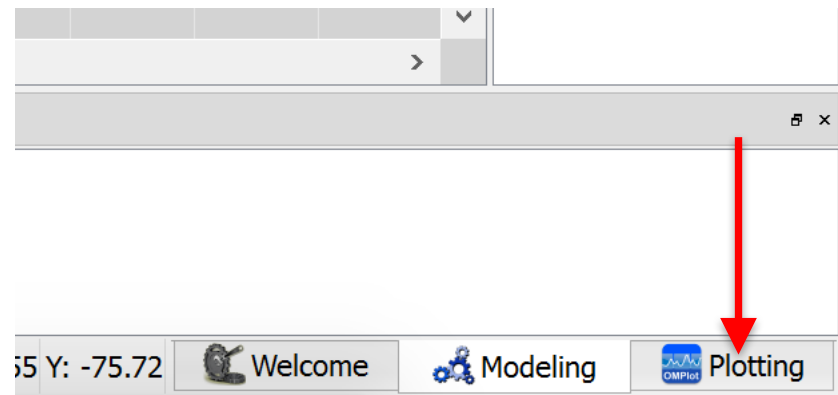


Check of OpenModelica – Step 4a

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

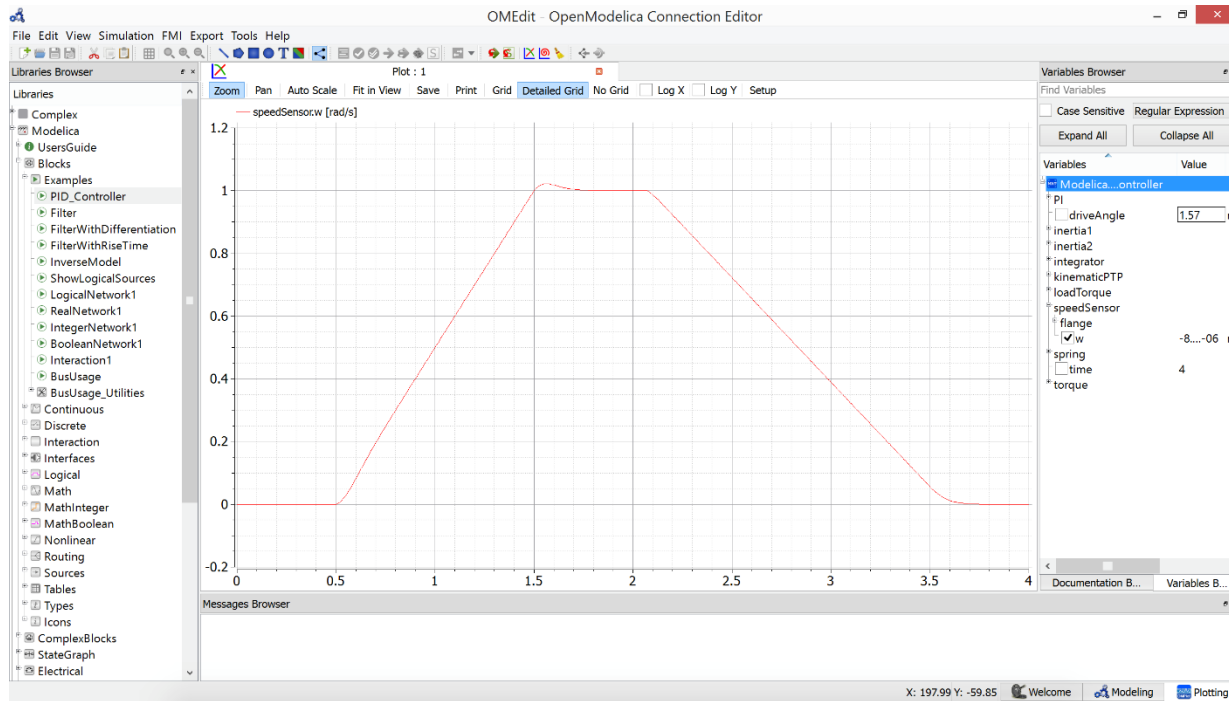
Case Sensitive Regular Expression

Expand All Collapse All

Variables	Value	Unit
Modelica...ontroller		
PI		
driveAngle	1.57	rad
inertia1		
inertia2		
integrator		
kinematicPTP		
loadTorque		
speedSensor		
flange		
<input checked="" type="checkbox"/> w	-8.000000	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 the OpenIPSL!



Go to our Github repo:

https://github.com/SmarTS-Lab/OpenIPSL/releases/tag/Tuto_TAMU_2017

The screenshot shows a web browser window with the URL https://github.com/SmarTS-Lab/OpenIPSL/releases/tag/Tuto_TAMU_2017. The page title is "OpenIPSL Tutorial @TAMU". It indicates a "Pre-release" and shows the release was made by "Ivanfretti" 6 days ago. Below the title is the logo for the "ELECTRICAL & COMPUTER ENGINEERING" department at "TEXAS A&M UNIVERSITY". The text on the page states: "This release of OpenIPSL's 'Tutorial' was prepared for a Tutorial at Texas A&M University organised by Associate Prof. Le Xie on April 20th 2017." It then lists three items: "The preparatory work prior to the tutorial can be found in this .pdf:", "The slides with the hands-on examples, explained step by step, can be found here:", and "The presentation on OpenIPSL given will be made available herein:". Under the heading "Downloads", 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 right side of the download buttons.

