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

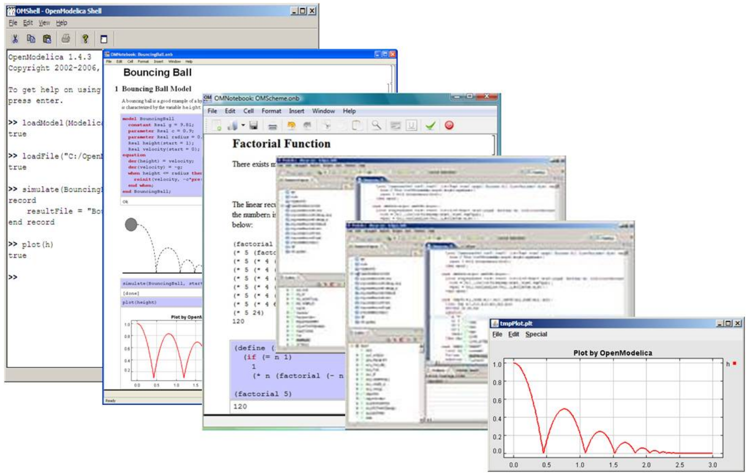
Mac:

- 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



```
OpenModelica 1.4.3
Copyright 2002-2006,

To get help on using
press enter.

>> loadModel(Modelic
true

>> loadFile("C:/OpenM
true

>> simulate(Bouncing
record
    resultFile = "Bo
end record

>> plot(h)
true

>>
```

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!

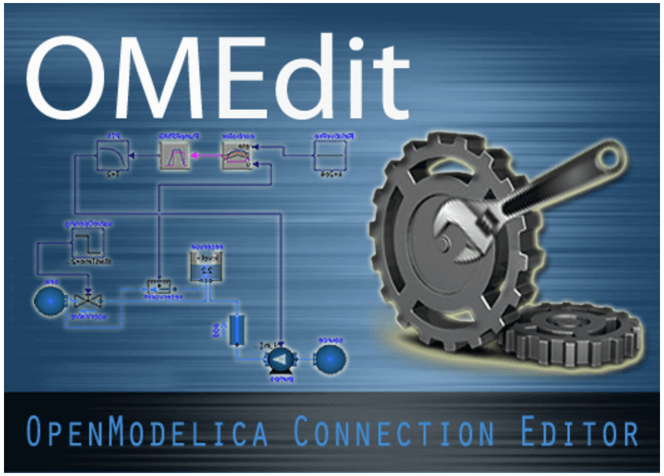
Compatibilty with OpenIPSL is checked for OpenModelica versions 1.9.6 (Mac and Windows) and 1.9.11 (on Windows)

