MODPROD 2016

iPSL Tutorial

Tin Rabuzin and Luigi Vanfretti

E-mail: luigiv@kth.se
Web: http://www.vanfretti.com

luigiv@kth.se
Associate Professor, Docent
Electric Power Systems Dept.
KTH
Stockholm, Sweden

Luigi.Vanfretti@statnett.no
Special Advisor in Strategy and Public Affairs
Research and Development Division
Statnett SF
Oslo, Norway

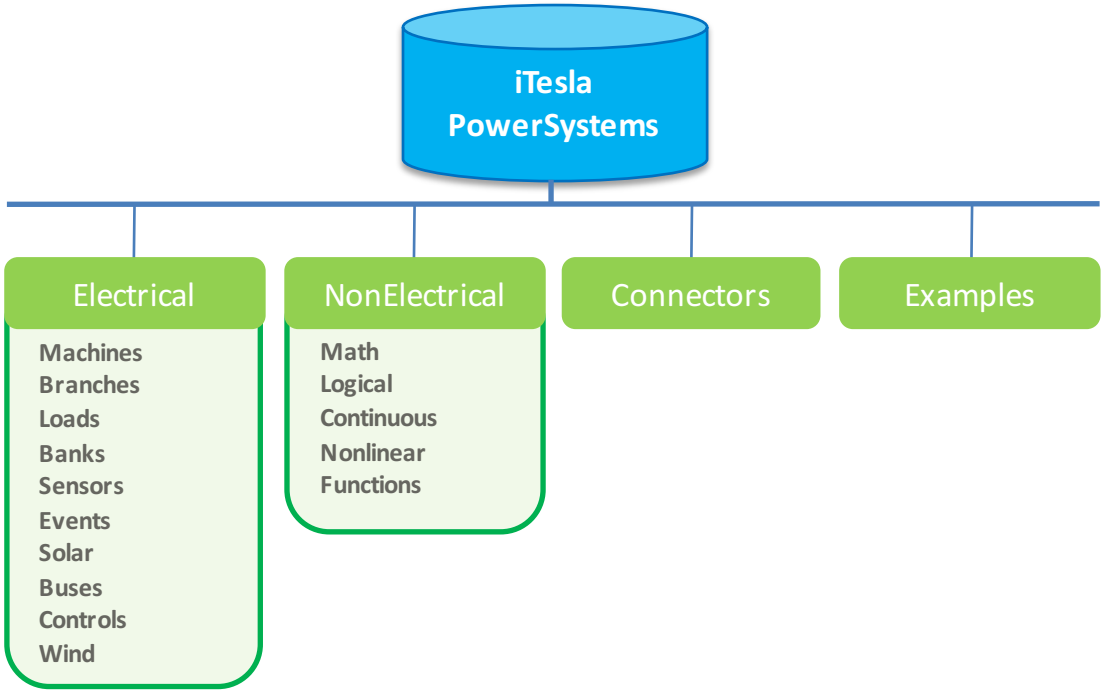
Outline

1. Aim of the workshop
2. Library introduction
3. Example 1
4. Example 2
5. Example 3

Aim of the workshop

- Introduction to the iTesla PowerSystems Library
- Modelling and simulation possibilities by using iPSL and Modelica
- Comparison of the performance with a reference simulation software
- 3 use cases with a dynamic simulation and linearization

Library introduction



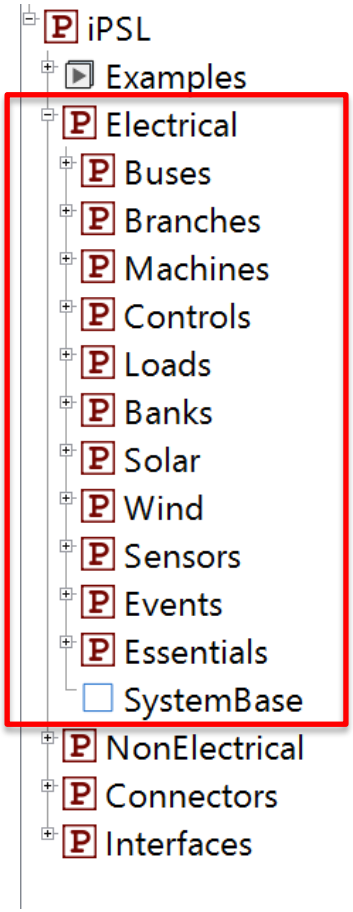
iTesla PowerSystems library is divided in four main categories:

- Examples
- Electrical
- NonElectrical
- Connectors

Library introduction

Electrical

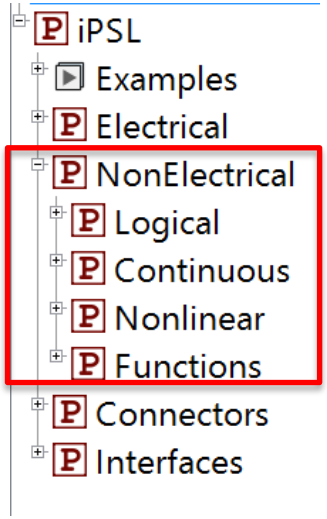
- The *Electrical* package contains most of the components that comprise an actual power network
- E.g., electrical machines, transmission lines, loads, excitation systems, turbine governors, etc.
- These are used to build the power system network models



Library introduction

NonElectrical

- The *NonElectrical* package is comprised by functions, blocks or models, which are used to build the aforementioned power system component models
- Transfer functions, logical operators, etc.
- They perform specific operations which were not available in the Modelica Standard Library (MSL)



Library introduction

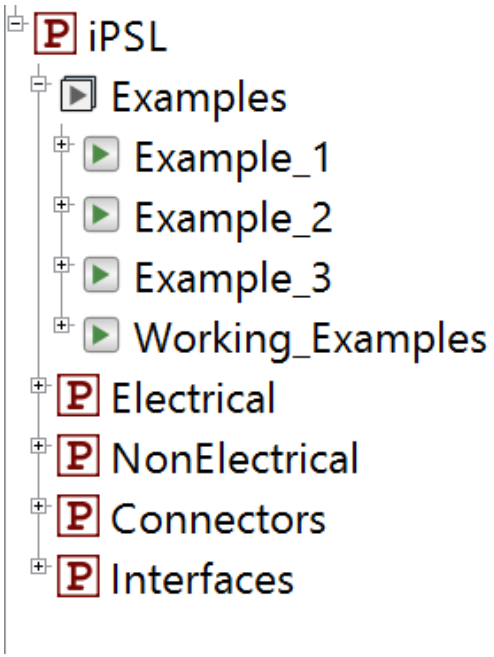
Connectors

- The *Connectors* package contains a set of specifically developed Modelica connectors to harmonize the models in this library
- As an example, *PwPin* is the connector, which contains voltage and current quantities in phasor representation

Library introduction

Examples

- In this workshop, *Examples* package will be used to showcase the possibilities of the library
- In the packages *Example_1, Example_2* and *Example_3* prepared use cases can be found where steps to build the models are described
- Package *Working_Examples* and corresponding subpackages will be used by attendees of the workshop to create use cases on their own
- Note! The library provided during the workshop is a stripped version with a reduced number of models.



OMEdit and iPSL

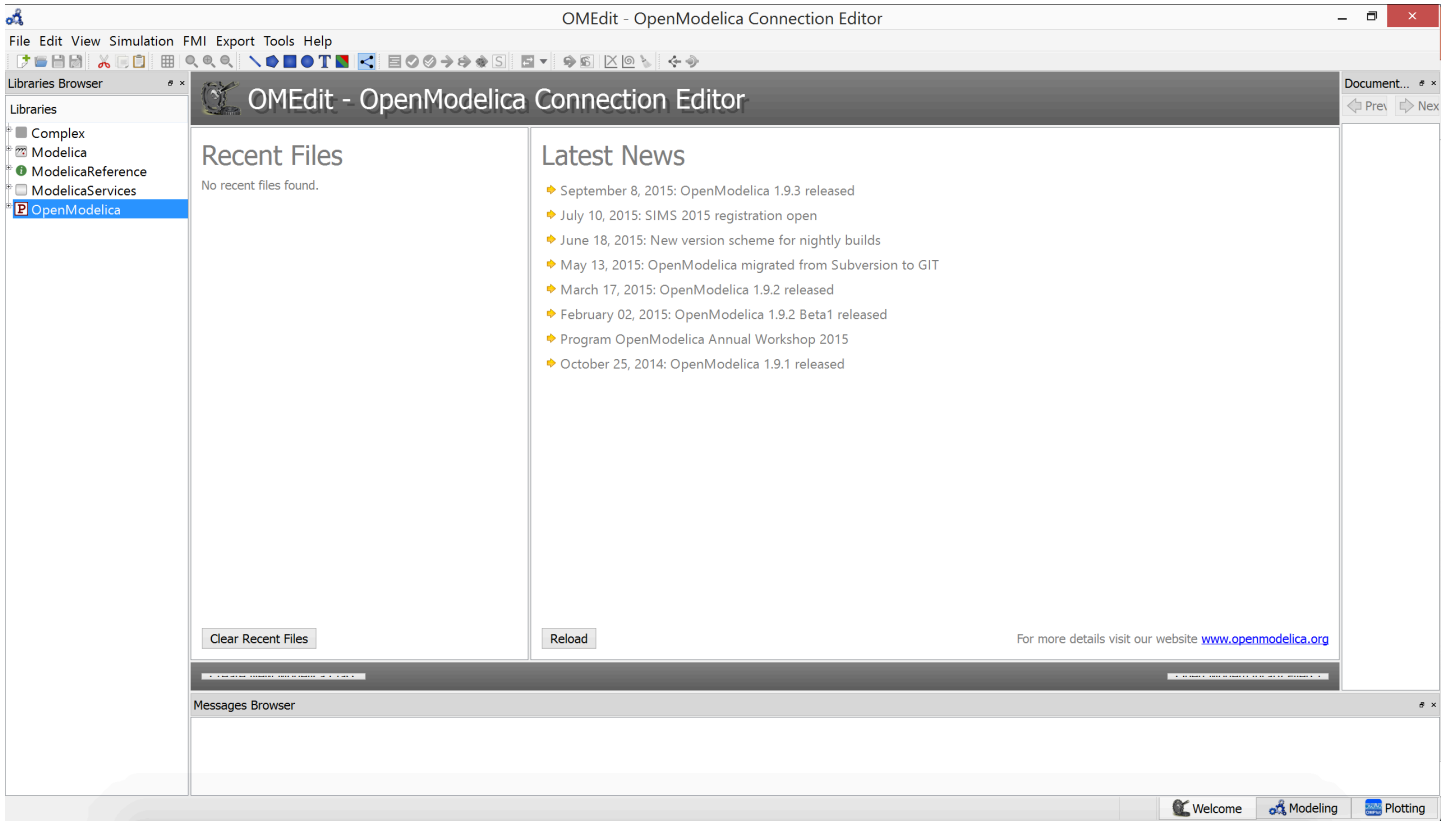
iPSL along with the additional files should be downloaded from:

https://www.dropbox.com/sh/m0tukq5089kjnmz/AAAEyIs7oGPqjDSZgZHfAjcNa?dl=0

Downloaded content should be placed on C:\iPSL_Workshop

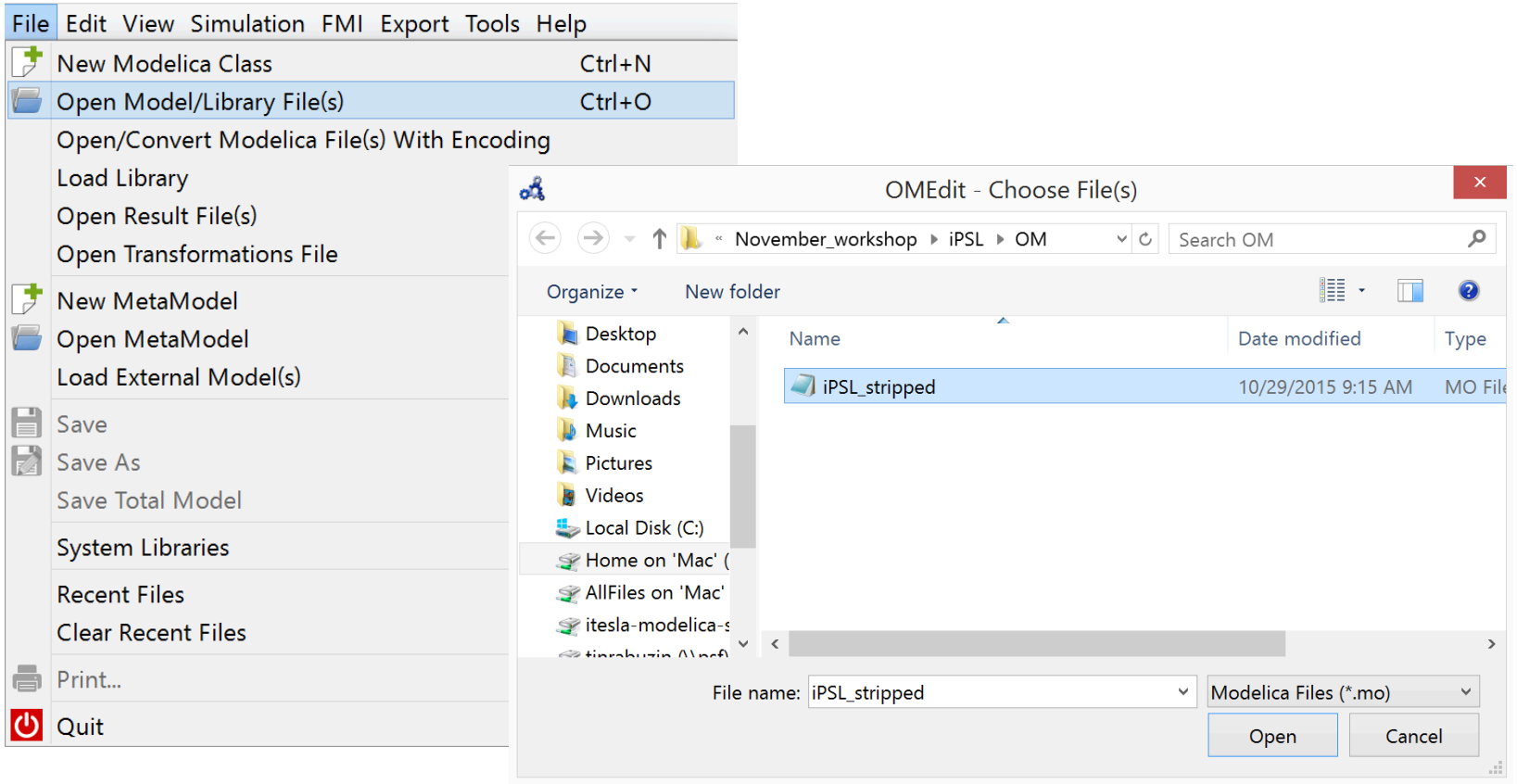
OMEdit and iPSL

- The workshop will be carried out in OpenModelica Connection Editor (OMEdit)
- The start screen of OMEdit should look like the one shown on the figure



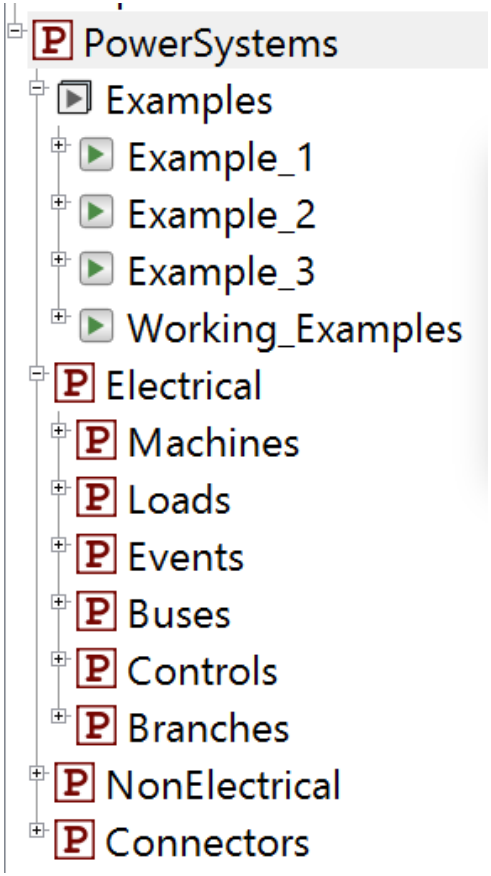
OMEdit and iPSL

- To open the iPSL one should go to File->Open Model/Library File(s) and navigate to the iPSL file that should be located in `C:/iPSL_Workshop/iPSL_stripped.mo`

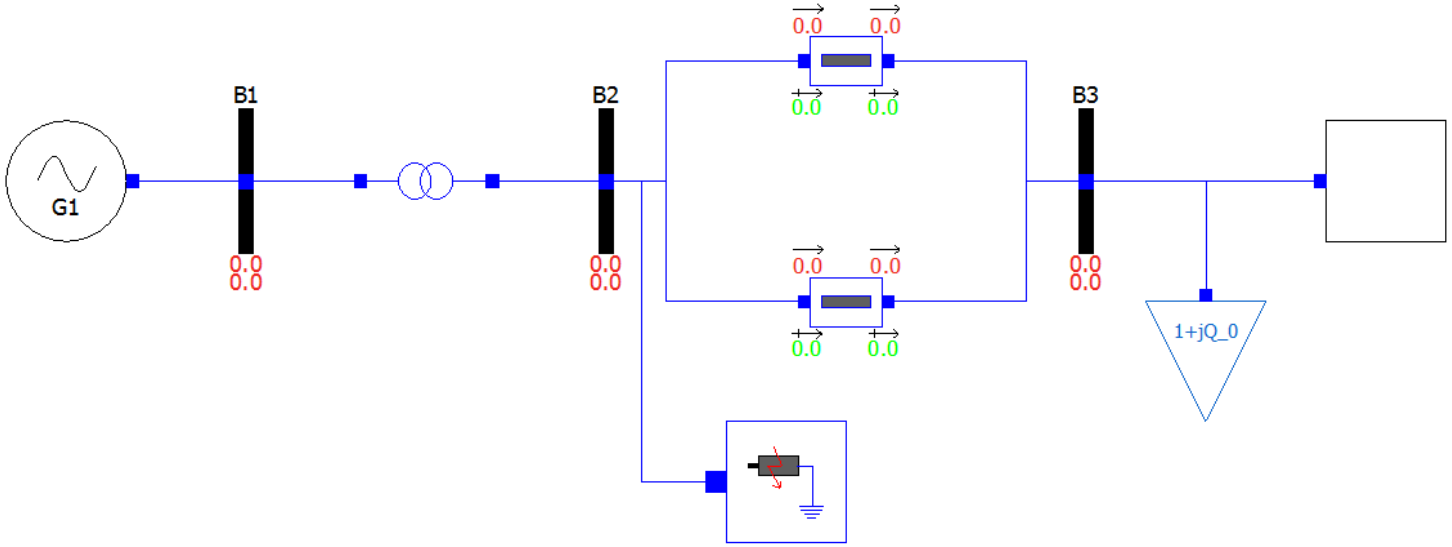


OMEdit and iPSL

- Now the library is loaded and its structure can be browsed in the Libraries Browser