Note: A dedicated package will be prepared for the tutorial and uploaded soon.

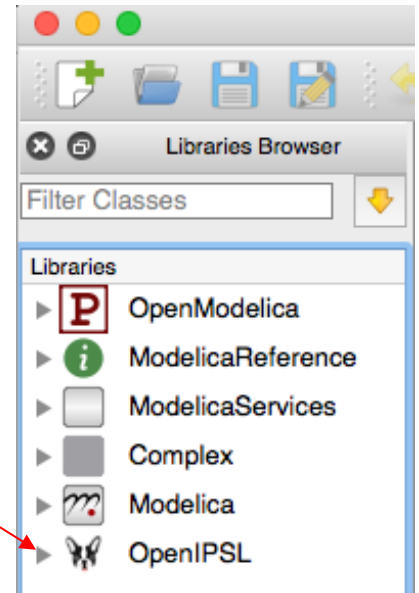
Please download (again!) the package on the day of the tutorial so that you have the most up to date files.

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:

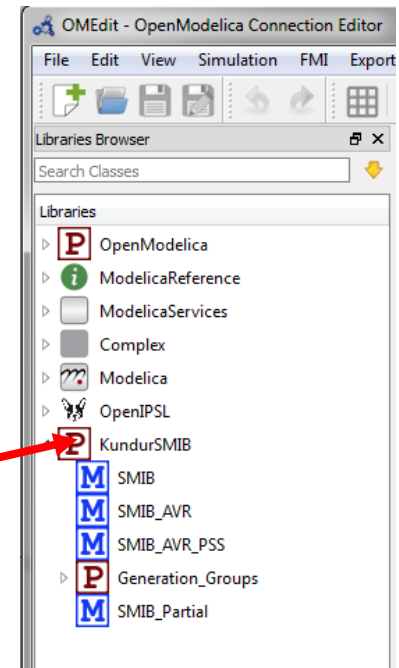
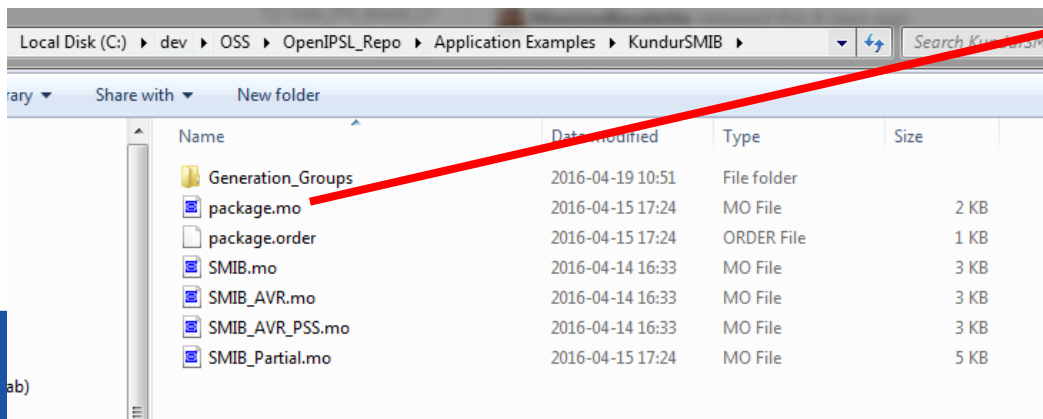
- Unzip the package downloaded at the previous step
- Open OpenModelica Connection Editor (OMEdit)
- Go to **File/Load Library**
- Browse to the location of the unzipped folder
- Choose the **/OpenIPSL** folder
- The icon with the OpenIPSL puppy should appear
- *Alternatively:*
- Drag & drop the **package.mo**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