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

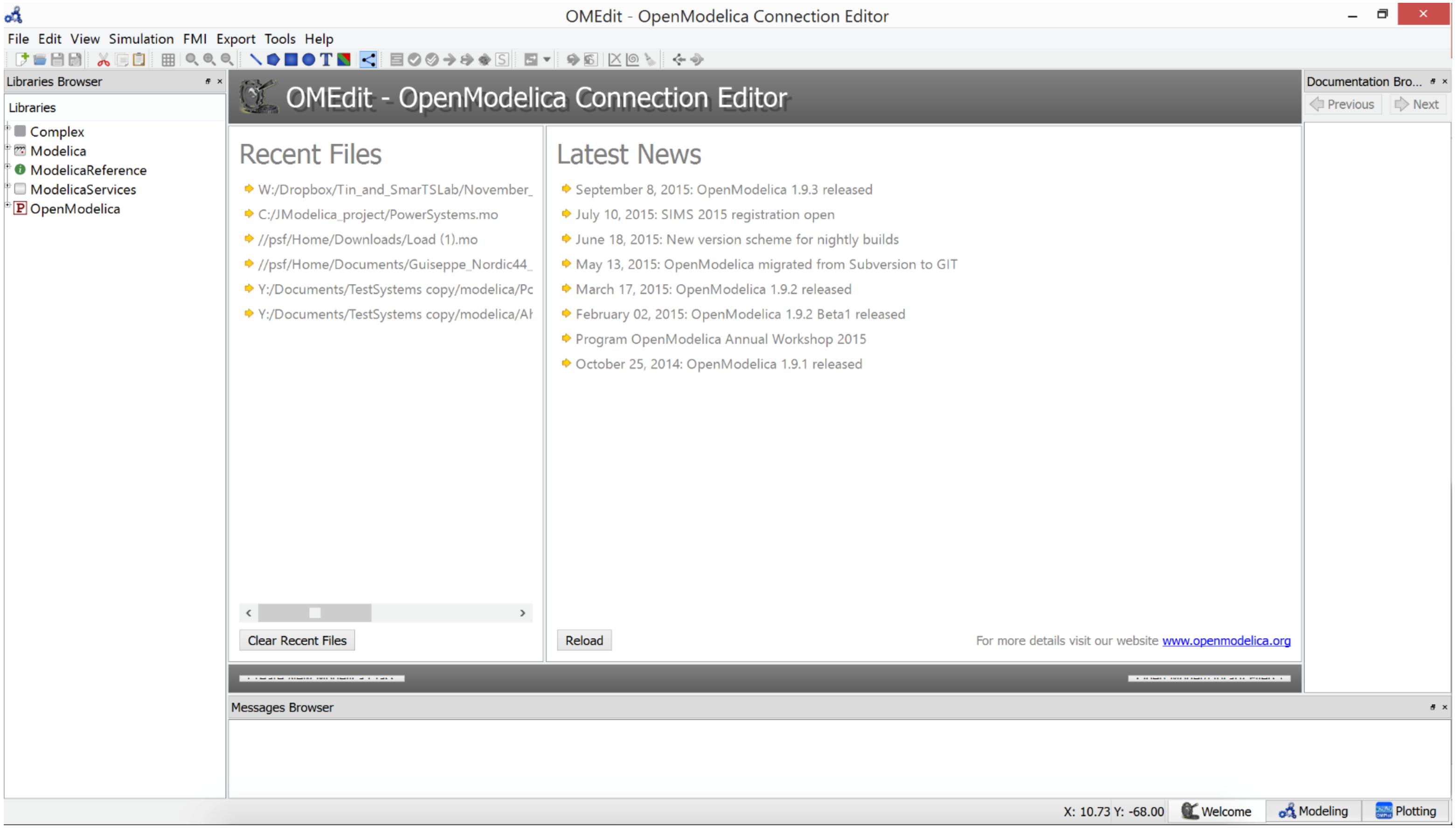


Tasks to check OpenModelica is correctly installed on your computer:

- 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



OMEdit - OpenModelica Connection Editor

Recent Files

- W:/Dropbox/Tin_and_SmarTSLab/November_
- C:/JModelica_project/PowerSystems.mo
- //psf/Home/Downloads/Load (1).mo
- //psf/Home/Documents/Guiseppe_Nordic44_
- Y:/Documents/TestSystems copy/modelica/Pc
- Y:/Documents/TestSystems copy/modelica/Al

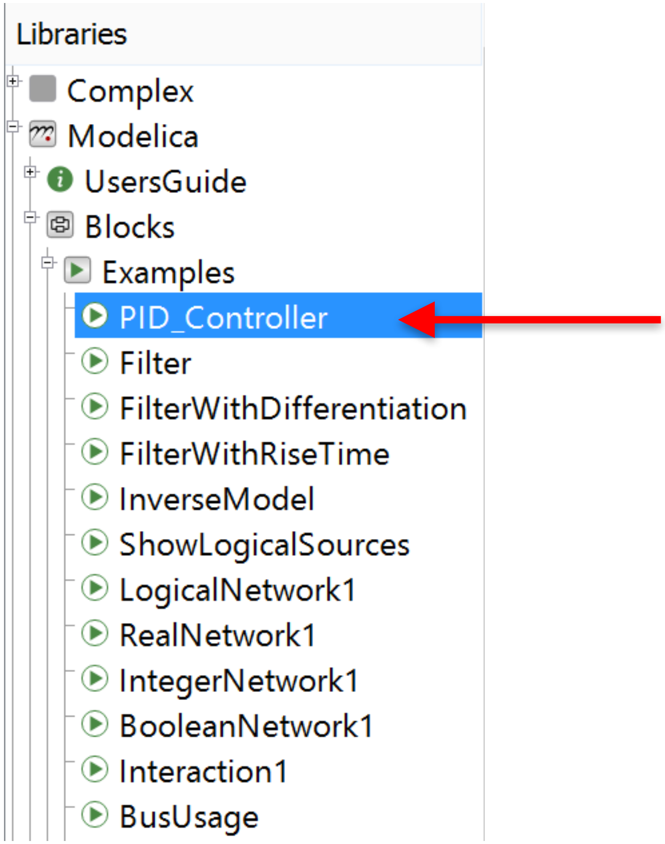
Latest News

- September 8, 2015: OpenModelica 1.9.3 released
- July 10, 2015: SIMS 2015 registration open
- June 18, 2015: New version scheme for nightly builds
- May 13, 2015: OpenModelica migrated from Subversion to GIT
- March 17, 2015: OpenModelica 1.9.2 released
- February 02, 2015: OpenModelica 1.9.2 Beta1 released
- Program OpenModelica Annual Workshop 2015
- October 25, 2014: OpenModelica 1.9.1 released

For more details visit our website www.openmodelica.org

Check of OpenModelica – Step 2

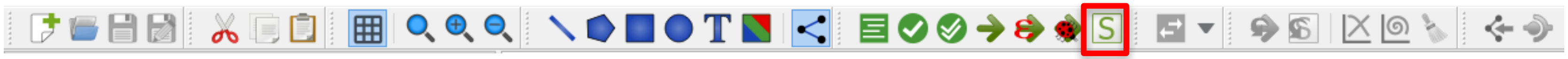
Browse the Modelica library to find the PID_Controller and open it