Example 1*



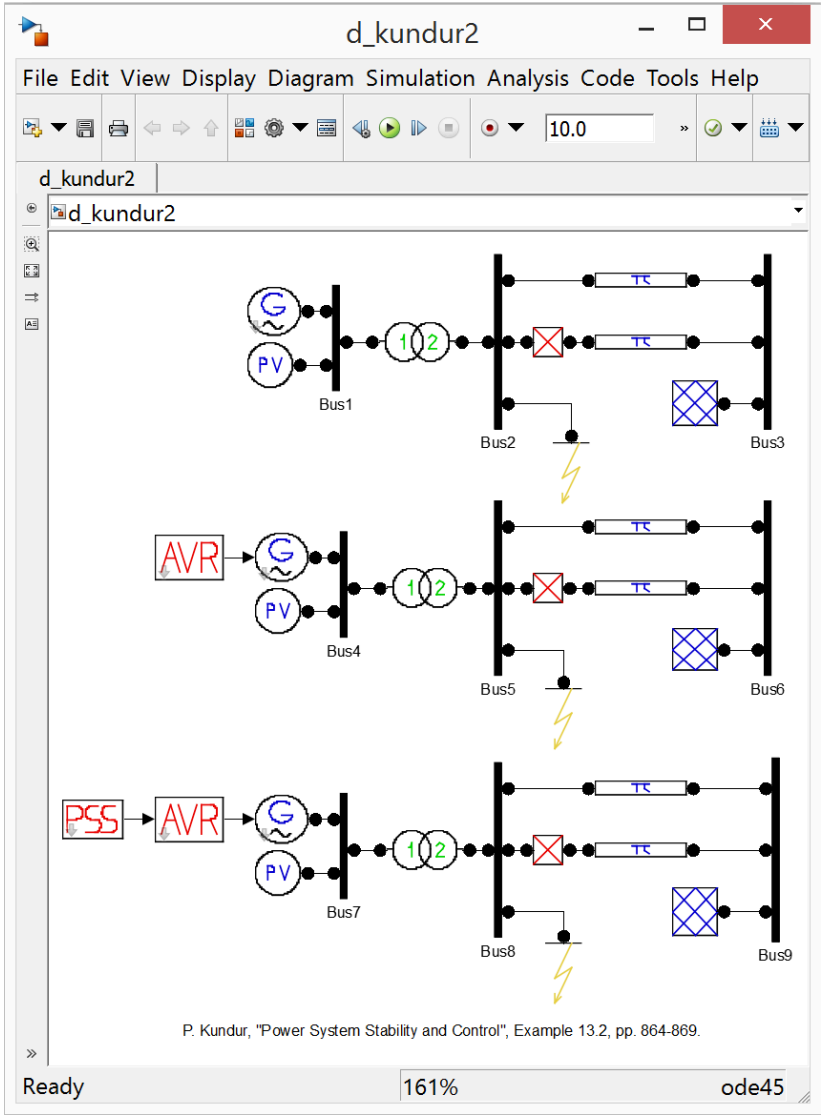
- Single Machine Infinite Bus (SMIB) system
- Analysis of the transient stability of the system including the effects of rotor circuit dynamics and excitation control
- Four machines represented by one connected via transformer and parallel lines to the infinite bus

* P. Kundur, "Power System Stability and Control", Example 13.2

Example 1

Power flow

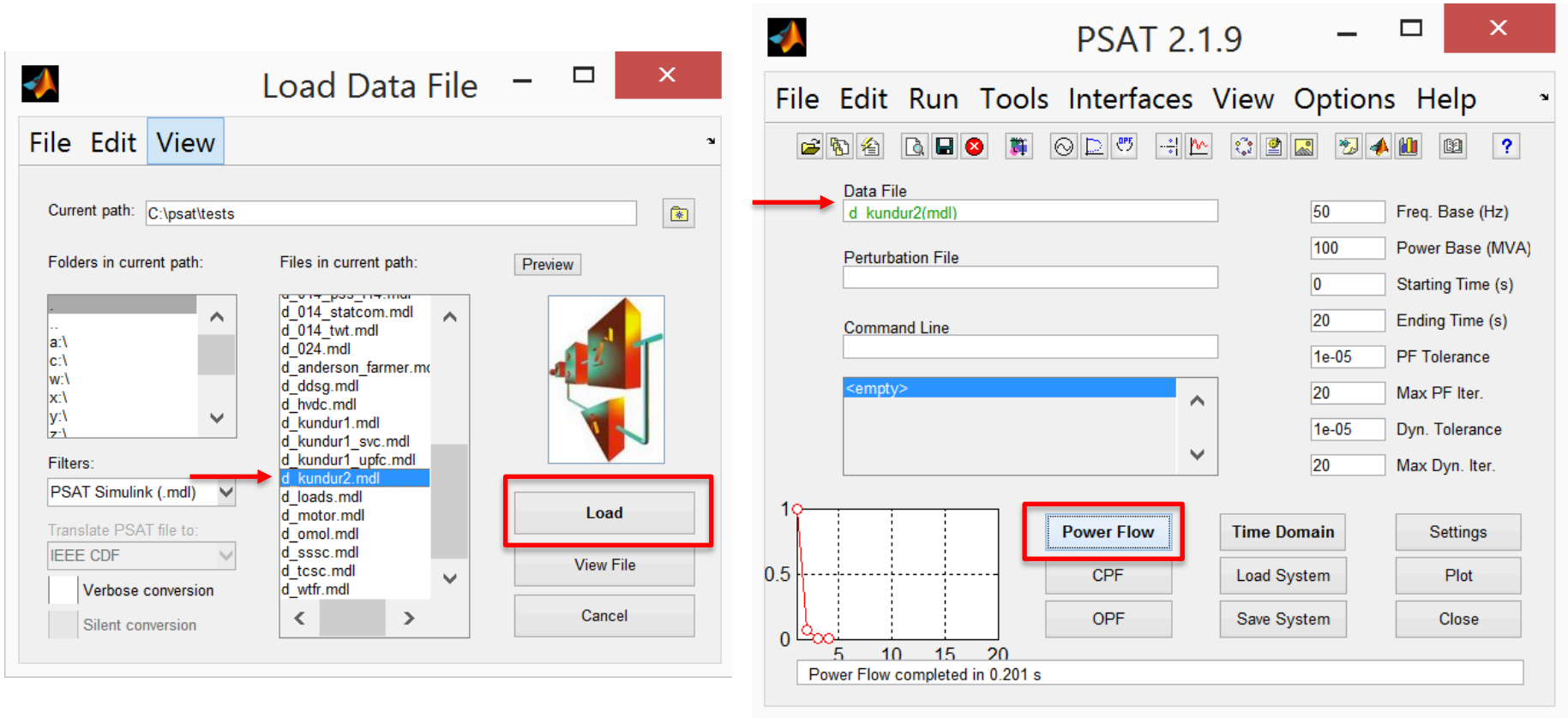
- Power flow results were obtained by PSAT
- Prepared Example 1 already exists in PSAT and can be used for power flow calculations and dynamic simulations



Example 1

Power flow

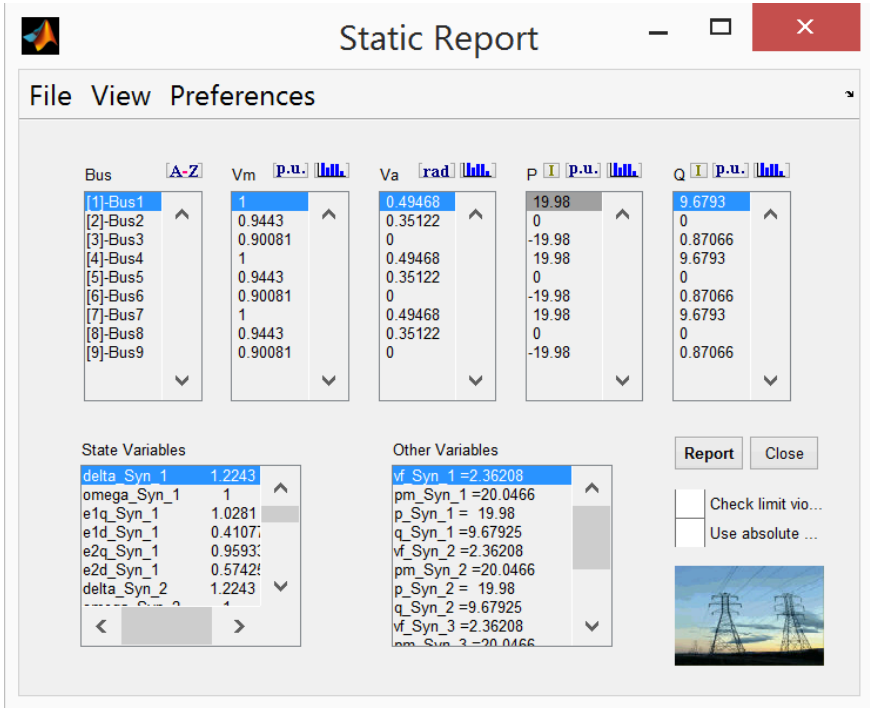
- Example 1 is loaded and the power flow calculations are executed



Example 1

Power flow

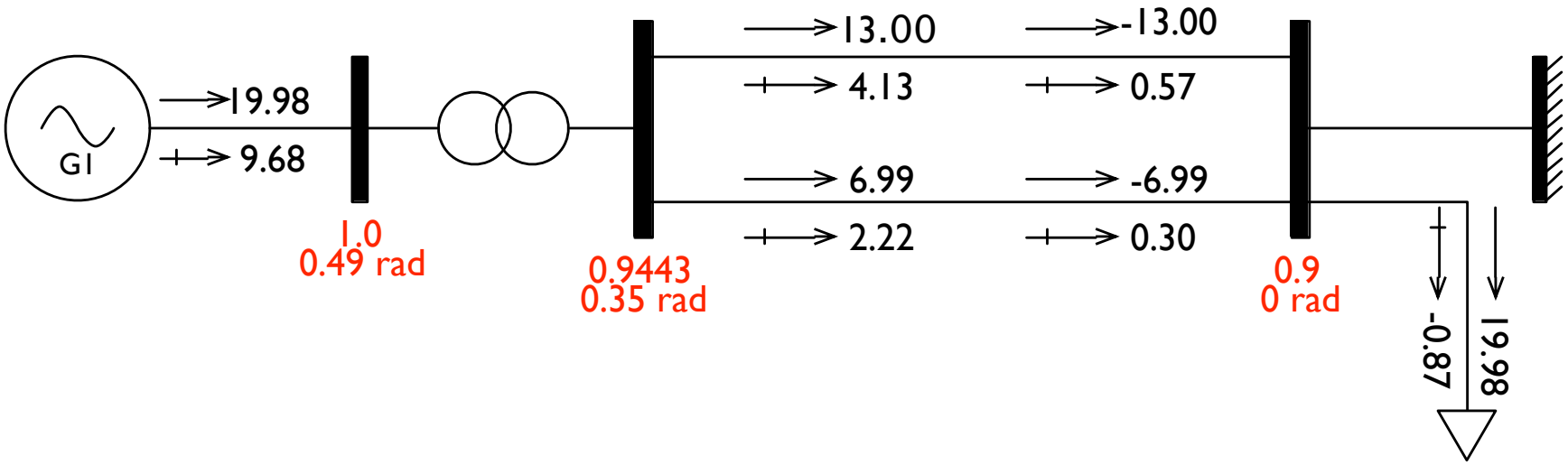
- Static Report can be access where all of the power flow results are listed along with the initial values of various state variables of the models
- In this tutorial, there is no need to run the power flow in PSAT since the data will be provided, but feel free to explore PSAT later



Example 1

Power flow

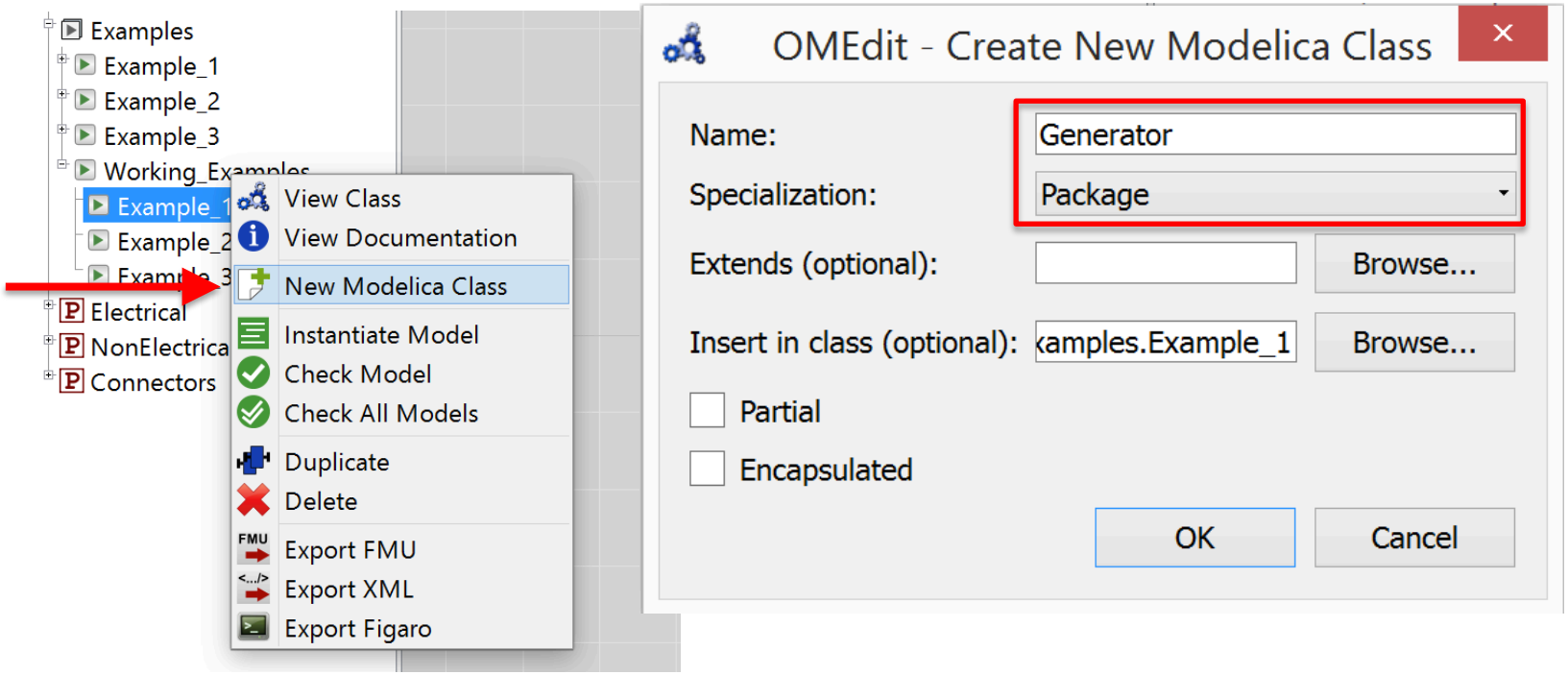
- The summary of all of the relevant data from the power flow is given on the figure below



Example 1

Generator model – Step 1

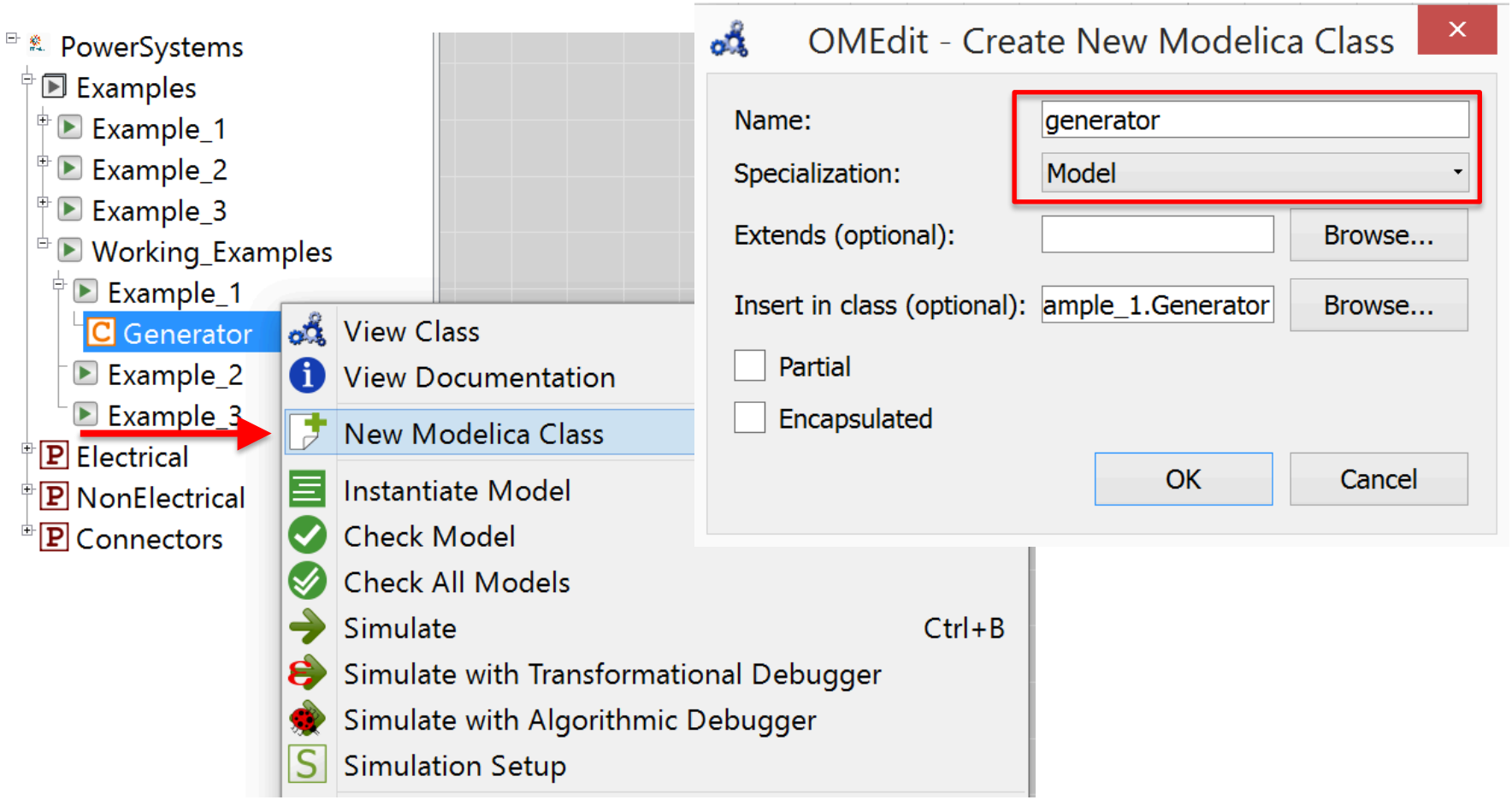
- First, the package where the generator model will be located has to be created
- This is done by right clicking on the *Example_1* in the *Working_Examples* package
- The package should be named *Generator*



Example 1

Generator model – Step 1

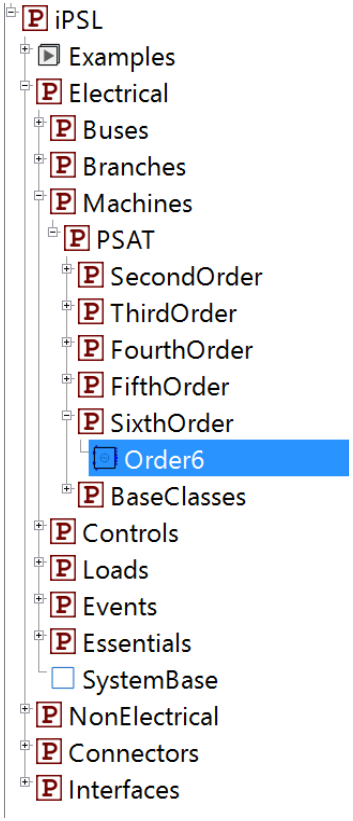
- Within the *Generator* package, model of the generator shall be created

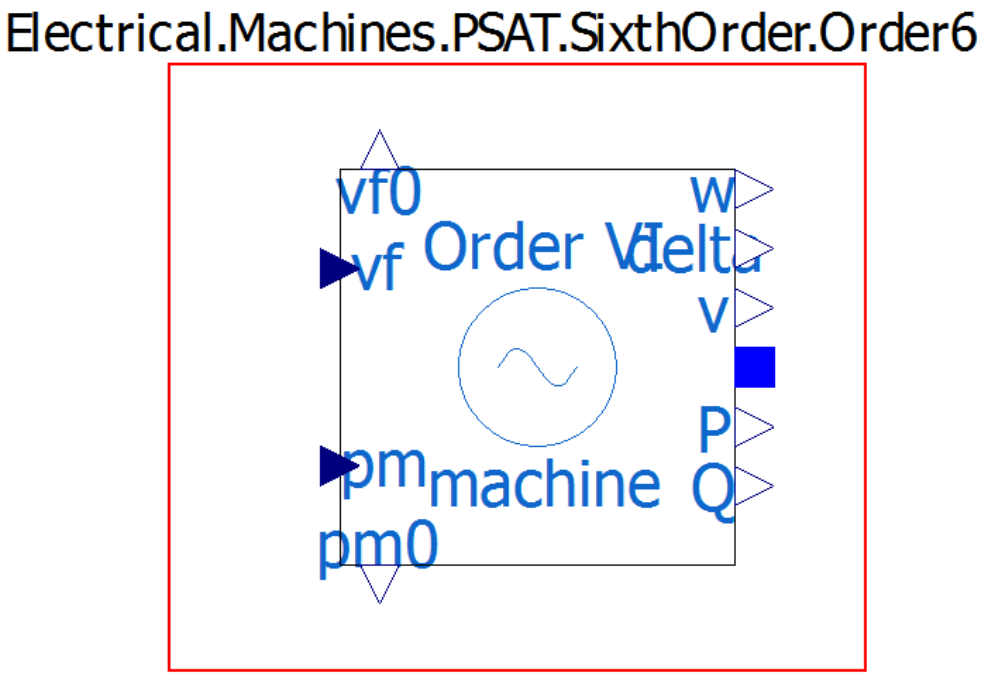


Example 1

Generator model – Step 1

- 6th order model of the generator from the PSAT is used
- The model is added by dragging the generator from the library and dropping it to the model