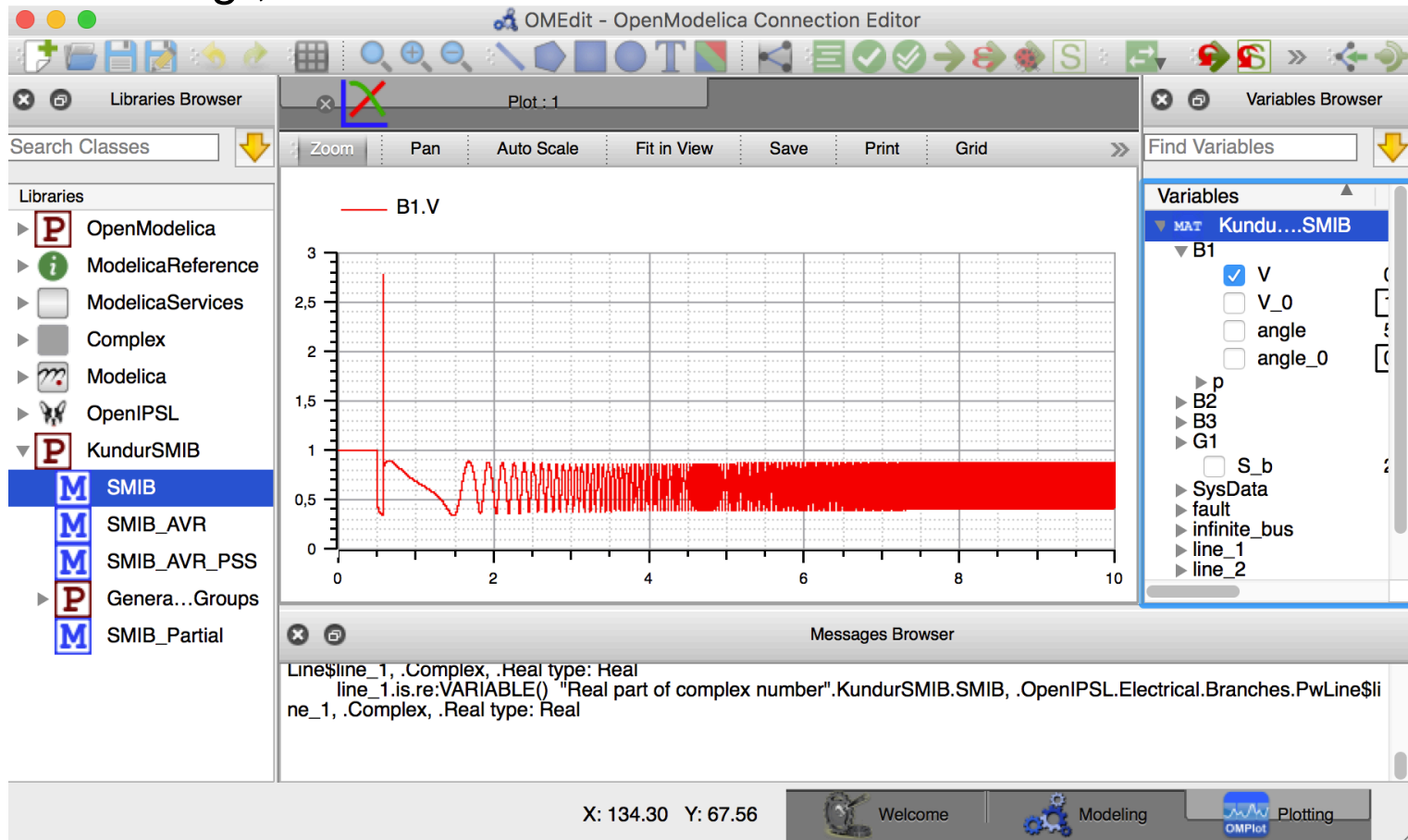
Click on “SMIB”, “Simulation Setup” and “Simulate”

The screenshot shows the OMEdit interface with the 'KundurSMIB.SMIB' model loaded. A red box highlights the 'Simulate' button (represented by a green 'S' icon) in the top toolbar. A red callout box with the text 'Click Here!' is positioned over the 'Simulate' button. The left sidebar shows the 'Libraries' list with 'KundurSMIB' expanded to show 'SMIB' and other sub-models. The main workspace displays a circuit diagram of a single-machine infinite bus model. The bottom status bar shows coordinates 'X: -118.67 Y: 38.52' and a 'Welcome Modeling' message.

The screenshot shows the 'Simulation Setup - KundurSMIB.SMIB' dialog box. The 'General' tab is active, showing simulation interval settings: Start Time: 0, Stop Time: 10, Number of Intervals: 100000, and Interval: 0.0001. The 'Integration' section shows the Method set to 'dassl', Tolerance: 1e-06, and DASSL Options including 'Colored Numerical', 'Root Finding', and 'Restart After Event'. The 'Compiler Flags (Optional):' field is empty, and 'Number of Processors' is set to 4. A red callout box with the text 'Click Here!' is positioned over the 'Simulate' button at the bottom right of the dialog. The 'Launch Algorithmic Debugger' checkbox is unchecked.

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" with a red line representing the variable "B1.V". The plot shows a transient response starting at t=0, with a peak voltage of approximately 2.8, followed by a damped oscillation that settles around 0.8. The x-axis represents time from 0 to 10, and the y-axis represents voltage from 0 to 3.

The Variables Browser on the right shows the hierarchy: MAT Kundu...SMIB > B1 > V. The variable "V" is selected with a checkmark. Other variables listed include V_0, angle, angle_0, p, B2, B3, G1, S_b, SysData, fault, infinite_bus, line_1, and line_2.

The Messages Browser at the bottom shows the following error message:

```

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
  
```

The status bar at the bottom indicates coordinates X: 134.30 Y: 67.56 and shows navigation buttons for Welcome, Modeling, and Plotting (OMPlot).



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!