Double-click to open

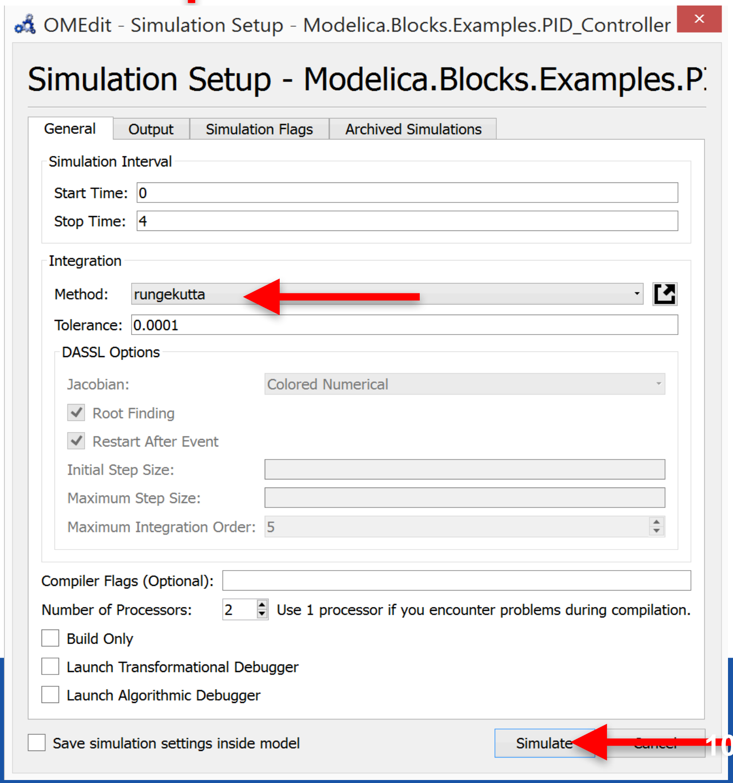
Check of OpenModelica – Step 3

Simulation settings are accessed on the toolbar:



Adjust the settings to match the followings :

Click on **Simulate** to launch the simulation



OMEdit - Simulation Setup - Modelica.Blocks.Examples.PID_Controller

Simulation Setup - Modelica.Blocks.Examples.P

General | Output | Simulation Flags | Archived Simulations

Simulation Interval

Start Time: 0

Stop Time: 4

Integration

Method: rungekutta

Tolerance: 0.0001

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: 2 Use 1 processor if you encounter problems during compilation.

□ Build Only

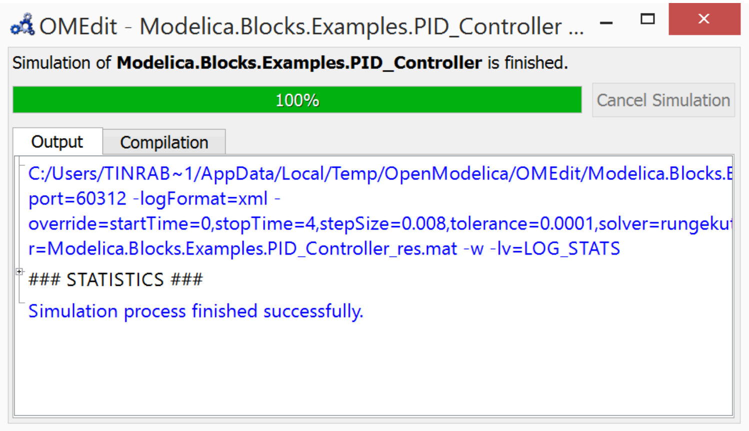
□ Launch Transformational Debugger

□ Launch Algorithmic Debugger

□ Save simulation settings inside model

Check of OpenModelica – Step 4a

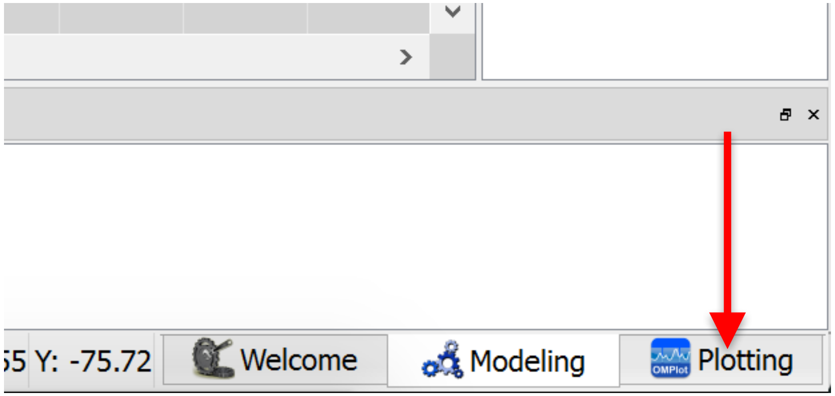
Once the simulation is completed (100 %):



Simulation of **Modelica.Blocks.Examples.PID_Controller** is finished.