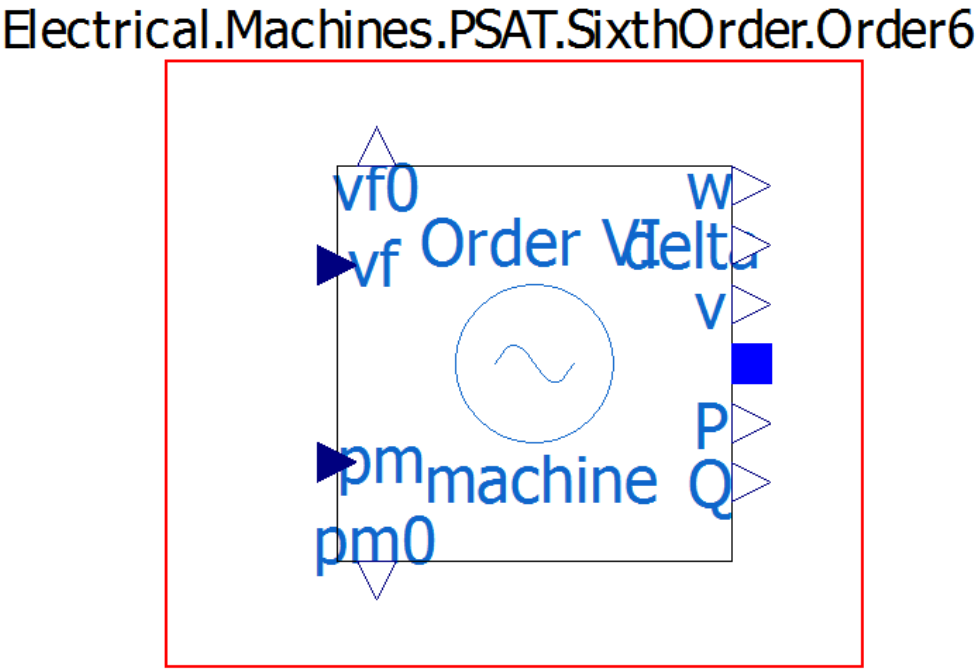


Example 1

Generator model – Step 1

- Parameters of the generator are given in the table

S_n	2220	x''_q	0.25
V_n	400	$T'_{d,0}$	8
r_a	0.003	$T'_{q,0}$	1
x_d	1.81	$T''_{d,0}$	0.03
x_q	1.76	$T''_{d,0}$	0.07
x'_d	0.3	T_{aa}	0.002
x'_q	0.65	M	7
x''_d	0.23	D	0

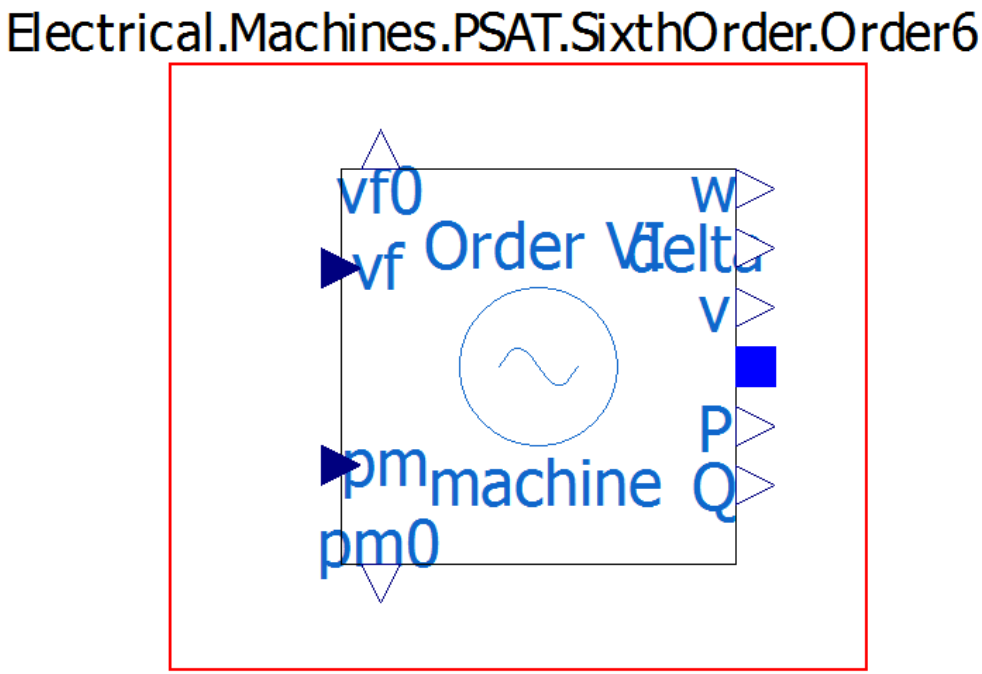


Example 1

Generator model – Step 1

- Power flow results:

V_0	1
$angle_0$	0.4946
P_0	19.9799
Q_0	9.6792
V_b	400
S_b	Do not edit
f_n	Do not edit



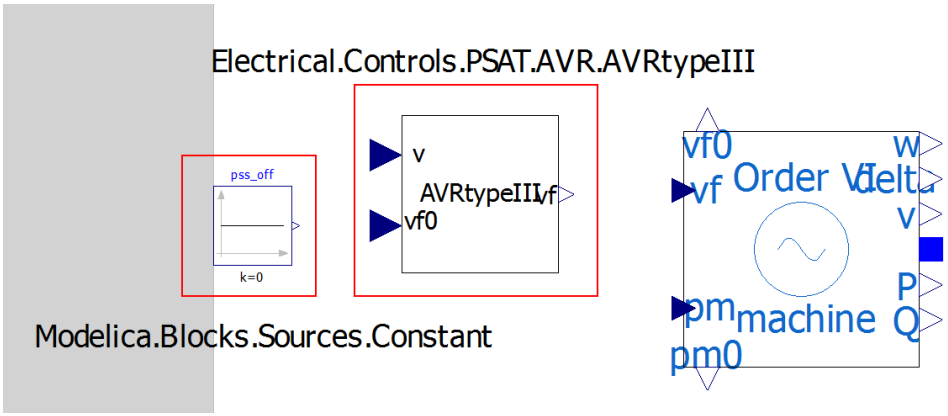
Example 1

Generator model – Step 2

- PSAT model of the AVR Type III is used
- Constant block pss_off will be used as a zero input to the PSS input signal of the AVR since the PSS is not used

- Parameters:

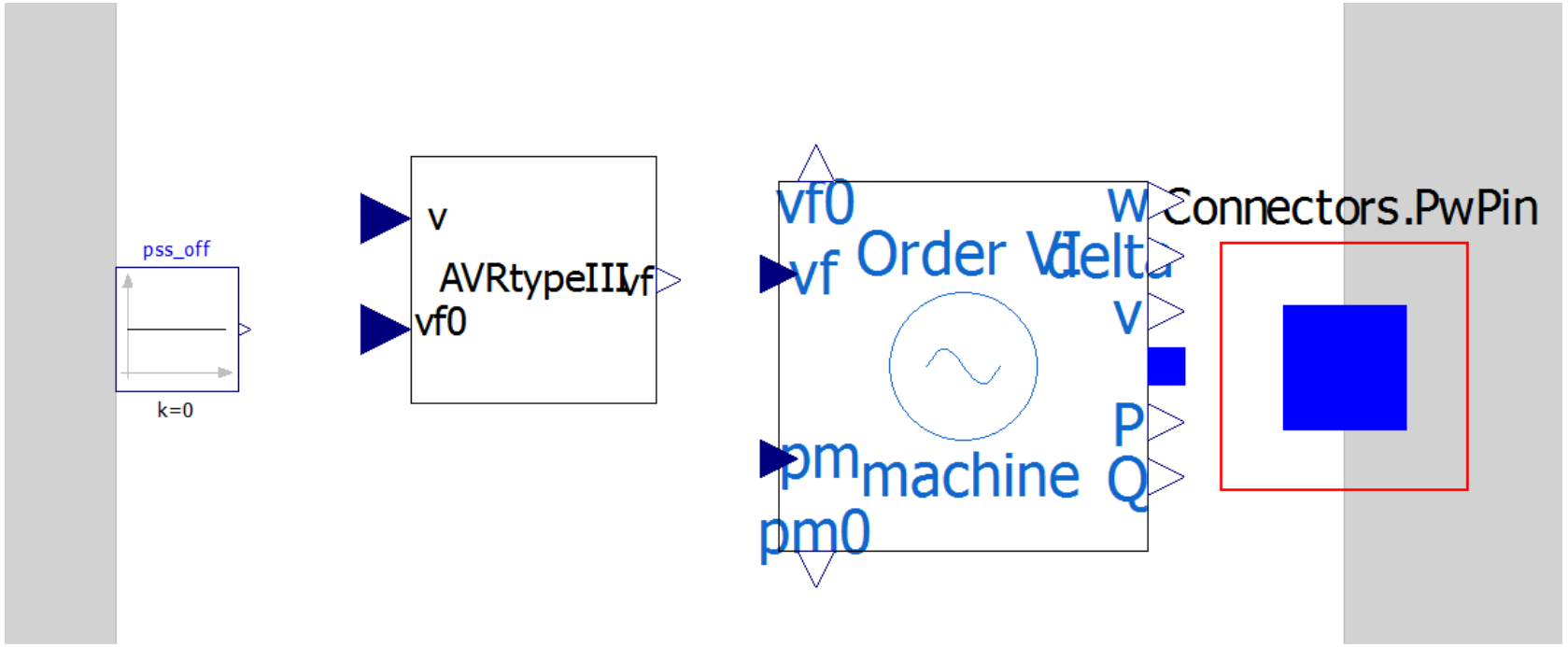
v_{ref}	1	T_1	1
v_0	1	T_e	0.0001
$v_{f,max}$	7	$v_{f,0}$	2.42
$v_{f,min}$	-6.4	T_r	0.015
K_0	200	s_0	0
T_2	1		



Example 1

Generator model – Step 3

- Additional PwPin will be used to interface generator model's voltages and currents with other components of the network model



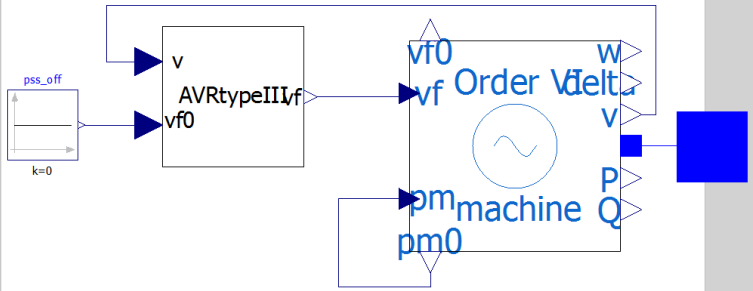
Example 1

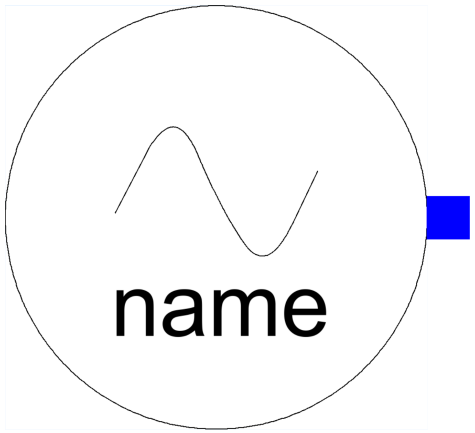
Generator model – Step 4

- To finish the generator model, different signals need to be connected
- Optionally, icon of the generator model can be altered

Step 4: Connecting model's signals

1. Machine's terminal voltage to AVR's input signal
2. AVR's output field voltage to machine's input field voltage
3. Initially calculated mechanical power to input signal of the machine's mechanical power
4. Machine's power terminal to the generator model power terminal
5. Constant pss_off to the PSS input at the AVR

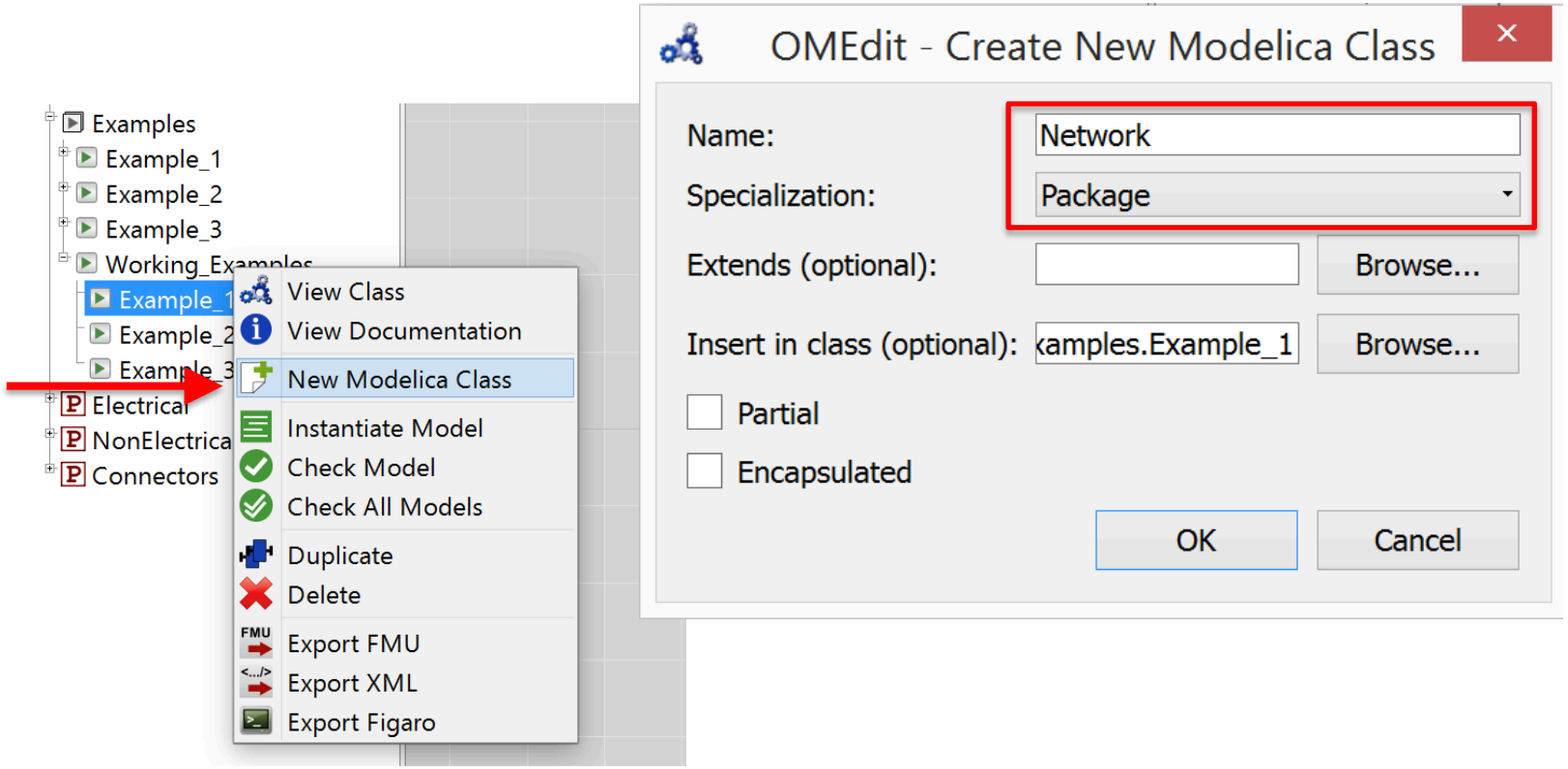




Example 1

Network model – Step 1

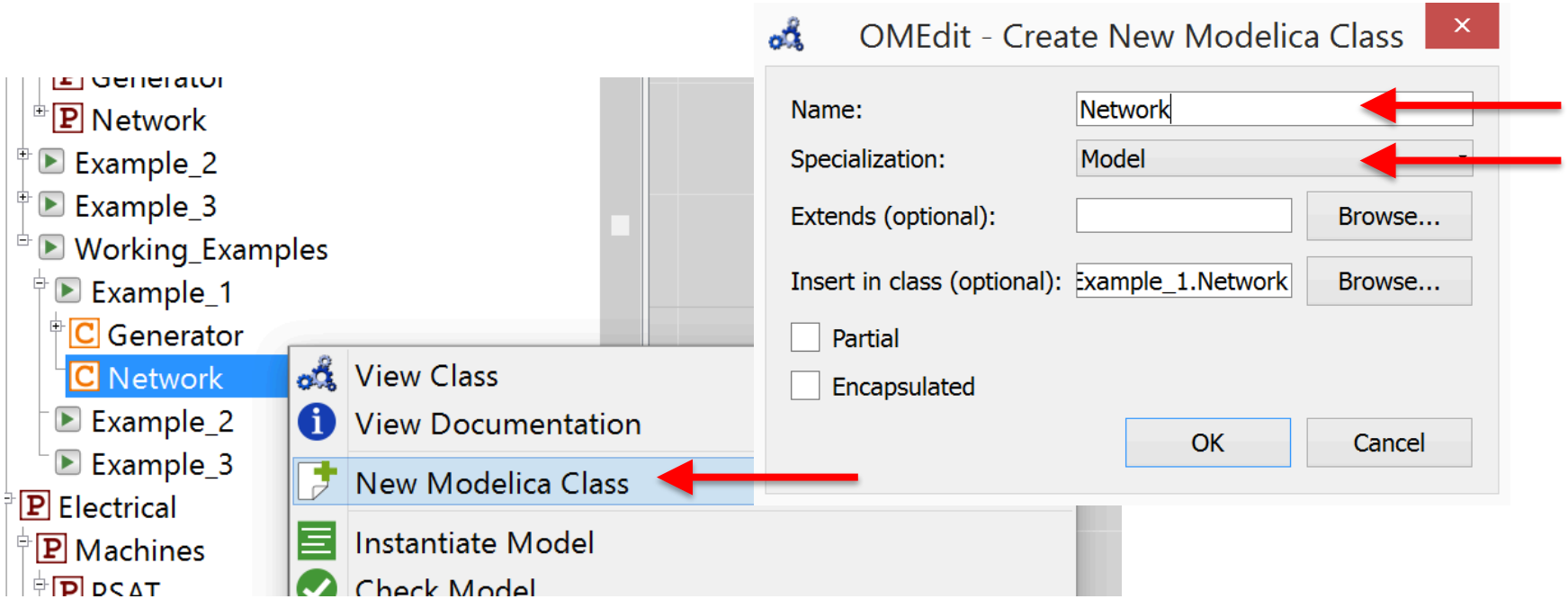
- Network package will be created in the *Example_1* package
- This package is created by right clicking on the *Example_1* in the *Working_Examples* package