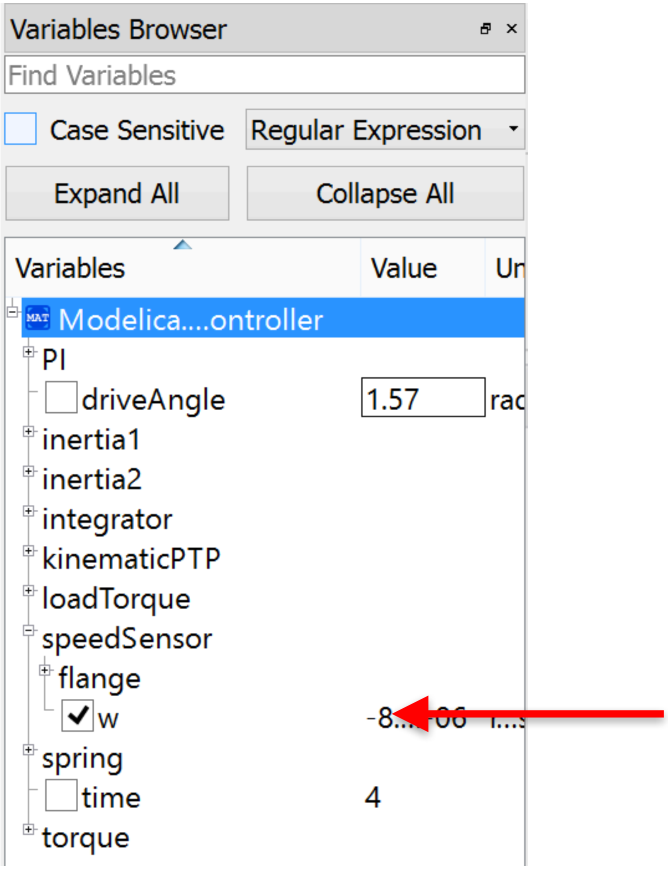
```
C:/Users/TINRAB~1/AppData/Local/Temp/OpenModelica/OMEdit/Modelica.Blocks.E
port=60312 -logFormat=xml -
override=startTime=0,stopTime=4,stepSize=0.008,tolerance=0.0001,solver=rungeku
r=Modelica.Blocks.Examples.PID_Controller_res.mat -w -lv=LOG_STATS
### STATISTICS ###
Simulation process finished successfully.
```

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

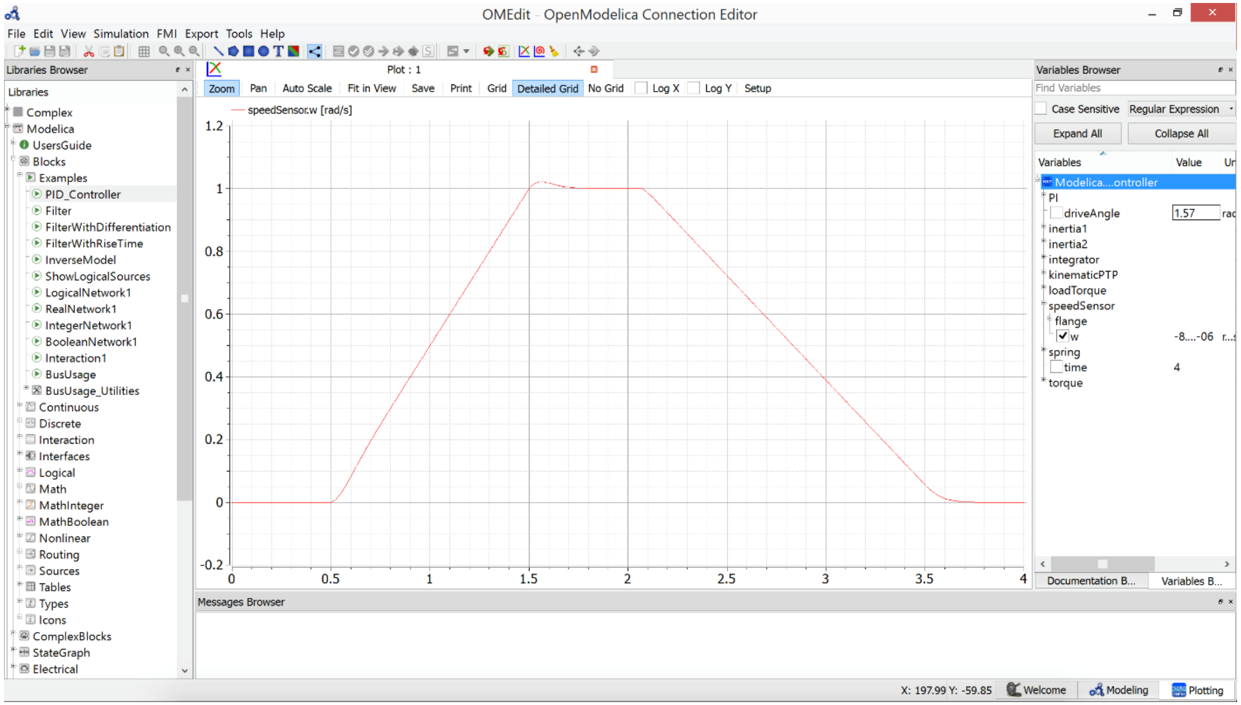


Check of OpenModelica – Step 4b

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



Check of OpenModelica – Final Result



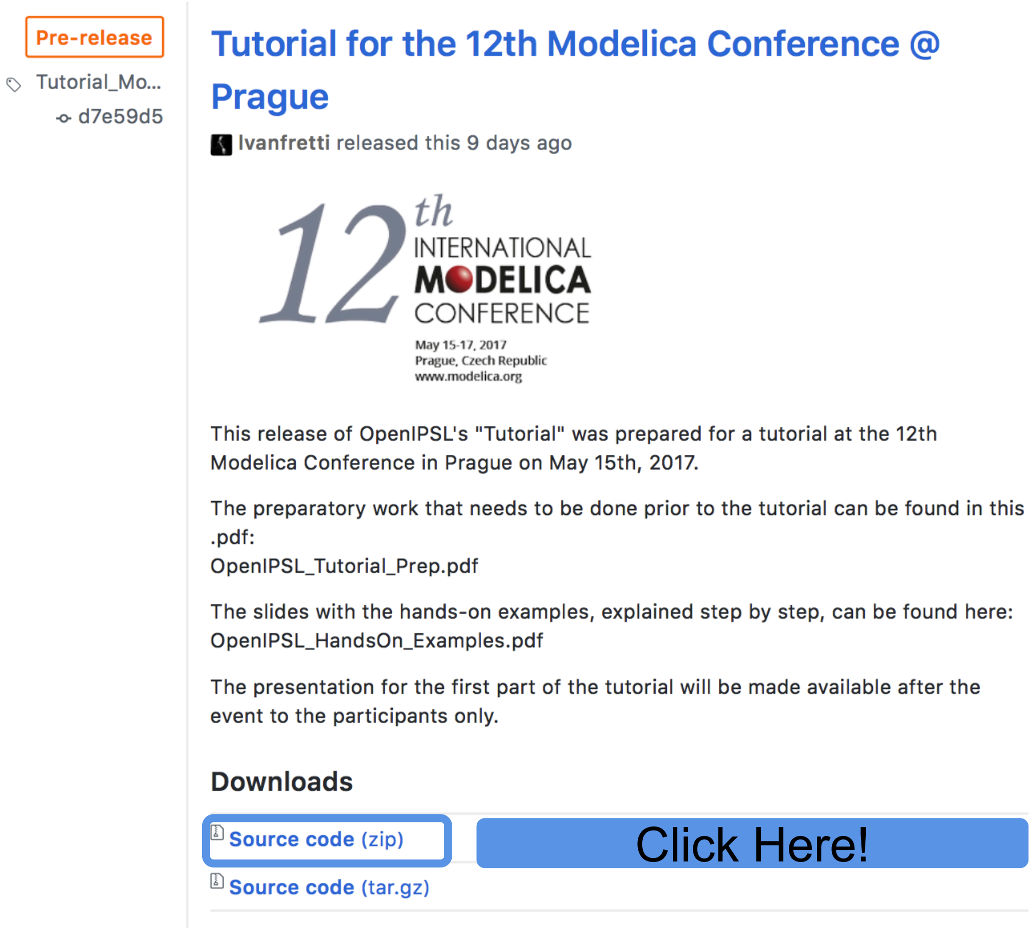
If your screen looks like this, you're ready to go!

Part 2

Setting up OpenIPSL

Download the OpenIPSL Source!

Go to our Github repo: https://github.com/SmarTS-Lab/OpenIPSL/releases/tag/Tutorial_ModelicaConf2017



Pre-release

Tutorial_Mo...

d7e59d5

Tutorial for the 12th Modelica Conference @ Prague

lvanfretti released this 9 days ago

12th INTERNATIONAL MODELICA CONFERENCE

May 15-17, 2017
Prague, Czech Republic
www.modelica.org

This release of OpenIPSL's "Tutorial" was prepared for a tutorial at the 12th Modelica Conference in Prague on May 15th, 2017.

The preparatory work that needs to be done prior to the tutorial can be found in this .pdf:
OpenIPSL_Tutorial_Prep.pdf

The slides with the hands-on examples, explained step by step, can be found here:
OpenIPSL_HandsOn_Examples.pdf

The presentation for the first part of the tutorial will be made available after the event to the participants only.

Downloads

Source code (zip)

Source code (tar.gz)

Click Here!

Note: The files will also be available on a USB stick(s) that we can circulate on the day of the tutorial.

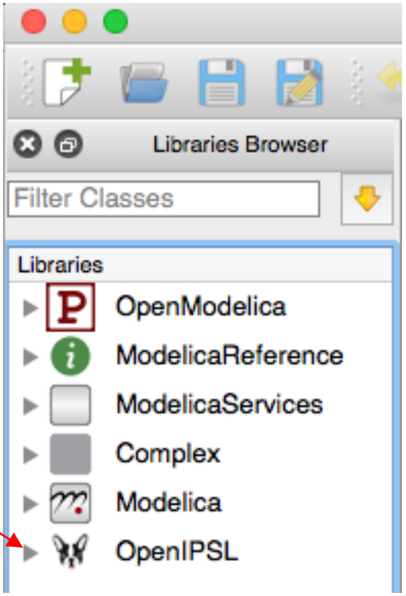
Please ask me for it if you need it.

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



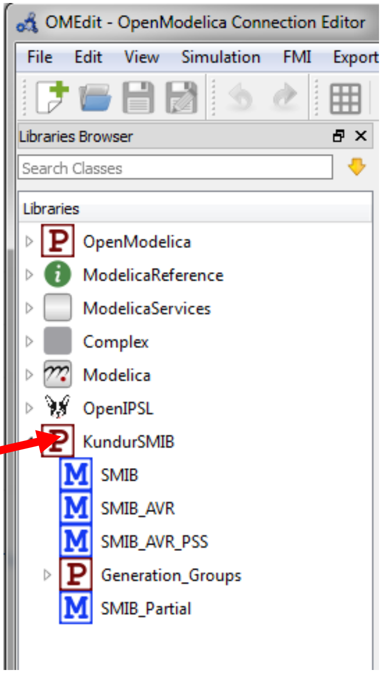
- *Alternatively:*
- Drag & drop the **package.mo** file to the **Library Browser** in OMEdit.