Example 1

Network model – Step 1

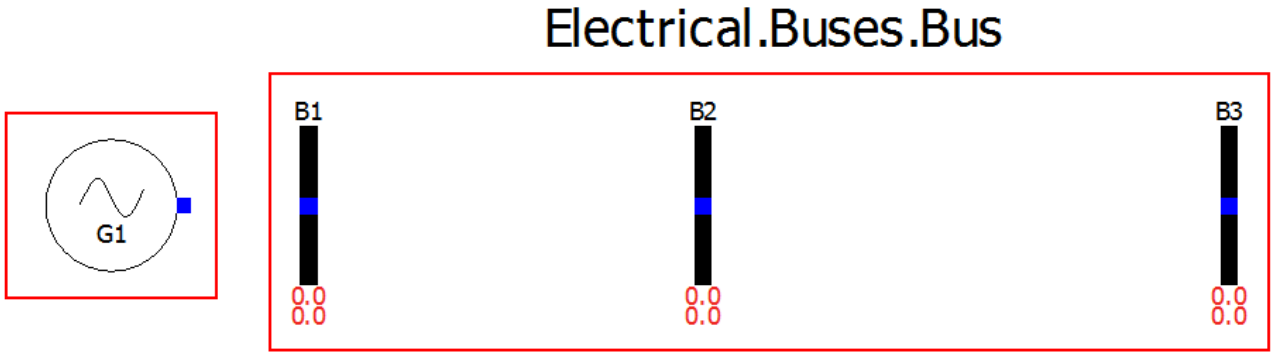
- Network model will be created in the *Network* package
- This package is created by right clicking on the *Network* package
- The name of the network model will be *Example_1*



Example 1

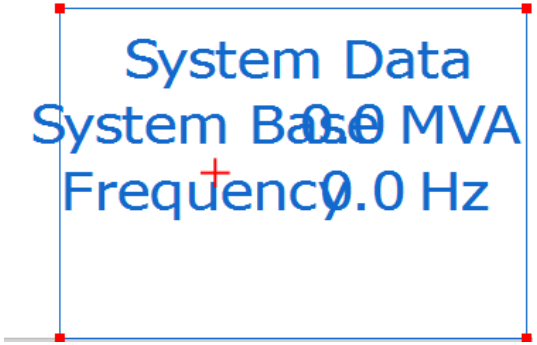
Network model – Step 1

- Created generator model and three bus models are added to the network model



- Also, model iPSL.Electrical.SystemBase shall be added to the network model which defines base parameters for all of the components in the network model

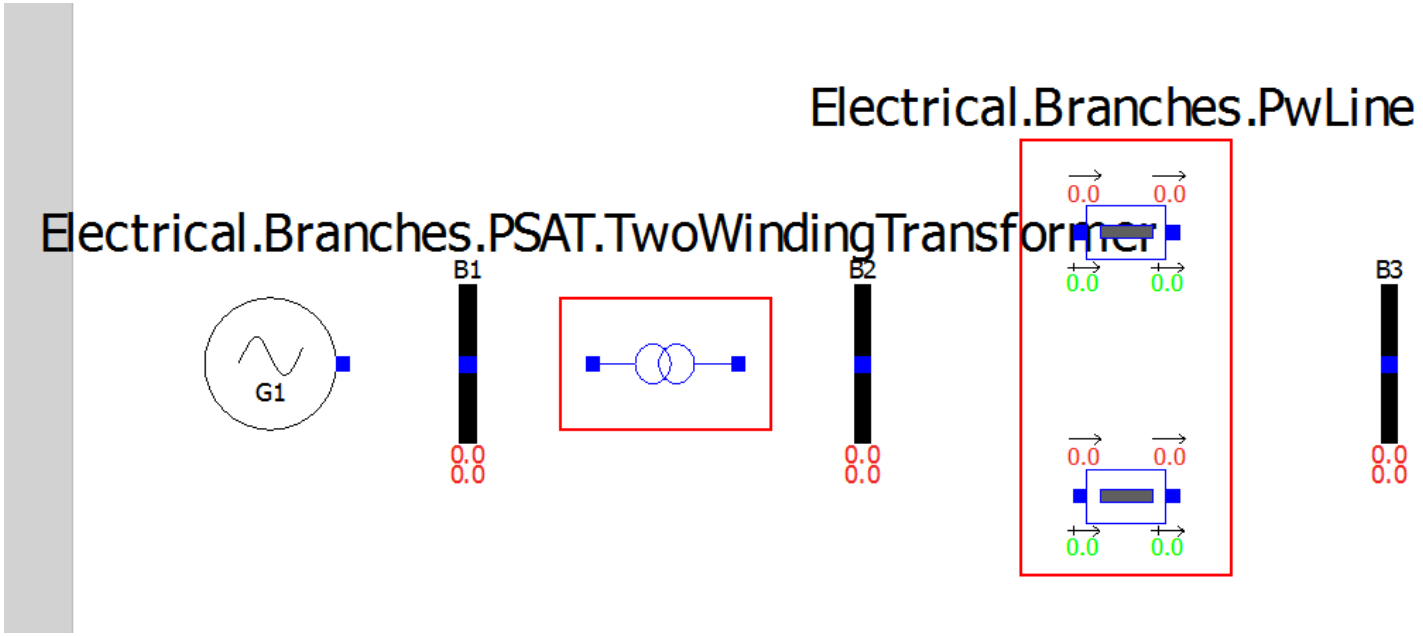
S_b	100
f_n	60



Example 1

Network model – Step 2

- Transformer and line models are added



Example 1

Network model – Step 2

- Transformer and line parameters

Transformer

S_b	100	f_n	60
S_n	2220	kT	1
V_b	60	x	0.15
V_n	0.001	r	0

Line 1

R	0.0	G	0.0
X	0.5*100/2220	B	0.0
S_b	100		

Line 2

R	0.0	G	0.0
X	0.93*100/2220	B	0.0
S_b	100		

Example 1

Network model – Step 3

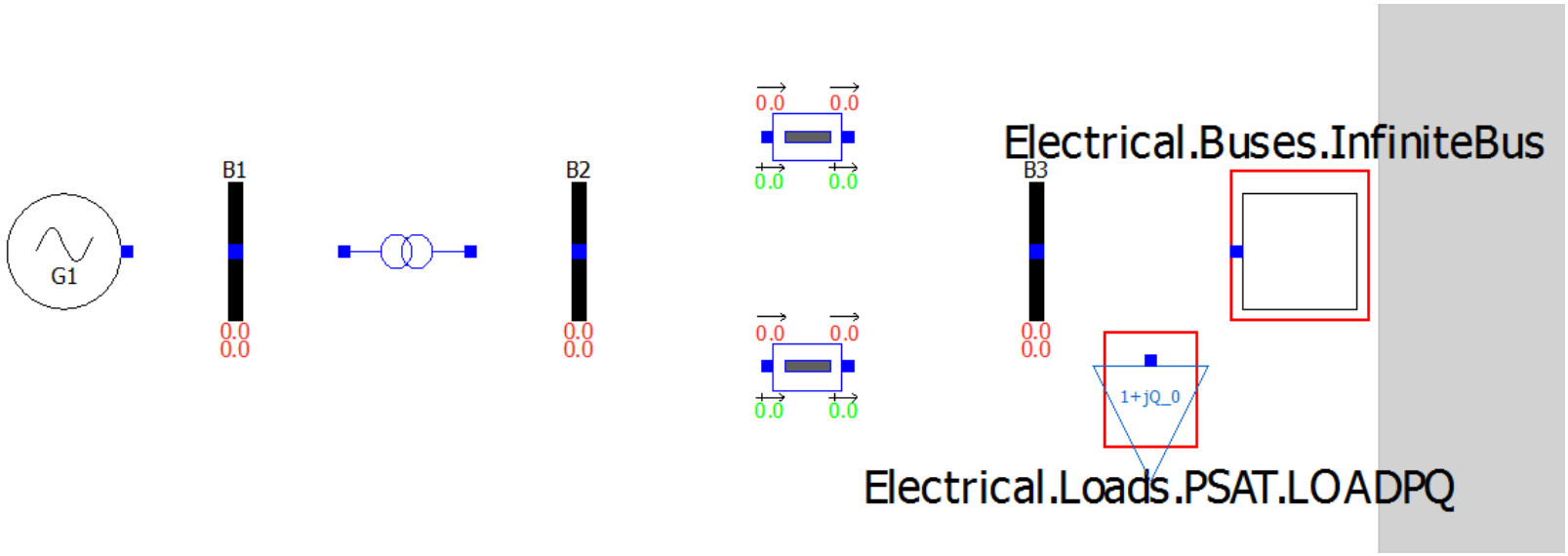
- Infinite bus and load models are added

Load

S_b	100	P_0	19.9799
S_n	100	Q_0	-0.8706

Infinite bus

V	0.90081	$angle$	0



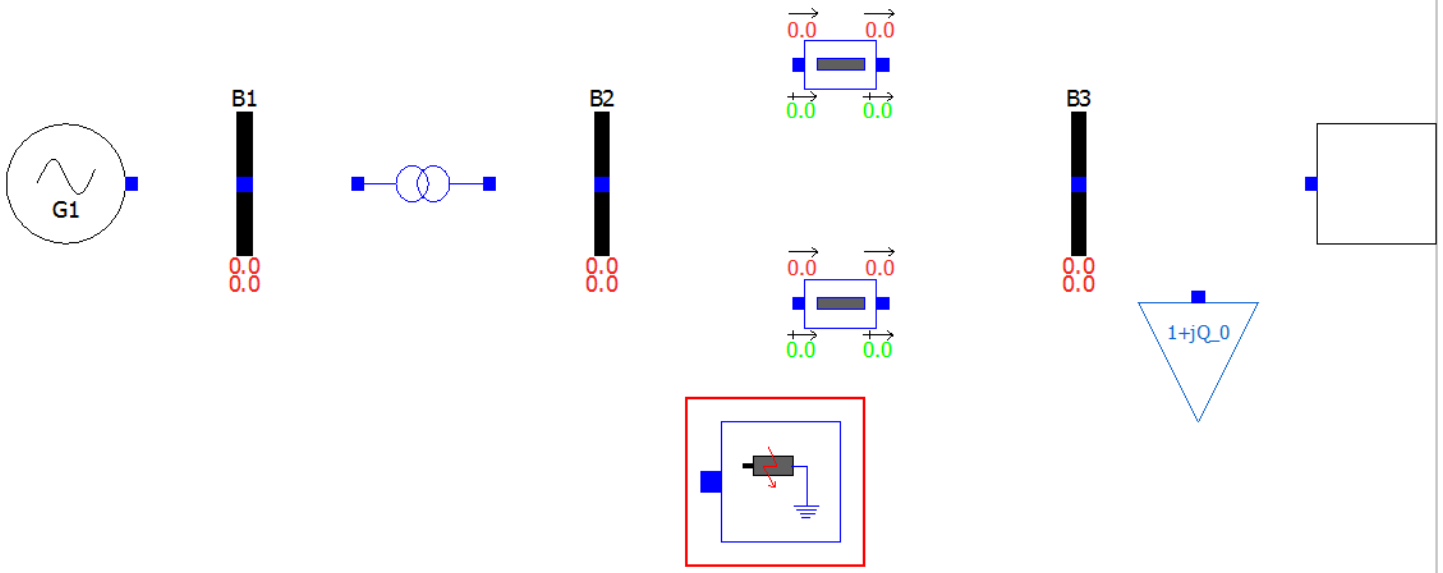
Example 1

Network model – Step 4

- 3-phase-to-ground fault is added

Fault

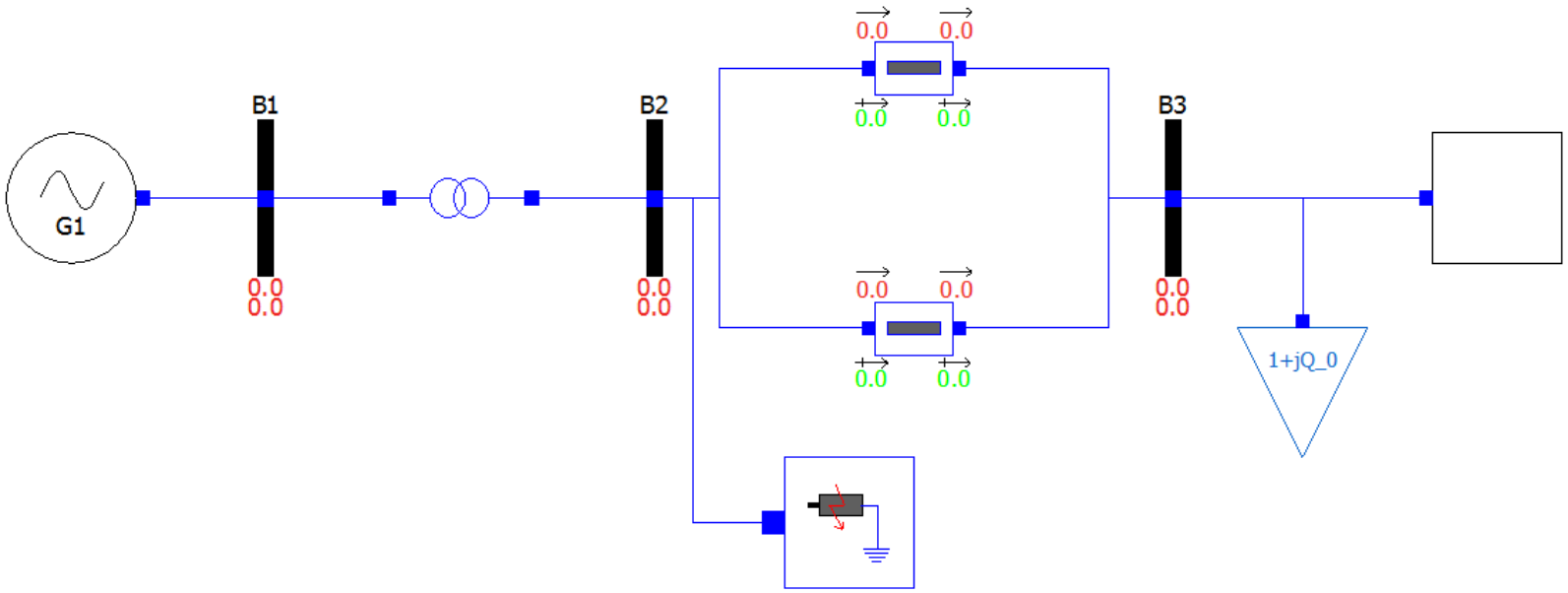
R	0	t_1	0.5
X	0.01*100/2220	t_2	0.57



Example 1

Network model – Step 5

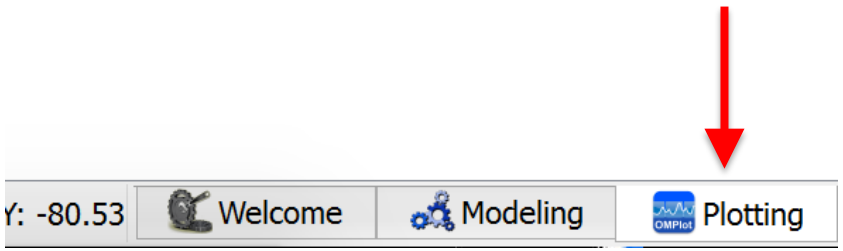
- The network model is completed by connecting all of the components
- Now, the model can be simulated and linearized



Example 1

Simulation

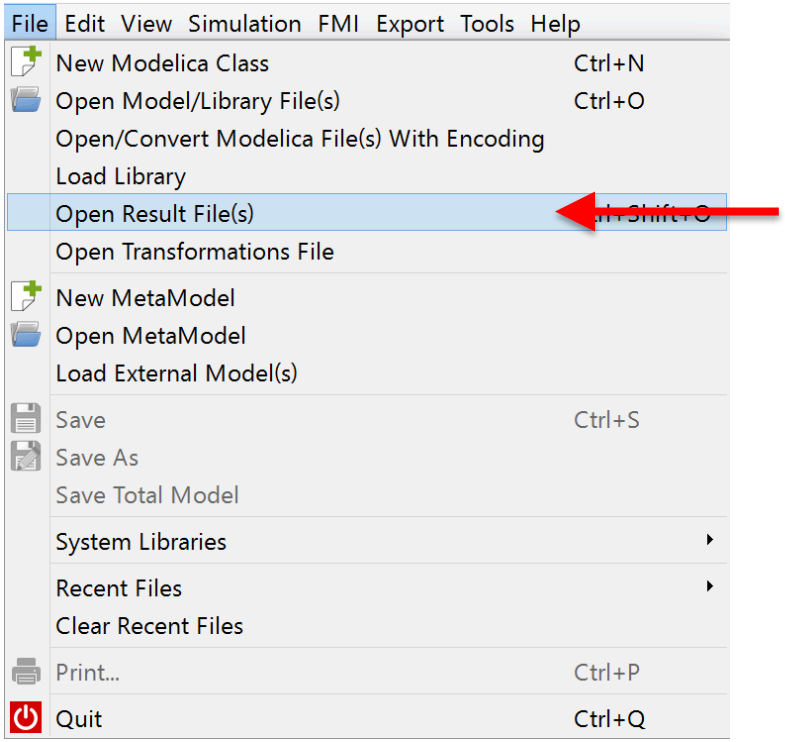
- System will be simulated with 3-phase-to-ground fault at t=0.5s with a duration of 70ms
- Simulation results will be compared with the reference results from the PSAT that will be loaded first
- PSAT results are provided in a file “PSAT_dyn.csv”
- To load the file, the view should be switched to “Plotting” tab

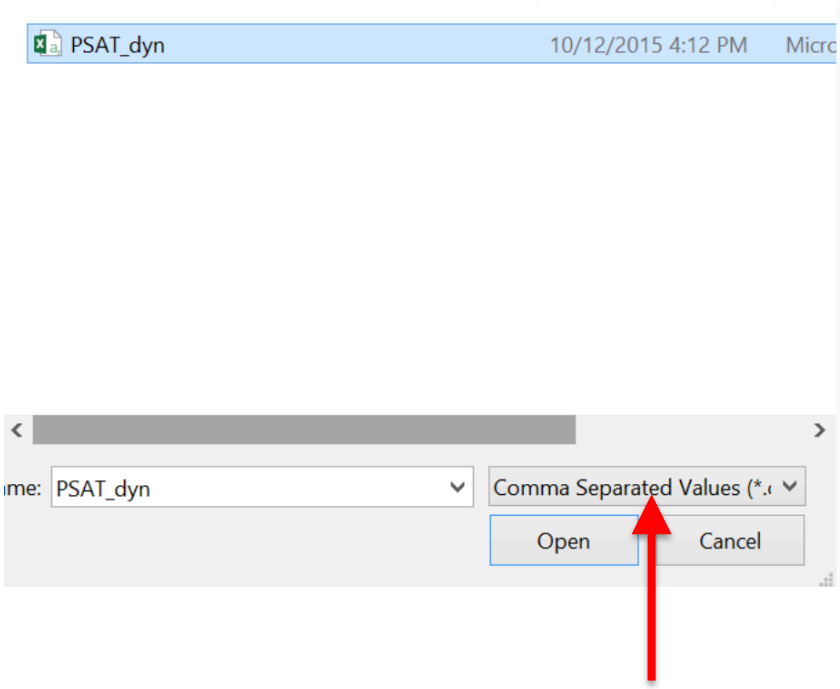


Example 1

Simulation

- Result file can be opened by navigating the menu to File->Open Resut File(s)
- In the pop-up menu, one has to select “Comma Separated Values” as a file type, navigate to the directory where the file is located and open it