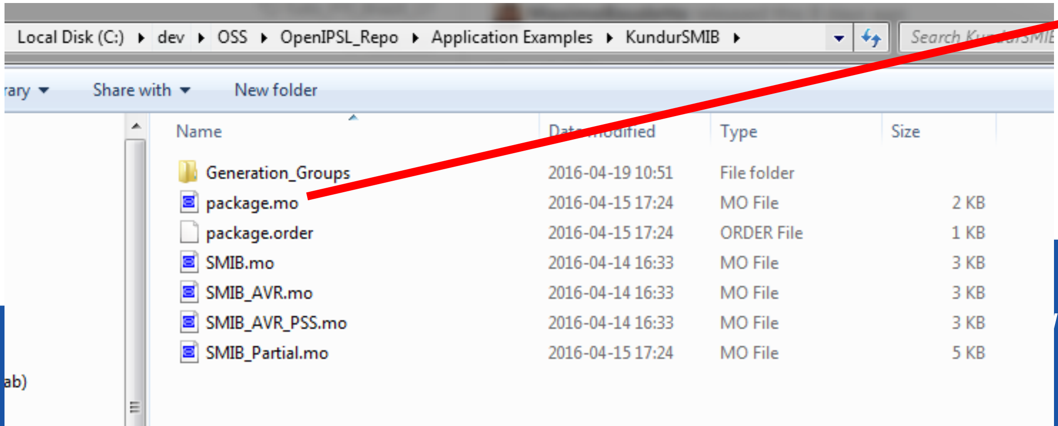
Load an Application Example to OMEdit

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**

- *Alternatively:*
- Drag & drop the **package.mo** file to the **Library Browser** in OMEdit.

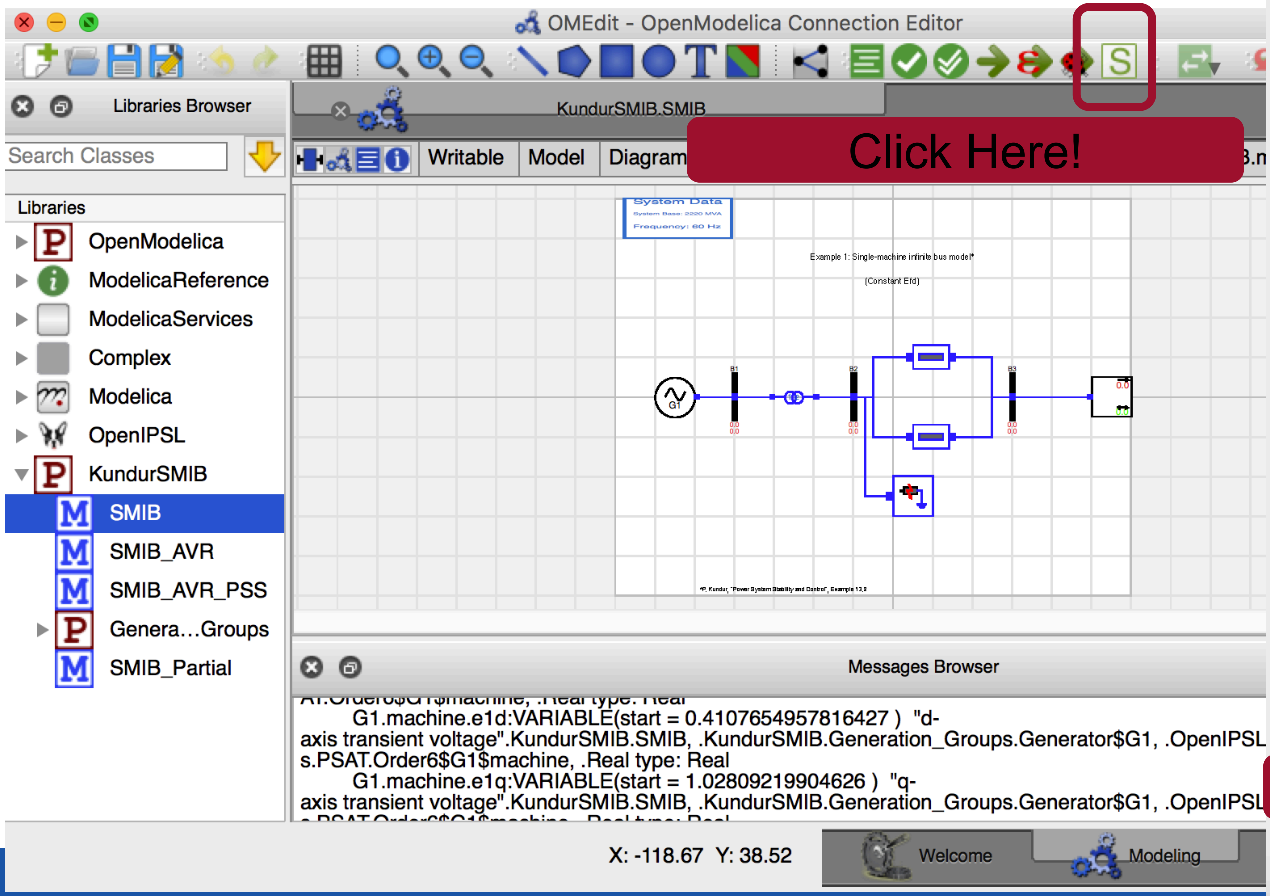



Local Disk (C:) ▸ dev ▸ OSS ▸ OpenIPSL_Repo ▸ Application Examples ▸ KundurSMIB ▸
Search KundurSMIB
rary ▾
Share with ▾
New folder
ab)

Name	Date modified	Type	Size
Generation_Groups	2016-04-19 10:51	File folder	
package.mo	2016-04-15 17:24	MO File	2 KB
package.order	2016-04-15 17:24	ORDER File	1 KB
SMIB.mo	2016-04-14 16:33	MO File	3 KB
SMIB_AVR.mo	2016-04-14 16:33	MO File	3 KB
SMIB_AVR_PSS.mo	2016-04-14 16:33	MO File	3 KB
SMIB_Partial.mo	2016-04-15 17:24	MO File	5 KB

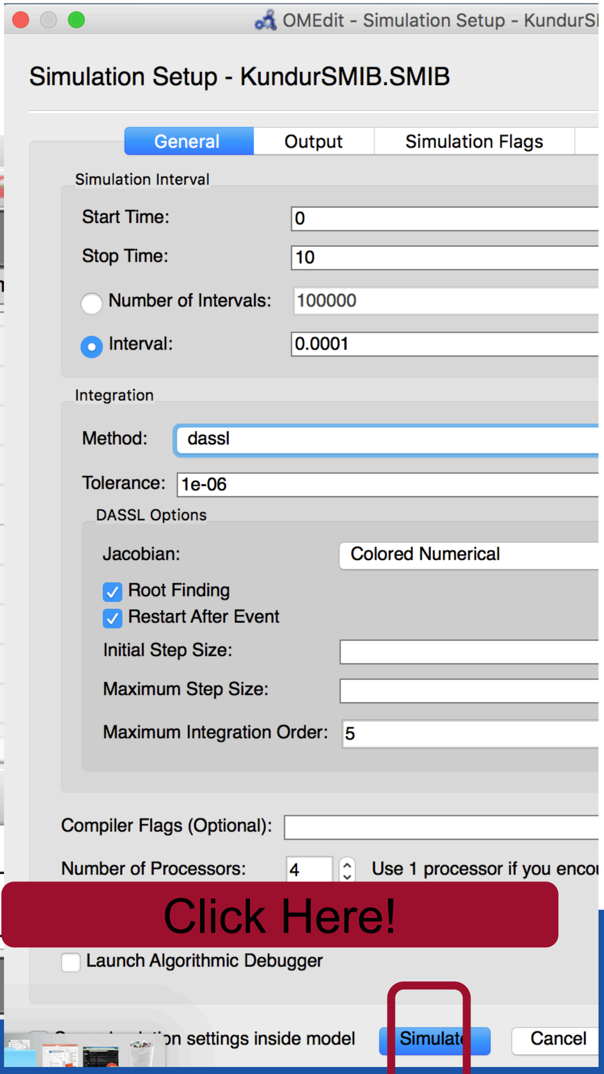
Check that it simulates

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



```
A1.Order6$G1$machine, .Real type: Real
        G1.machine.e1d:VARIABLE(start = 0.4107654957816427 )  "d-
axis transient voltage".KundurSMIB.SMIB, .KundurSMIB.Generation_Groups.Generator$G1, .OpenIPSL
s.PSAT.Order6$G1$machine, .Real type: Real
        G1.machine.e1q:VARIABLE(start = 1.02809219904626 )  "q-
axis transient voltage".KundurSMIB.SMIB, .KundurSMIB.Generation_Groups.Generator$G1, .OpenIPSL
s.PSAT.Order6$G1$machine, .Real type: Real
```

Click Here!