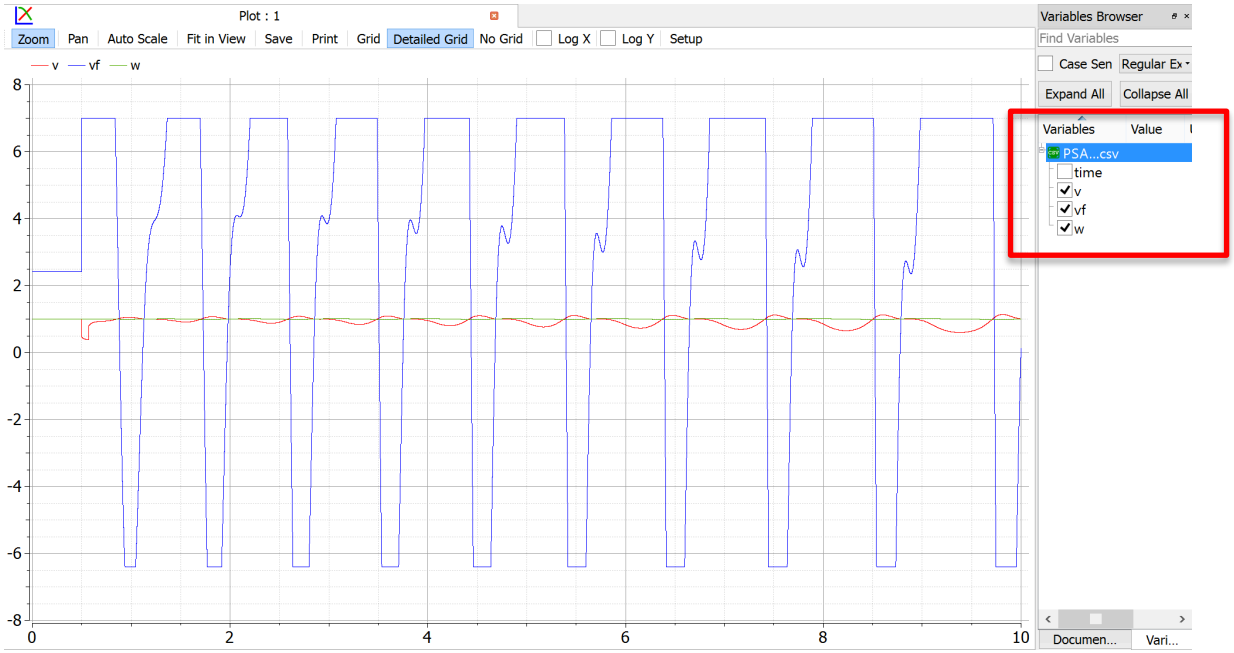




Example 1

Simulation

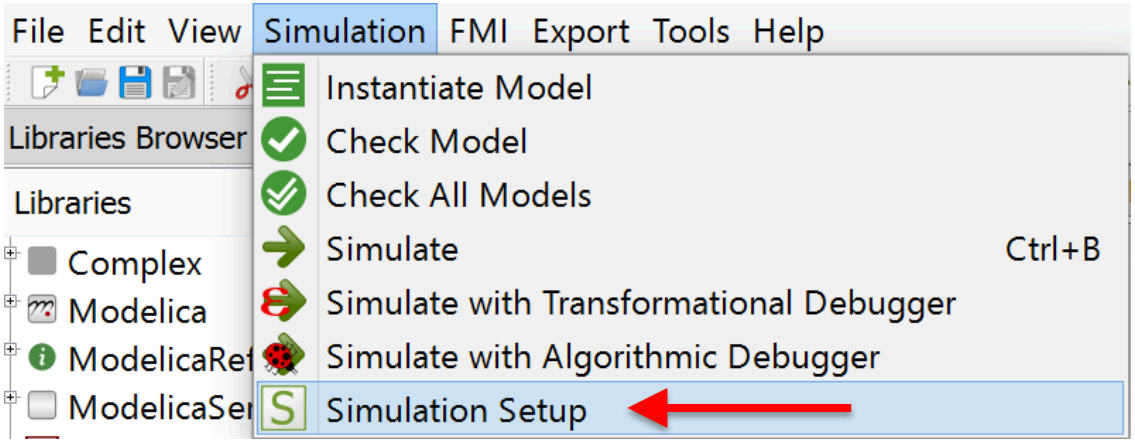
- In the variable browser, three waveforms from the PSAT results are loaded which can be displayed on the plot as it is shown in the figure below
- Loaded waveforms are generator terminal voltage, excitation field voltage and the generator speed



Example 1

Simulation

- Before the simulation, solver and its parameters are set to be the same as in the PSAT
- Solver is chosen to be Runge-Kutta 2 with a fixed step
- More solvers can be chosen in Modelica (depending on the tool), however, to match the model's response with the one in PSAT choice of the solver is limited

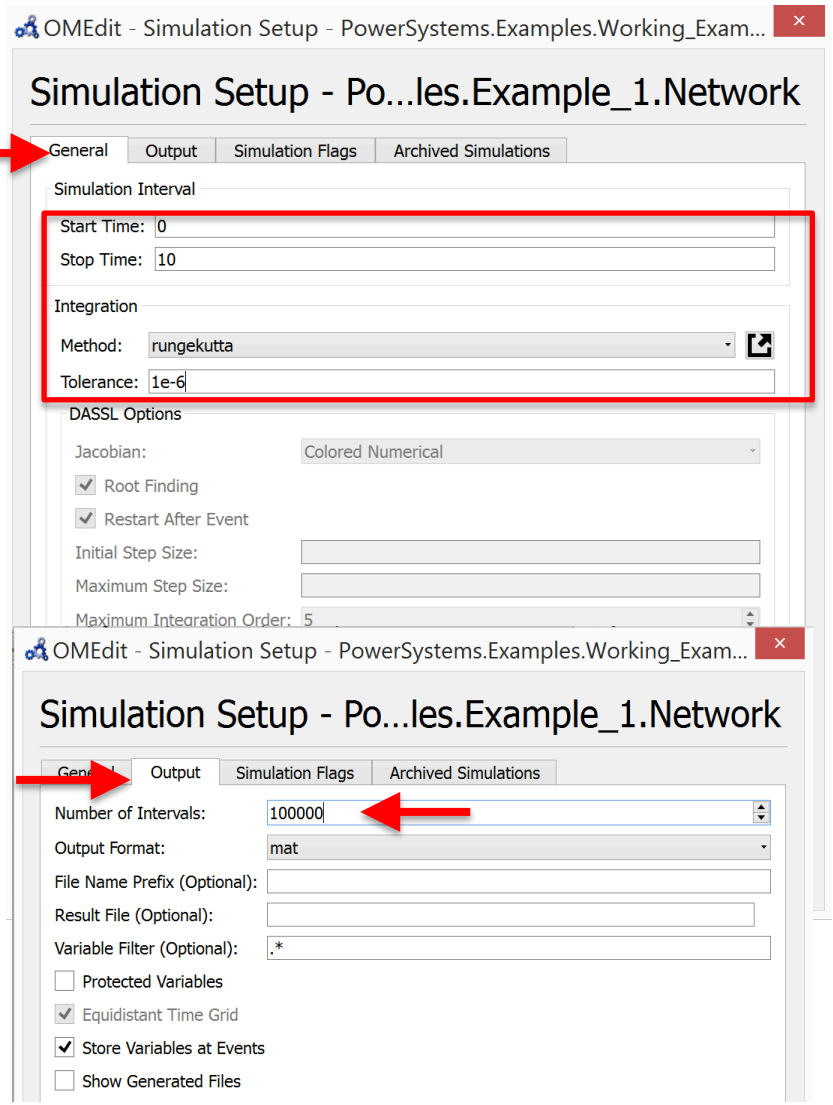


Example 1

Simulation

- Simulation time is set to 10s and the tolerance of the solver is set to 1e-6
- The size of the time step is set by the number of intervals
- Number of intervals:

$$n_{int} = \frac{Simulation\ Time}{Time\ Step} = \frac{10}{1e^{-4}} = 10^5$$

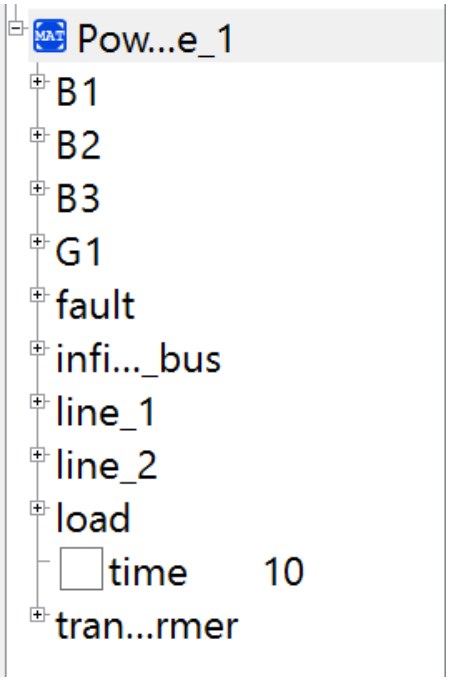


Example 1

Simulation

- By pressing the “Simulate” button on the toolbar, simulation of the model is executed

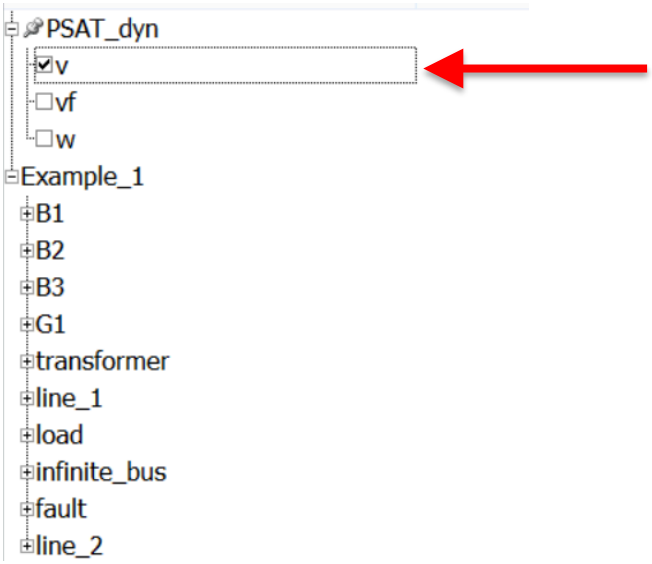
- Once the simulation is completed, Variable Browser is populated with the simulation results

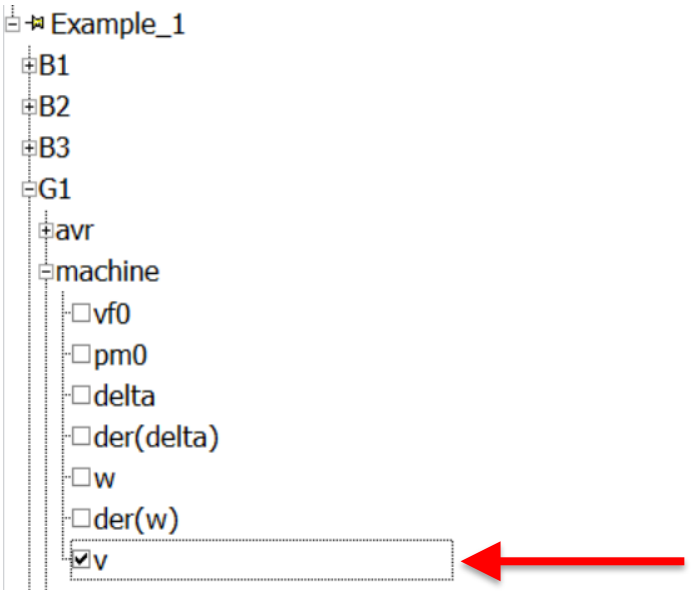


Example 1

Simulation

- To display the simulation results or compare it with the results from PSAT, one can mark the check-box next to the variable which will be shown on the plot
- To show the terminal voltage of the generator in PSAT and modelica, variables “PSAT_dyn.v” and “Example_1.G1.machine.v” have to be selected

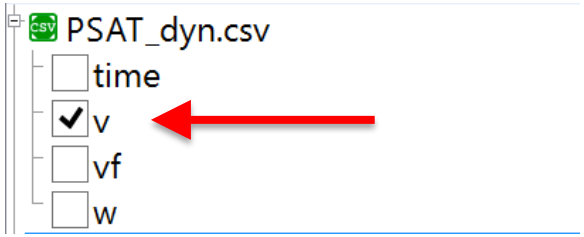


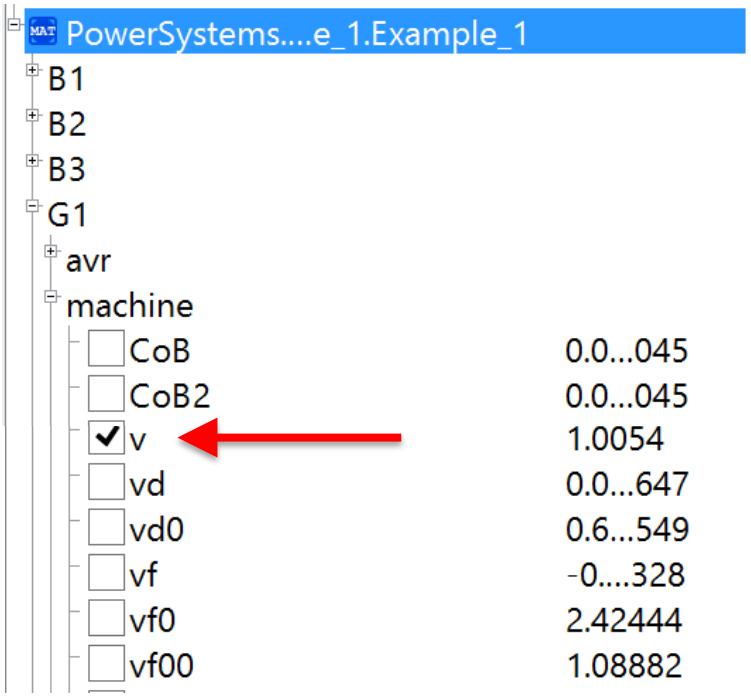


Example 1

Simulation

- To display the simulation results or compare it with the results from PSAT, one can mark the check-box next to the variable which will be shown on the plot
- To show the terminal voltage of the generator in PSAT and modelica, variables “PSAT_dyn.csv.v” and “G1.machine.v” have to be selected

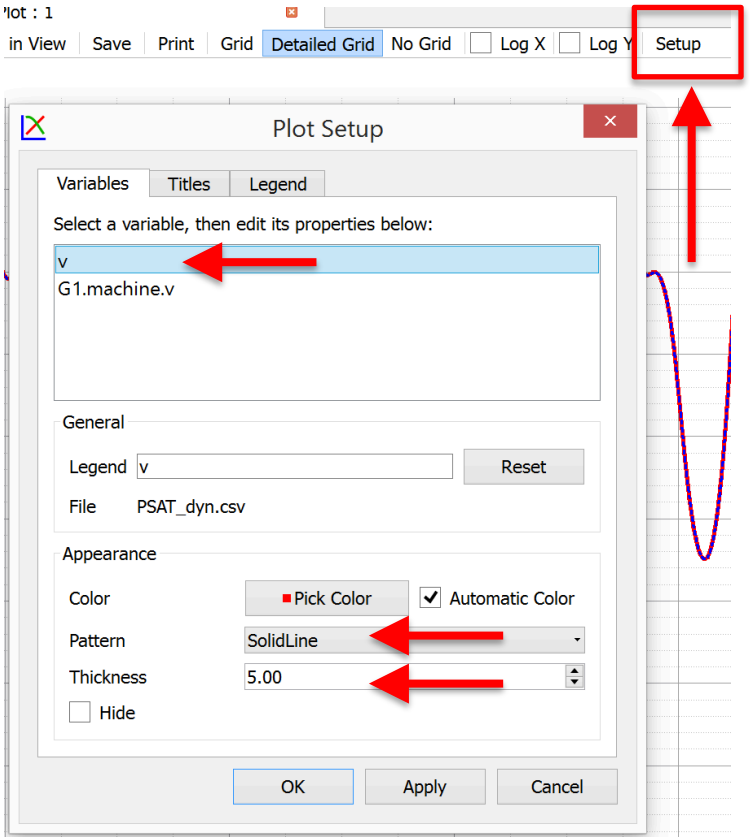


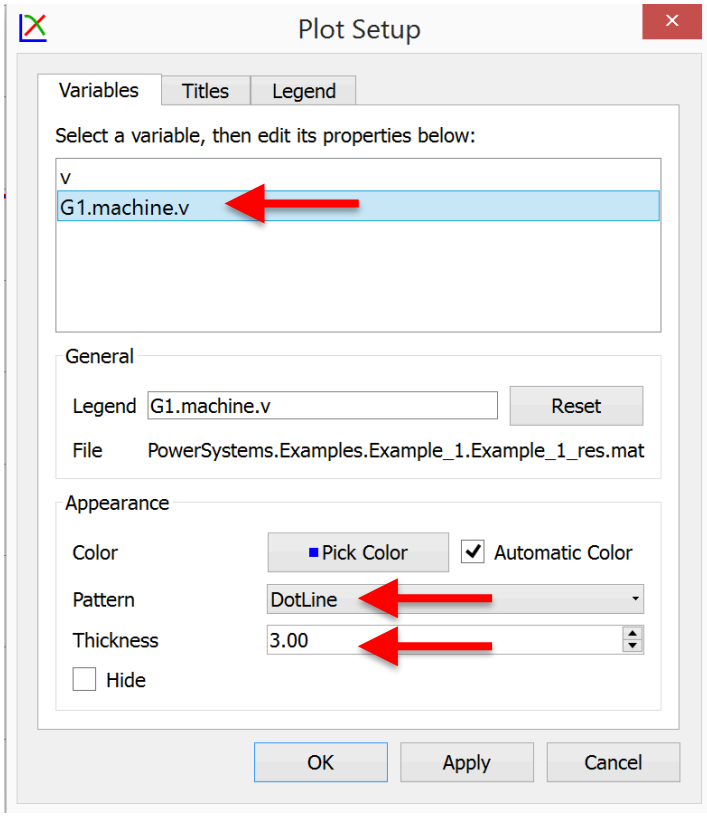


Example 1

Simulation

- To be able to destinguish different signals, one should adjust the thickness and the patter of the signal line

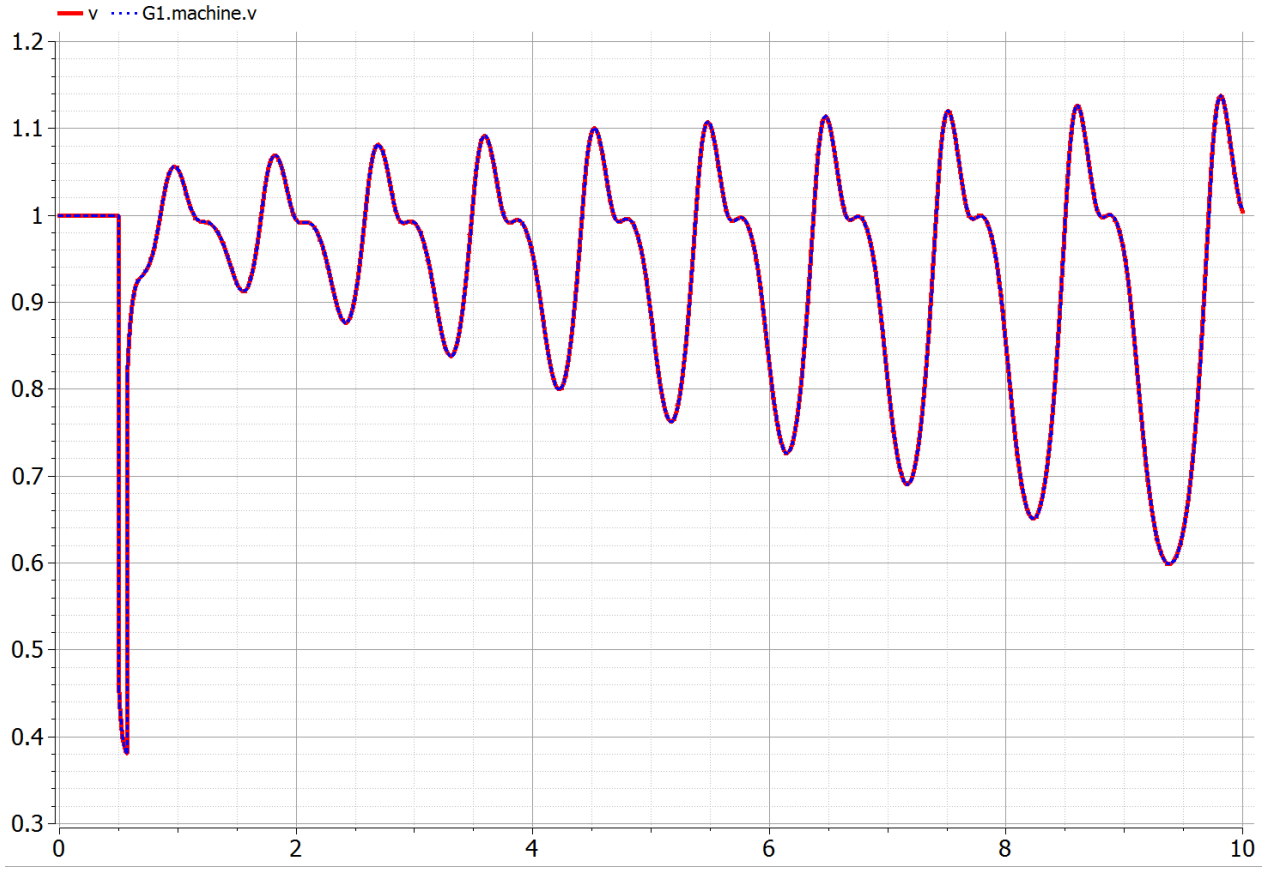




Example 1

Simulation

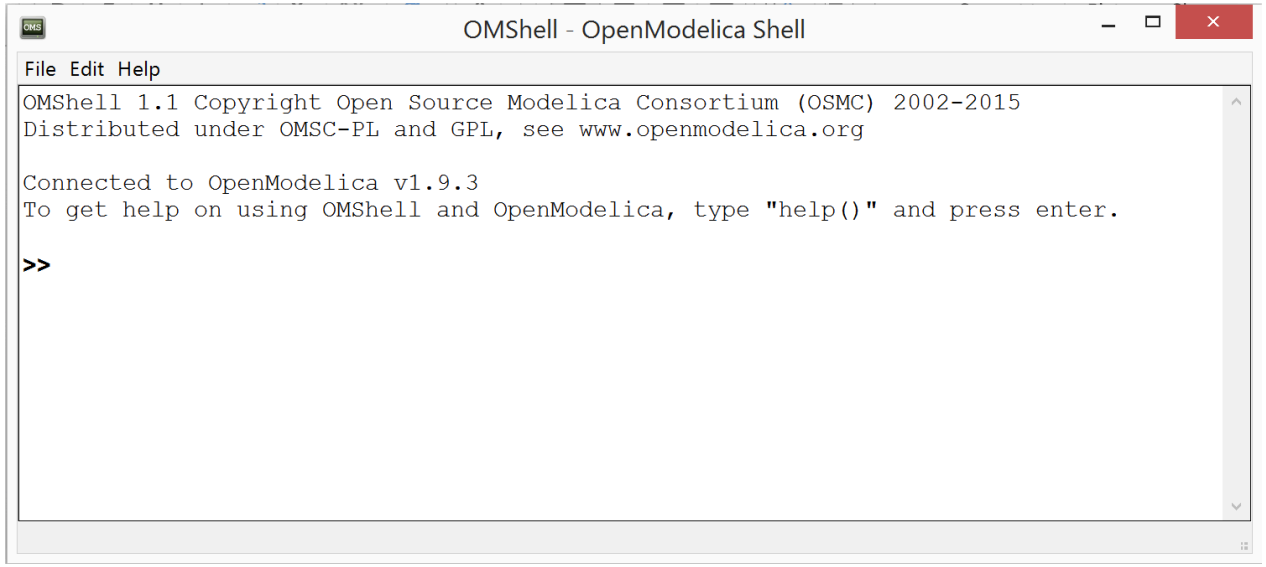
- Previous steps produce the plot shown in the figure below showing that the Modelica produces the same simulation results as the PSAT does



Example 1

Linearization

- To linearize the system, OpenModelica scripting will be needed
- Along with the library, script `linearize_iPSL.mos` was provided which performs the linearization procedure and provides the A matrix from the linearized state-space model
- The scripting shall be done in the command line interface of OpenModelica called OMShell



Example 1

Linearization

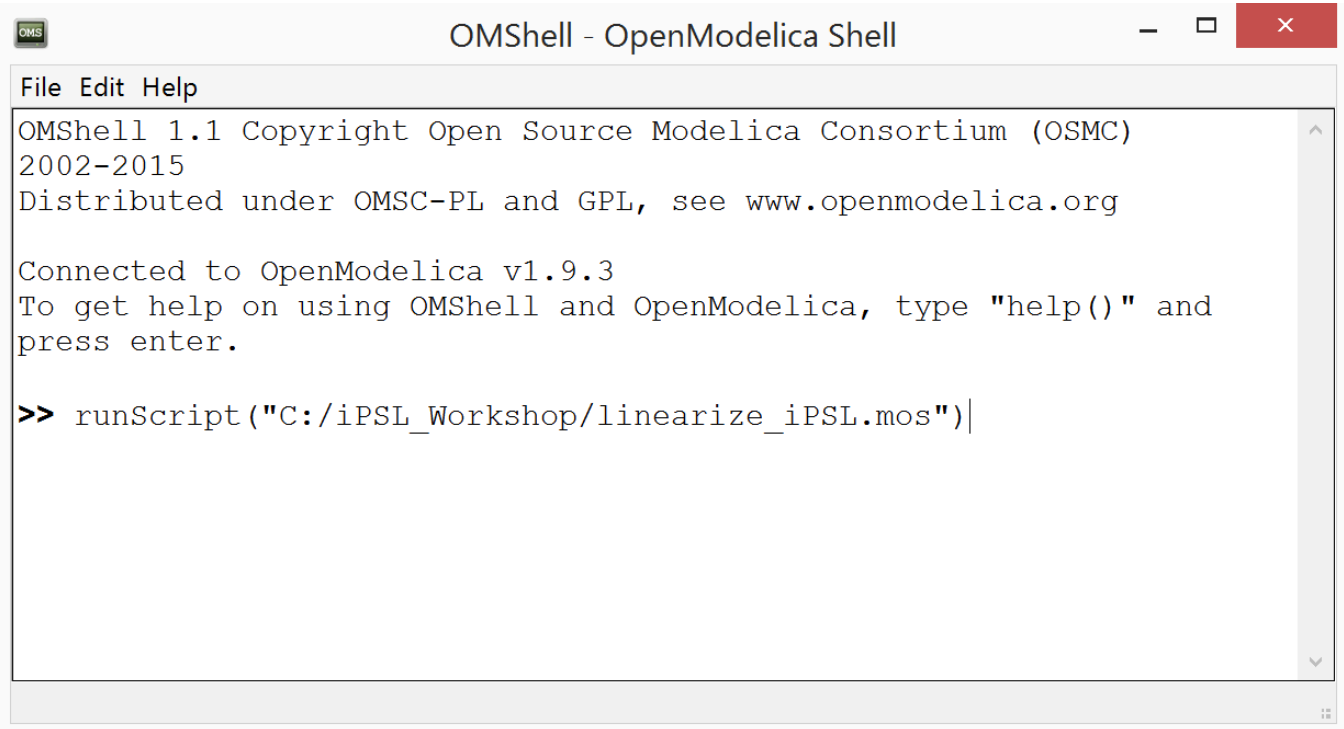
- First, open the script `linearize_iPSL.mos` with a text editor and make sure that the path in the second line points to the iPSL distribution on your hard drive

```
1   loadModel(Modelica);
2   loadFile("C:/iPSL_Workshop/iPSL_stripped.mo");
3   instantiateModel(iPSL;
4   instantiateModel(iPSL.Examples.Example_1.Example_1);
5
6   linearize(iPSL.Examples.Example_1.Example_1,stopTime=0.0);
7
8   loadFile("linear_iPSL.Examples.Example_1.Example_1.mo");
9
10  (a) := getParameterValue(linear_iPSL_Examples_Example__1_Example__1,"A");
```

Example 1

Linearization

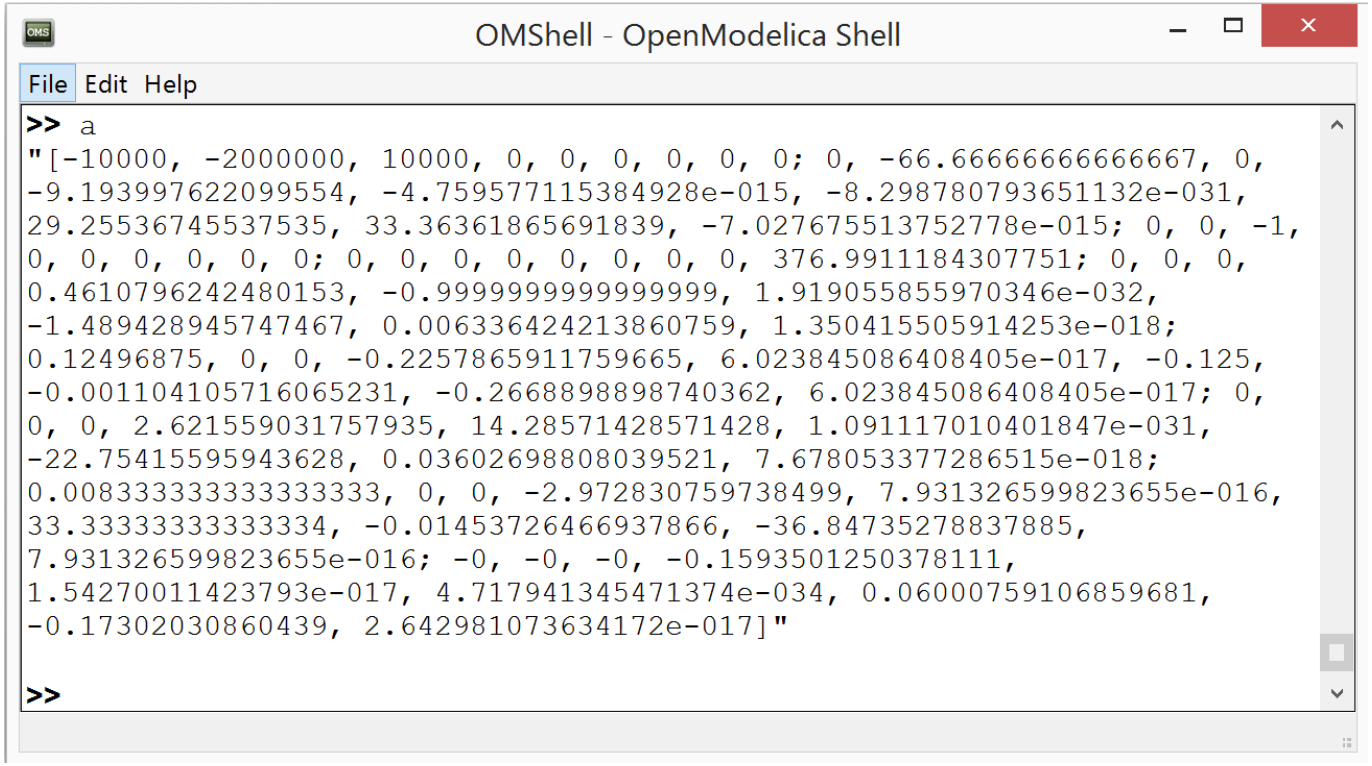
- To linearize system, Modelica script `linearize_iPSL.mos` is run by executing a command `runScript("C:/iPSL_Workshop/linearize_iPSL.mos")`



Example 1

Linearization

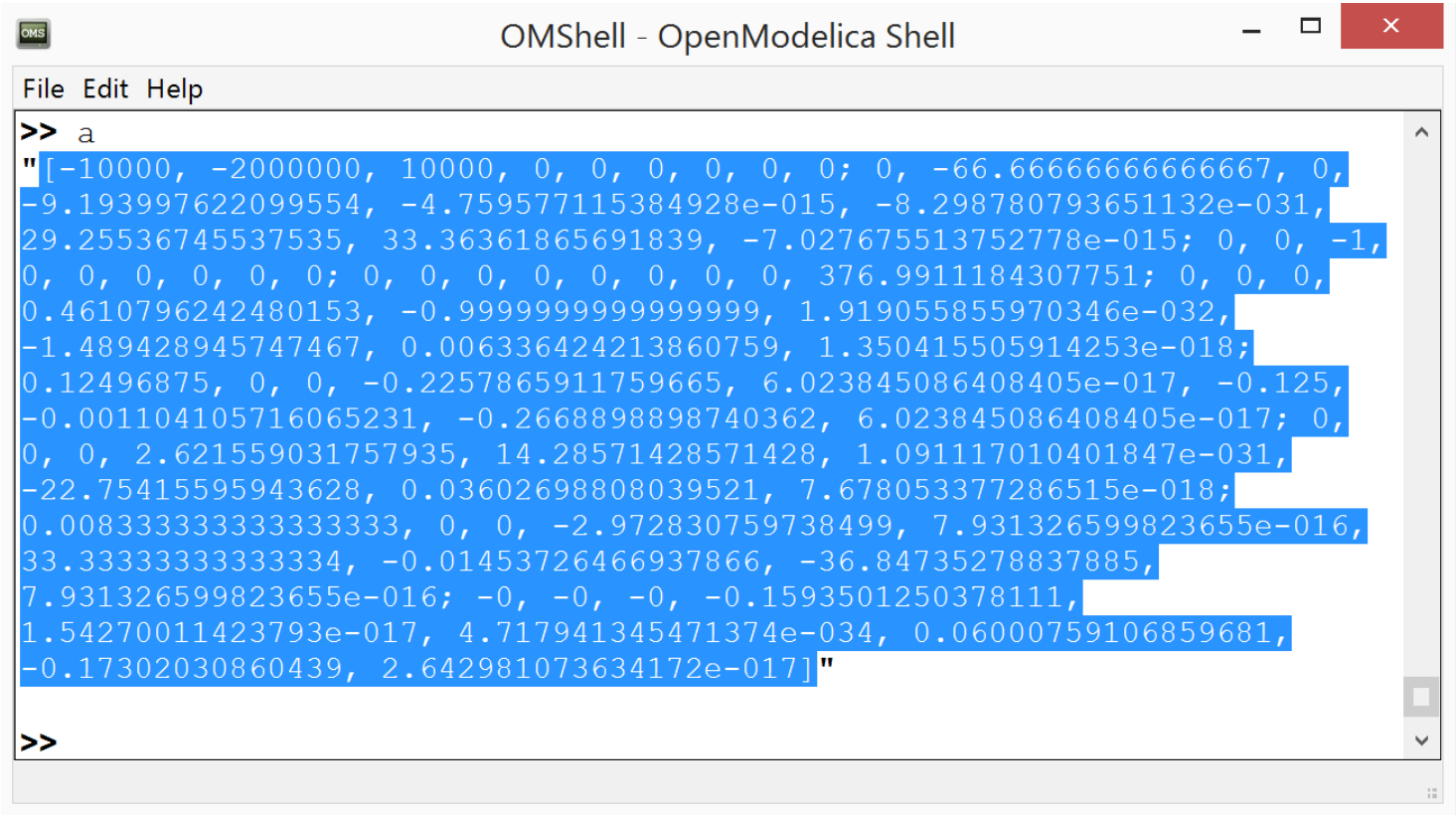
- Upon the execution, the A matrix of the linearized state-space model will be saved in the variable `a` as a string
- By typing `a` and pressing Enter, it will be displayed



Example 1

Linearization

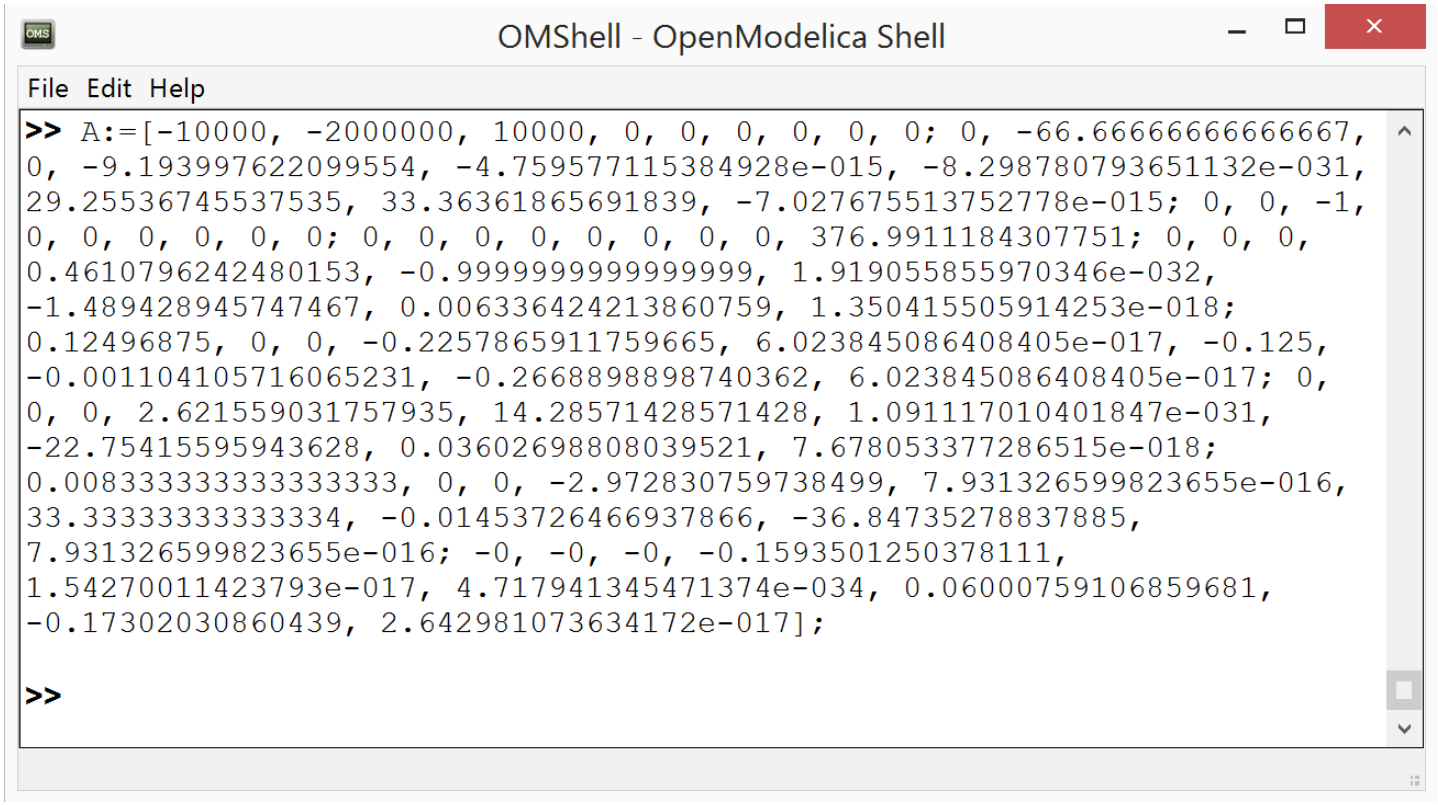
- Copy the output from the previous command without the quotation marks by pressing Ctrl+C



Example 1

Linearization

- To save the matrix A as a matrix of Real values type `A :=` and then press Ctrl+V to paste the copied matrix