Simulation Setup - KundurSMIB.SMIB

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 enco

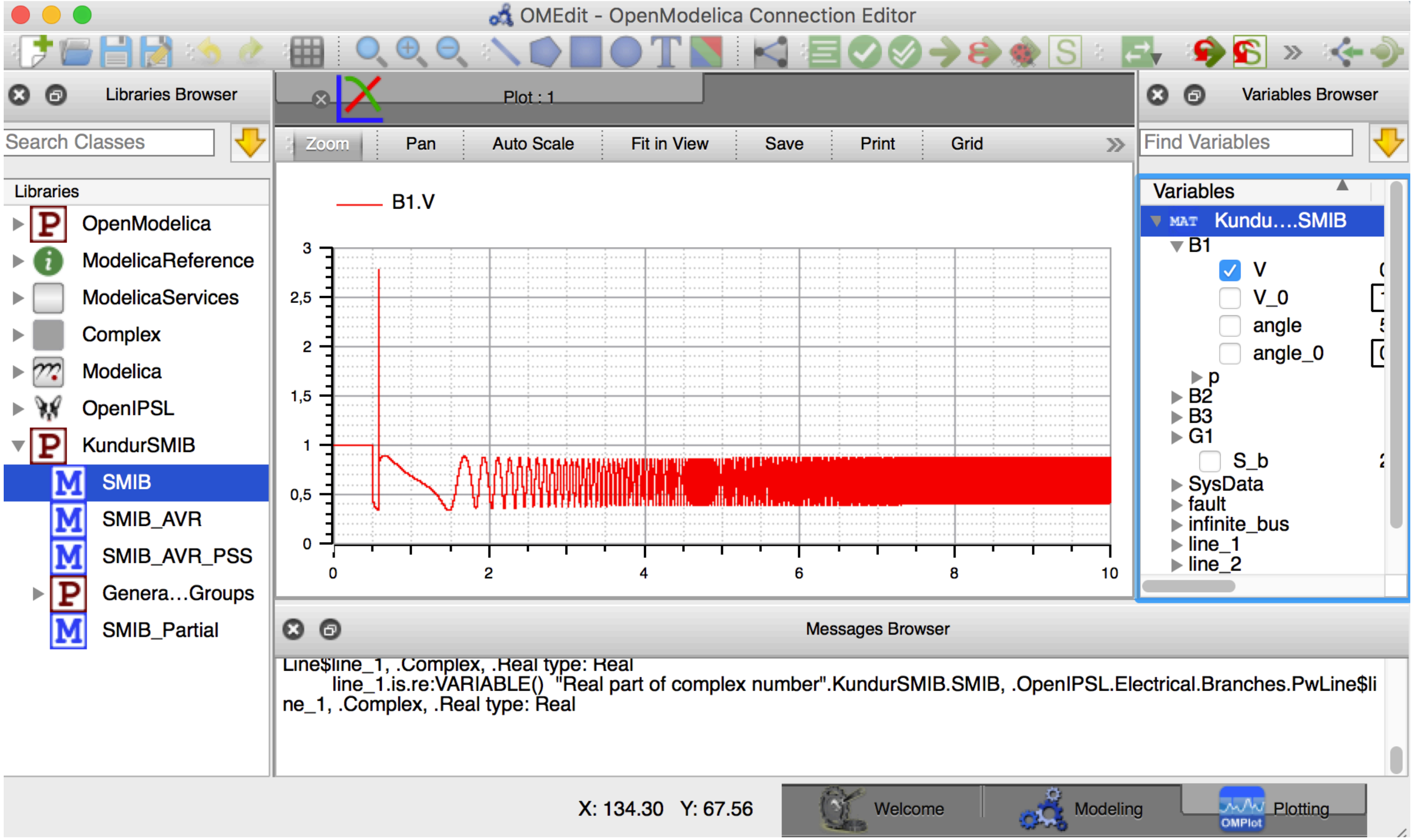
□ Launch Algorithmic Debugger

on settings inside model

Click Here!

Plot the results

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

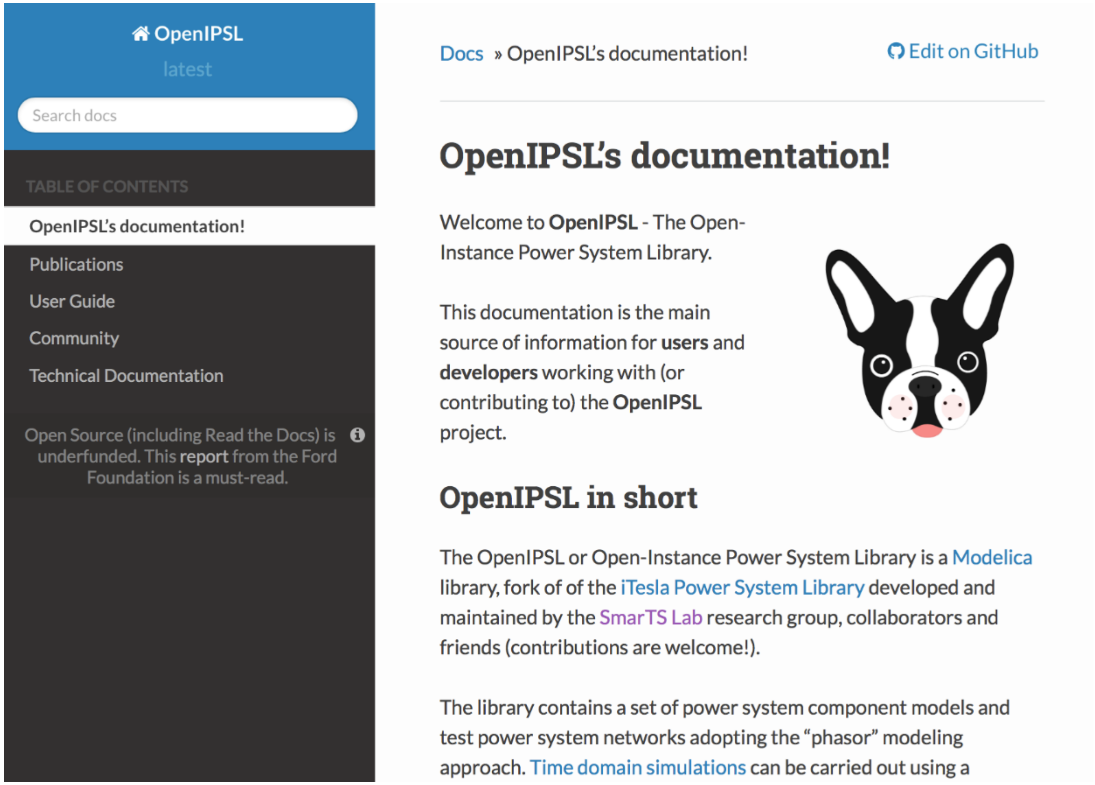


```
Line$line_1, .Complex, .Real type: Real
          line_1.is.re:VARIABLE()  "Real part of complex number".KundurSMIB.SMIB, .OpenIPSL.Electrical.Branches.PwLine$li
ne_1, .Complex, .Real type: Real
```

Finally, take a look a our repository and documentation!

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

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



OpenIPSL

latest

Search docs

TABLE OF CONTENTS

- OpenIPSL's documentation!
- Publications
- User Guide
- Community
- Technical Documentation

Open Source (including Read the Docs) is underfunded. This **report** from the Ford Foundation is a must-read.

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!