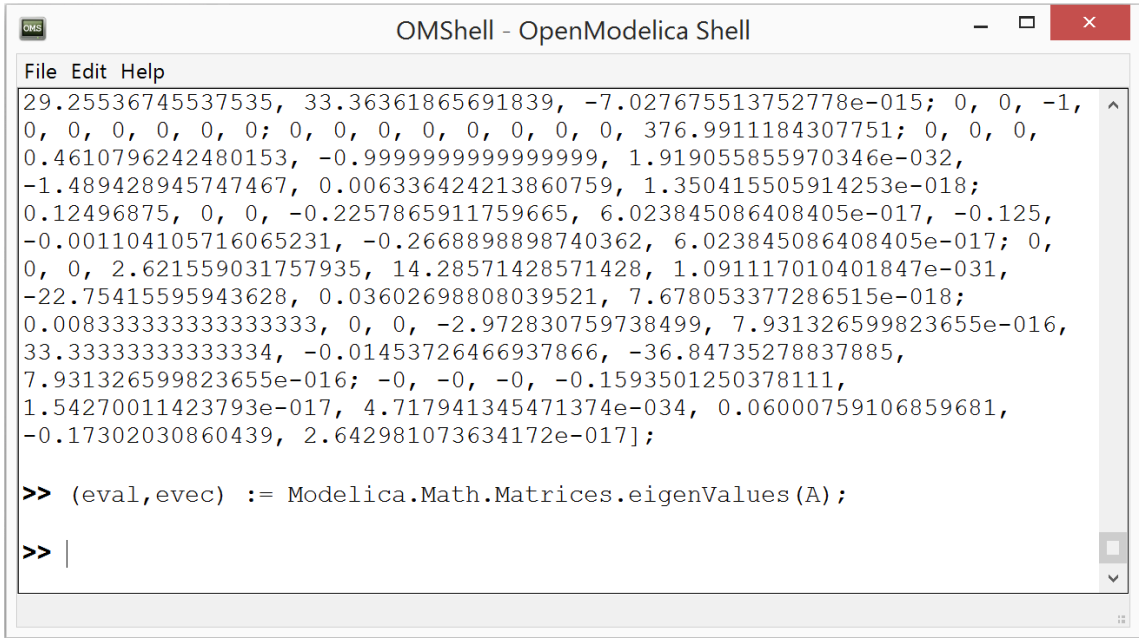
Example 1

Linearization

- It is known that the eigenvalues of the linearized system can be found by solving the following equation:

$$det(\boldsymbol{A} - \lambda \boldsymbol{I}) = \boldsymbol{0}$$

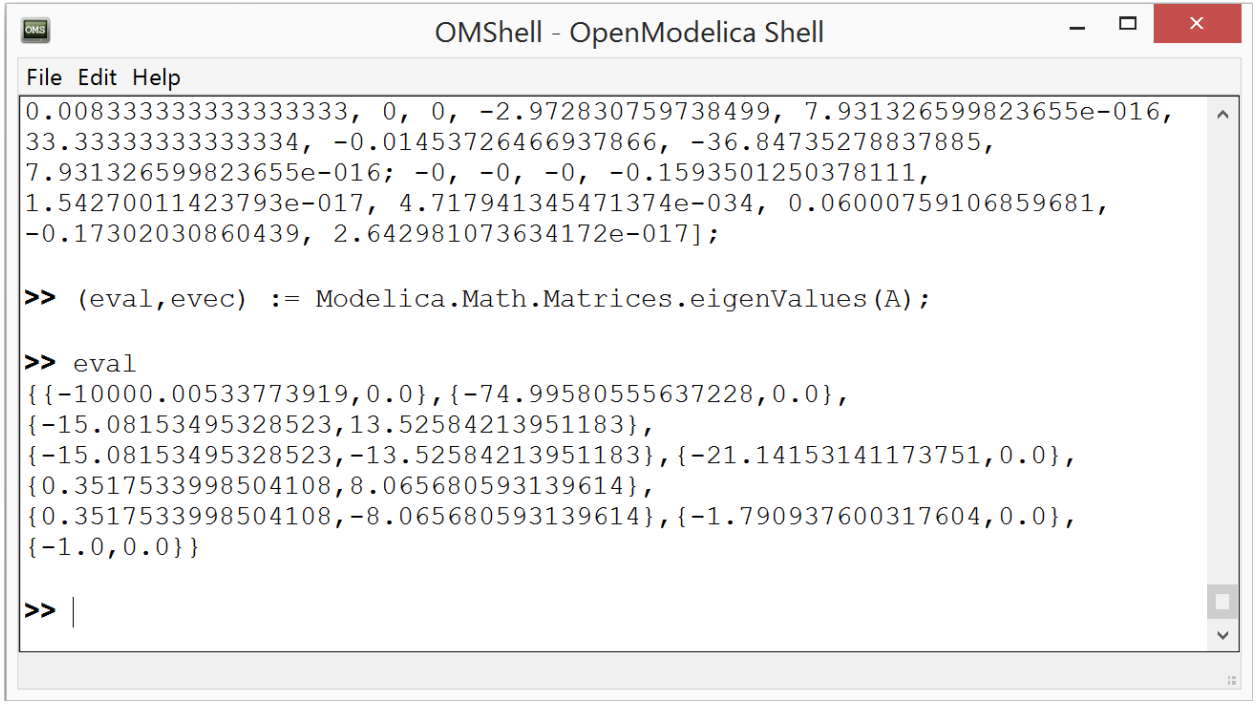
- This can be done by executing command
`(eval,evec) := Modelica.Math.Matrices.eigenValues(A);`



Example 1

Linearization

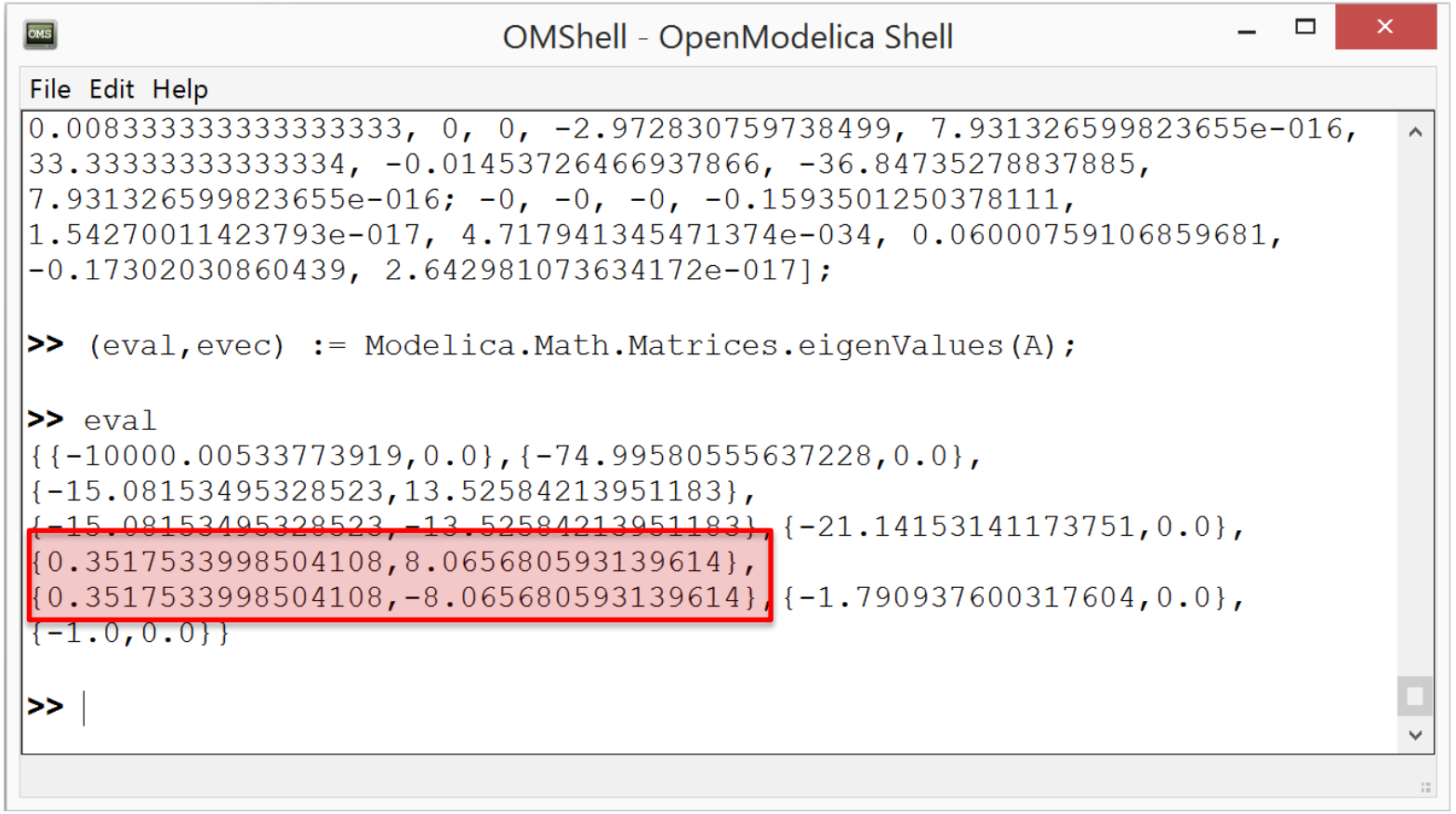
- The eigenvalues are now stored in the eval variable and they can be listed by executing `eval`
- Groups of numbers are listed where the first number is real part of the system's pole and the second one is the imaginary part



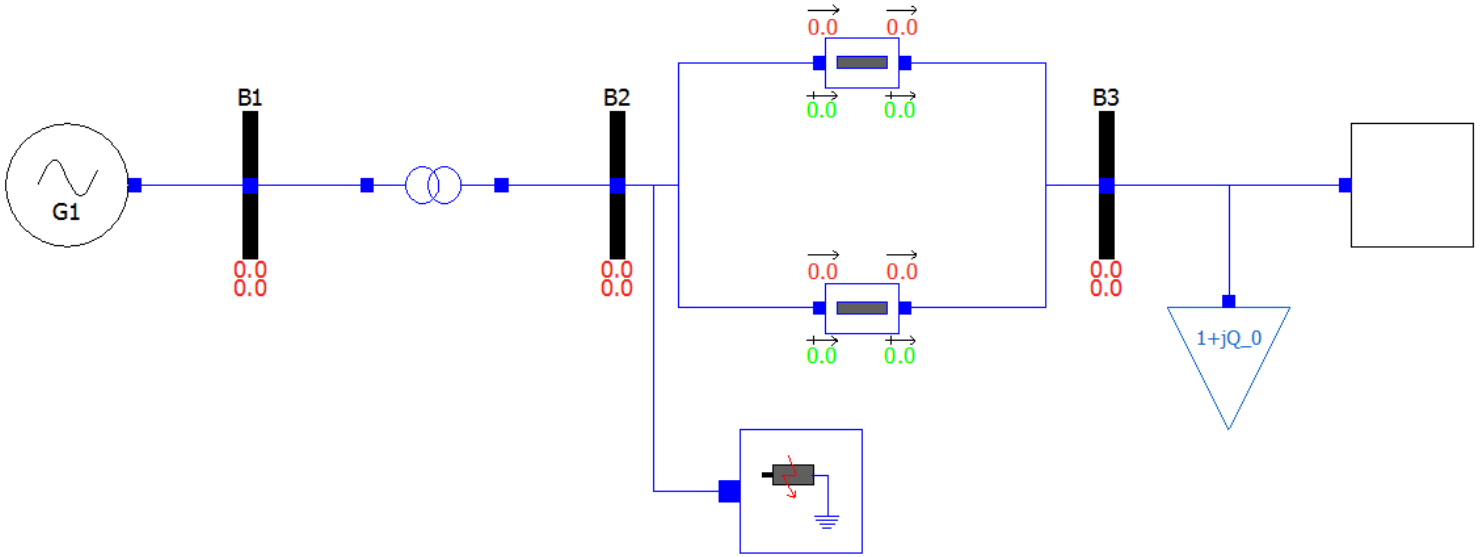
Example 1

Linearization

- It can be seen that the pair of conjugate poles exists on the right side of the stability plane and thus, the behaviour of the system is unstable



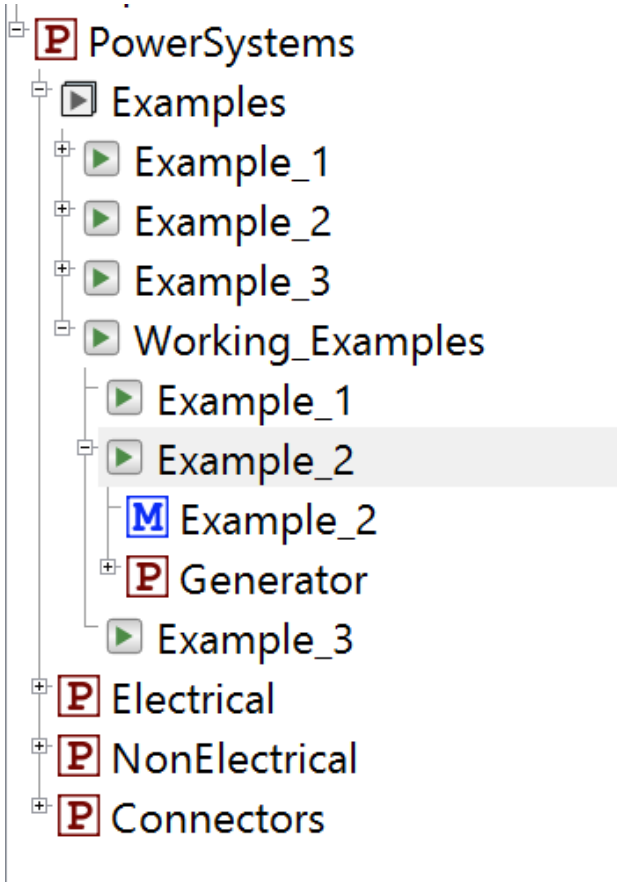
Example 2



- In the Example 1, it was shown that the system was unstable with a pair of poles on the right side of the stability plane
- In the Example 2, Power System Stabilizer (PSS) will be added to the generator in order to stabilize the system

Example 2

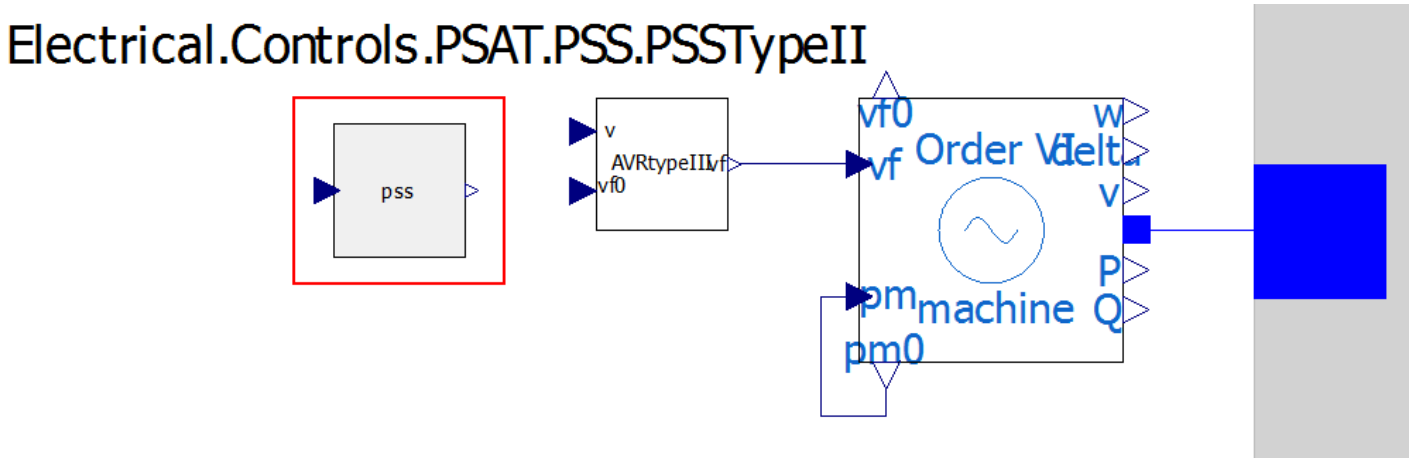
- The work on Example 2 should continue with the files prepared in a package PowerSystems.Examples.Working_Examples.Example_2



Example 2

Generator model – Step 1

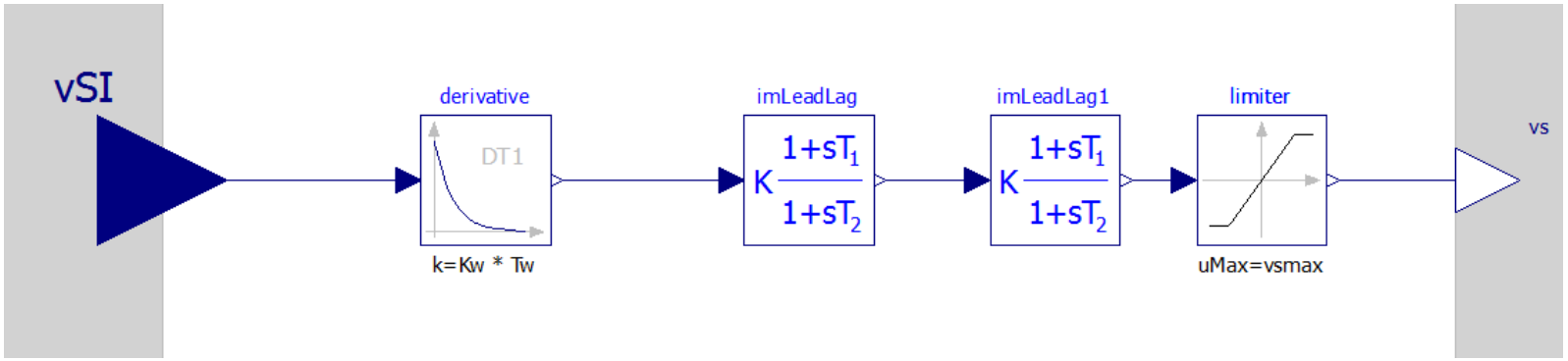
- The first step is to add the model of the PSS Type II and the summation block to the model of the generator



Example 2

Generator model – Step 1

- The internal control structure of the PSS can be accessed by right-clicking on the PSS block and selecting *"View Class"*



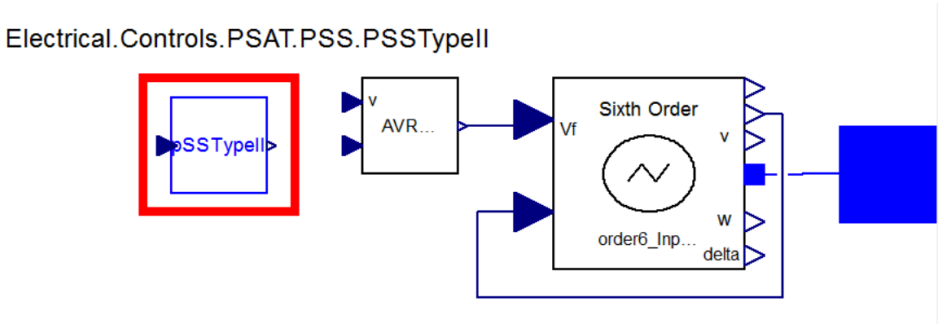
Example 2

Generator – Step 1

- PSS should be parameterized as shown in the table

PSS

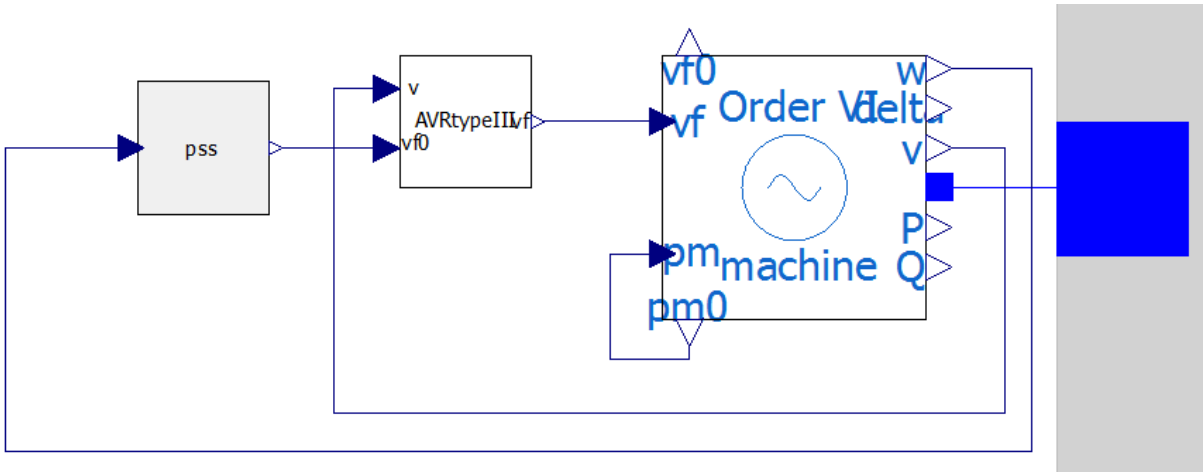
$v_{s,max}$	0.2	T_1	0.154
$v_{s,min}$	-0.2	T_2	0.033
K_w	1.41	T_3	1
T_w	0.001	T_4	1



Example 2

Generator – Step 2

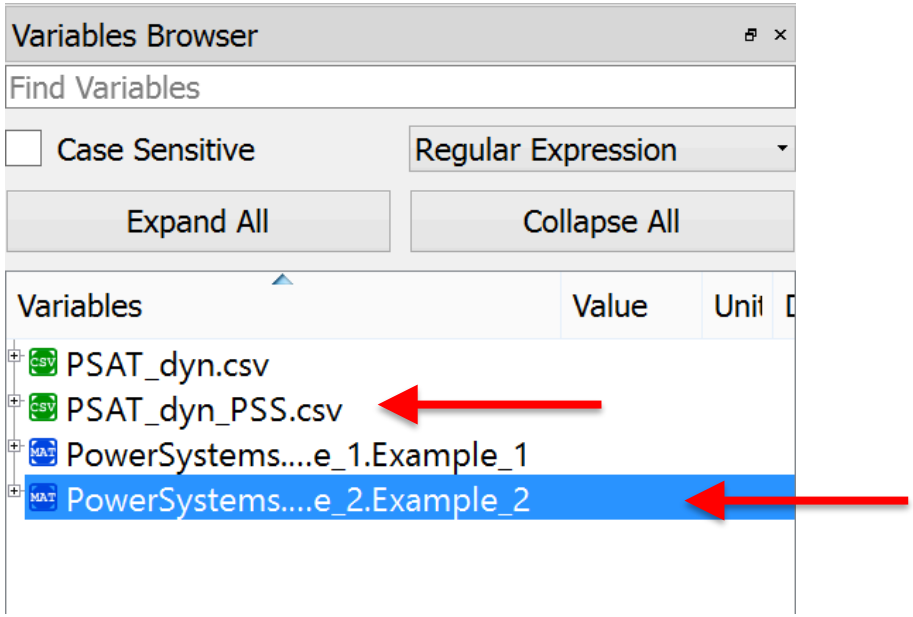
- When the signals of the generator model are connected as shown, model of the generator is completed



Example 2

Simulation

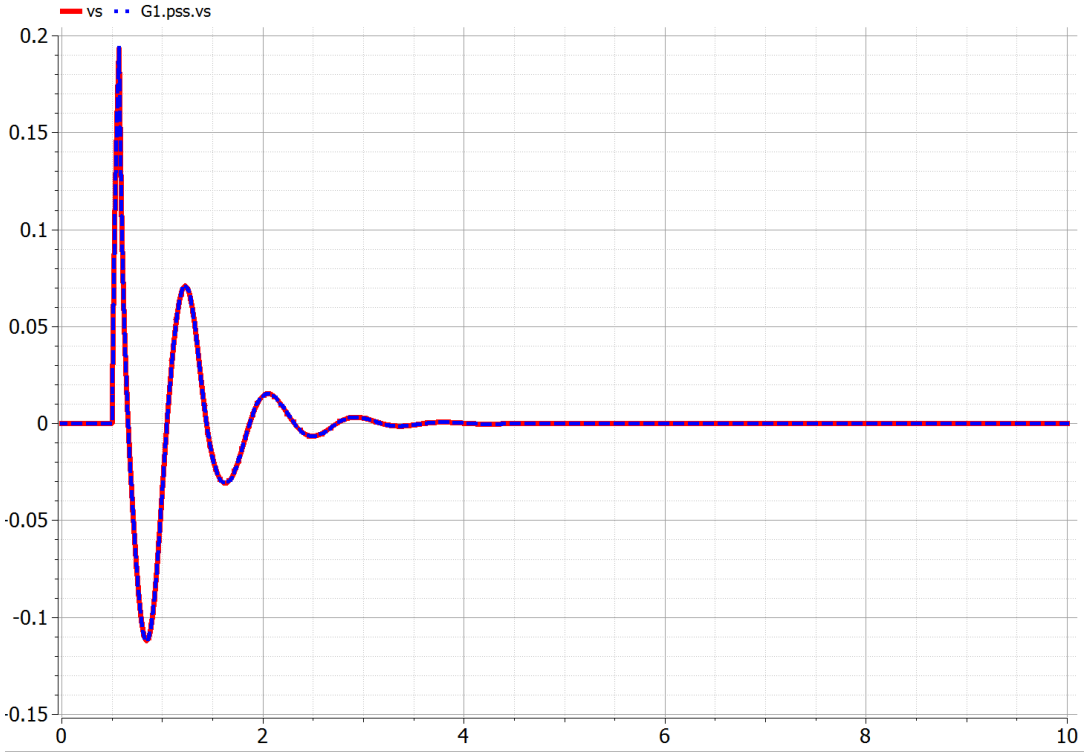
- Simulation steps can be repeated as it was shown in the Example 1
- This time, reference simulation results from the PSAT can be found in the file “PSAT_dyn_PSS.csv”
- After the simulation is executed, variable browser should look as it is shown below



Example 2

Simulation

- Simulation results can be plotted again
- Comparison of the PSAT and Modelica simulation results of the PSS signal is shown on the figure below



Example 2

Linearization

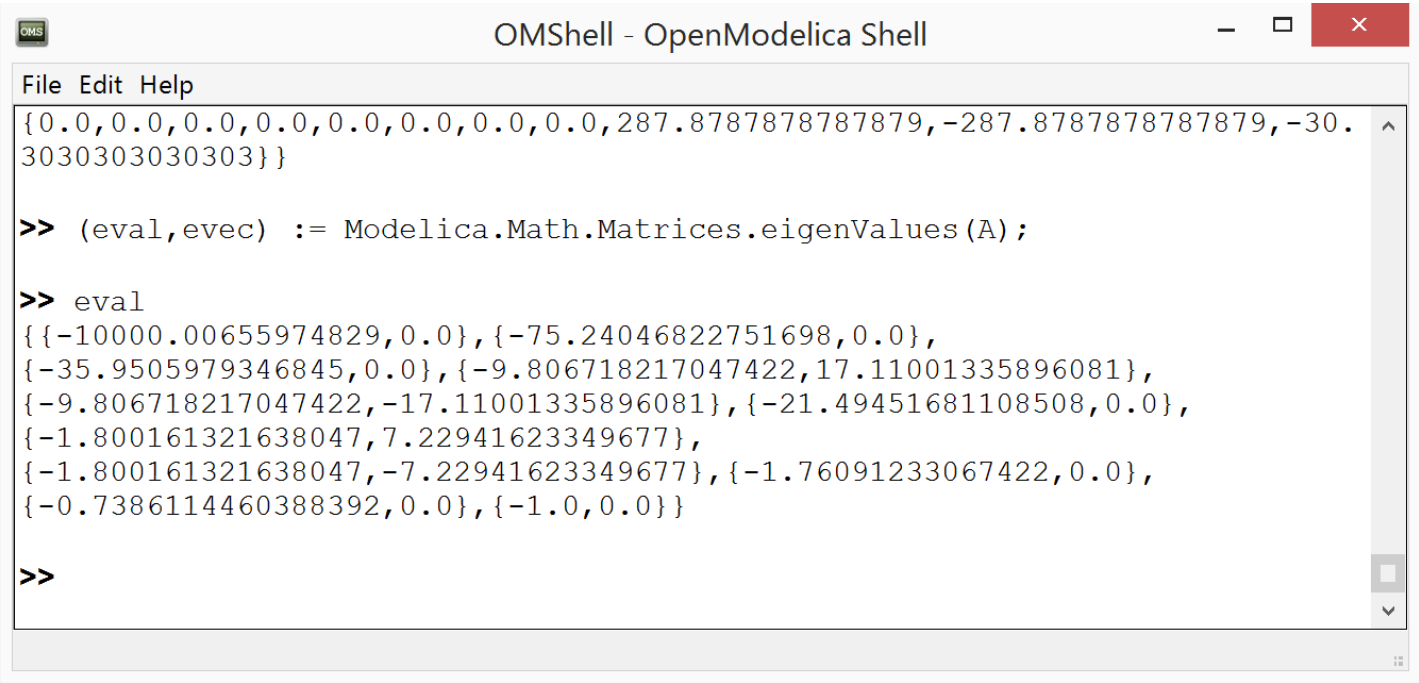
- Before the system is linearized again, changes to the `linearize_iPSL.mos` script shall be made
- Make sure to change the example number from 1 to 2 in lines 4-10
- The script should look as it is shown on the figure below

```
1   loadModel(Modelica);
2   loadFile("C:/iPSL_Workshop/iPSL_stripped.mo");
3   instantiateModel(iPSL;
4   instantiateModel(iPSL.Examples.Example_2.Example_2);
5
6   linearize(iPSL.Examples.Example_2.Example_2,stopTime=0.0);
7
8   loadFile("linear_iPSL.Examples.Example_2.Example_2.mo");
9
10  (a) := getParameterValue(linear_iPSL_Examples_Example__2_Example__2,"A");
```

Example 2

Linearization

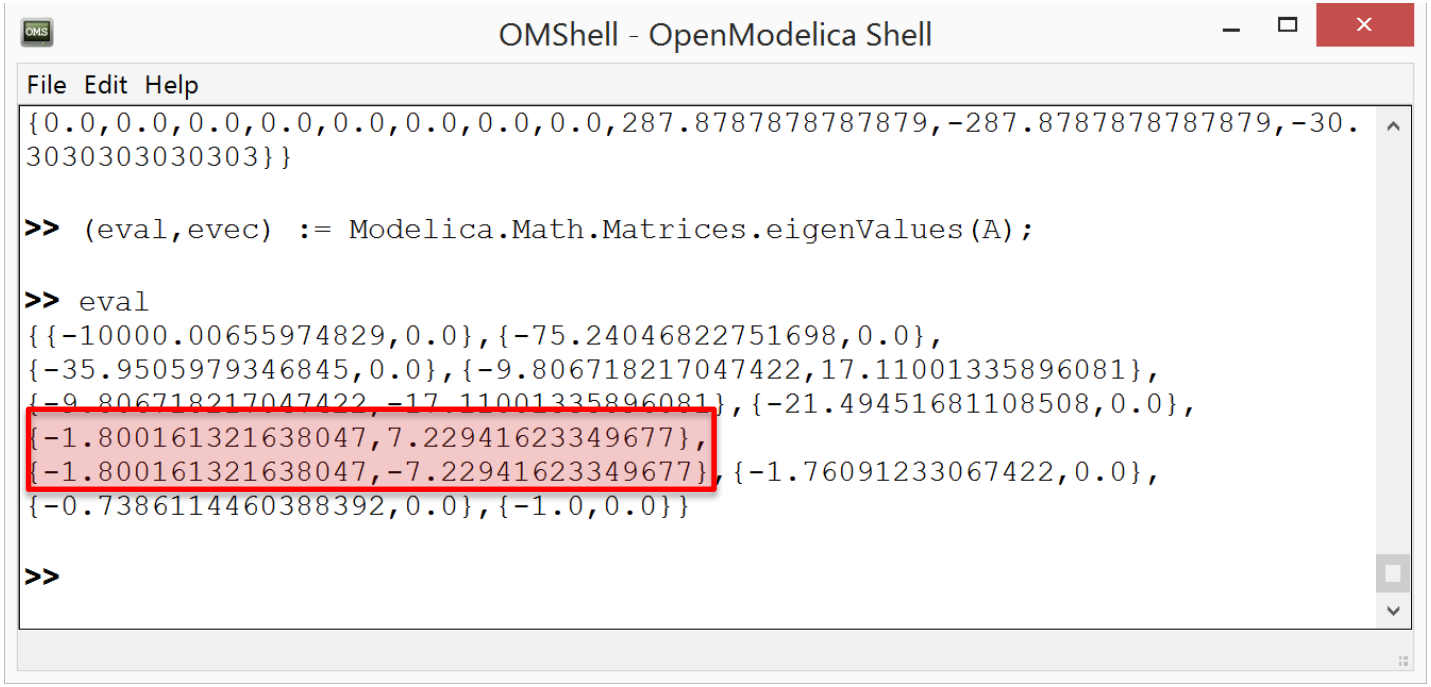
- The rest of the steps shall be repeated as it was shown in Example 1
- The same procedure with a linearized system from Example 2 results in the new set of eigenvalues



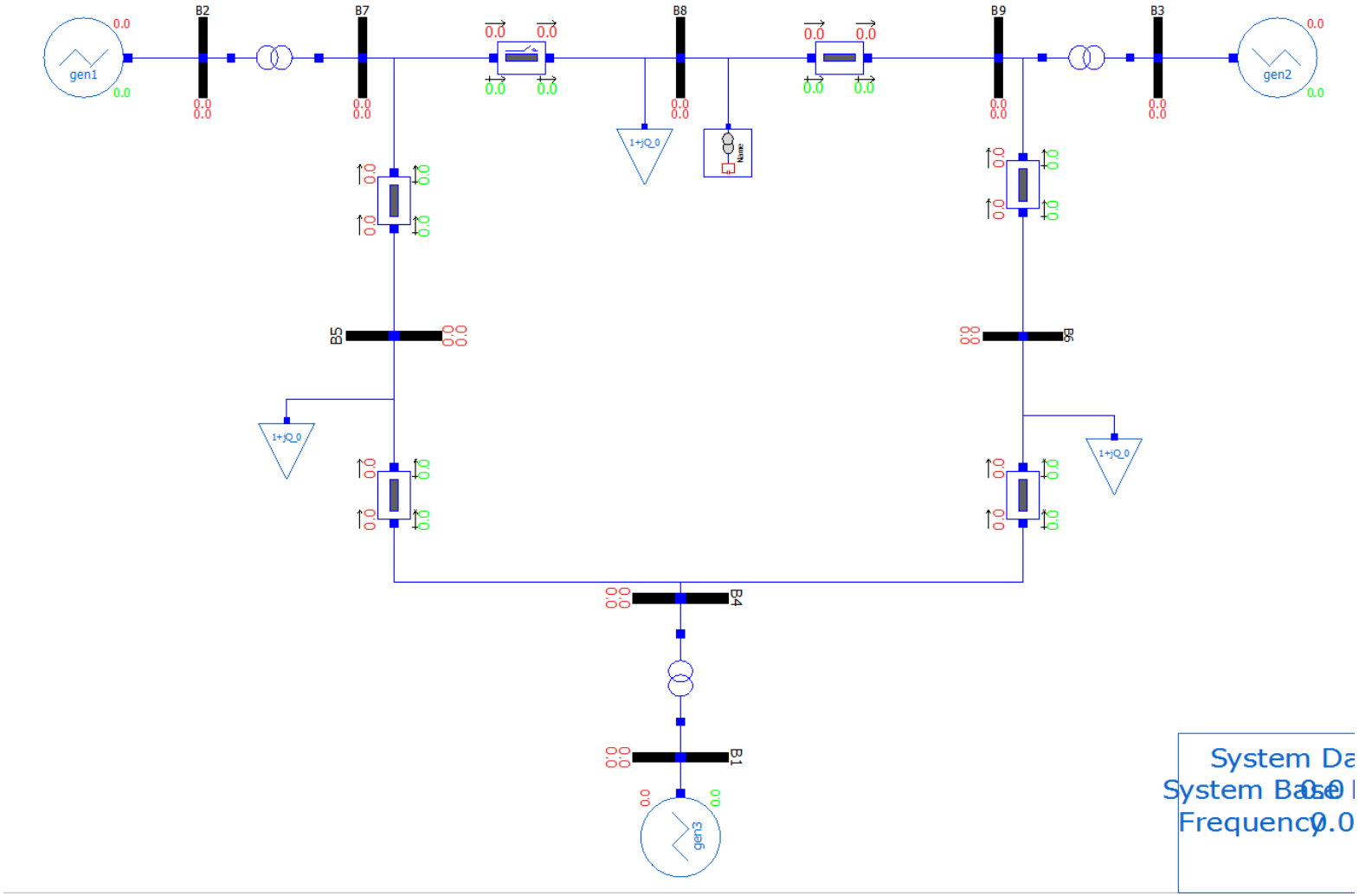
Example 2

Linearization

- The conjugate pair of poles that was on the right side of the plane in Example 1 was, by introducing the PSS, moved to the left side of the stability plane and, thus, the system is now stable



Example 3

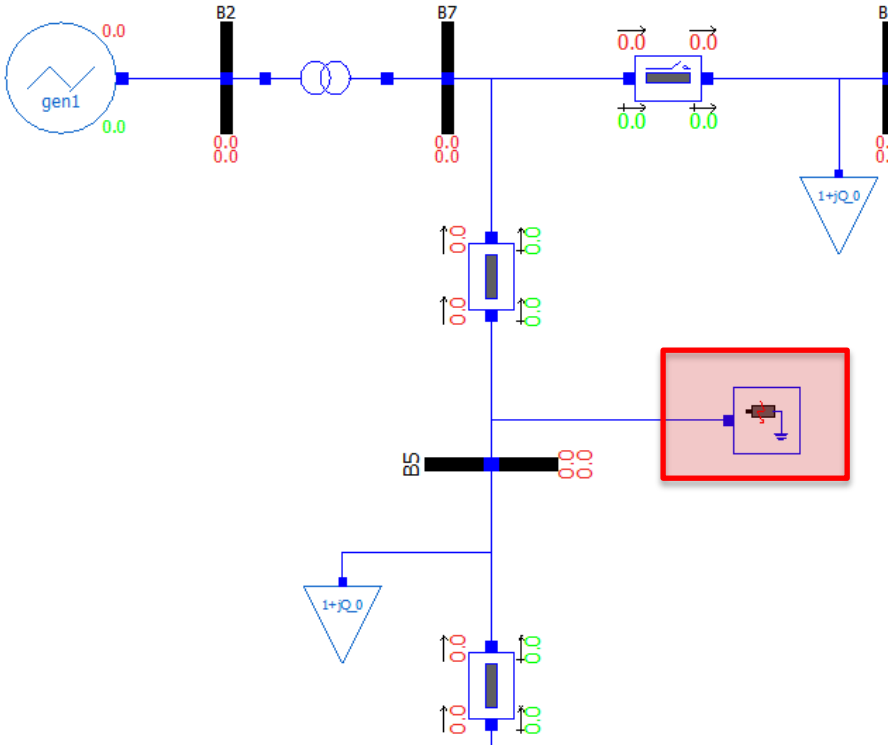

B2
B7
B8
B9
B3
gen1
gen2
gen3
B5
B6
B4
B1
1+jQ_0
1+jQ_0
1+jQ_0
System Da
System Base
Frequenc

Example 3

- Example 3 containes the model of the IEEE 9 Bus system with the STATCOM at bus 8
- It is pre-configured with all of the power flow and dynamic data
- In the previous two examples, you learned how to build the models of the power system, introduce the faults, run the dynamic simulations and perform the linearization of the model
- In Example 3 you are free to explore the model and introduce various faults

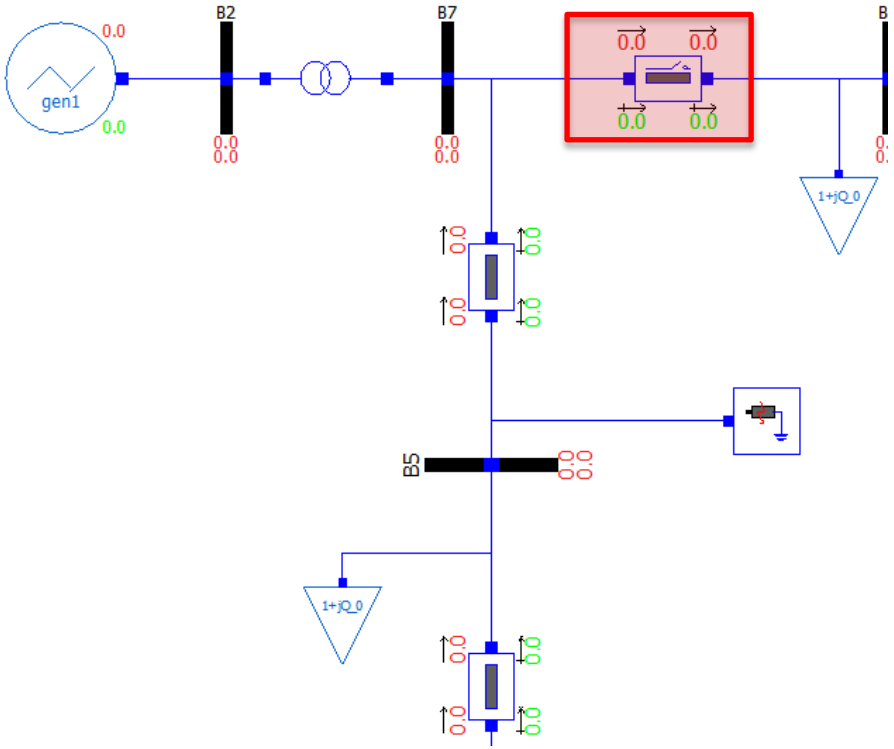
Example 3

- You can, for instance, introduce the bus fault …



Example 3

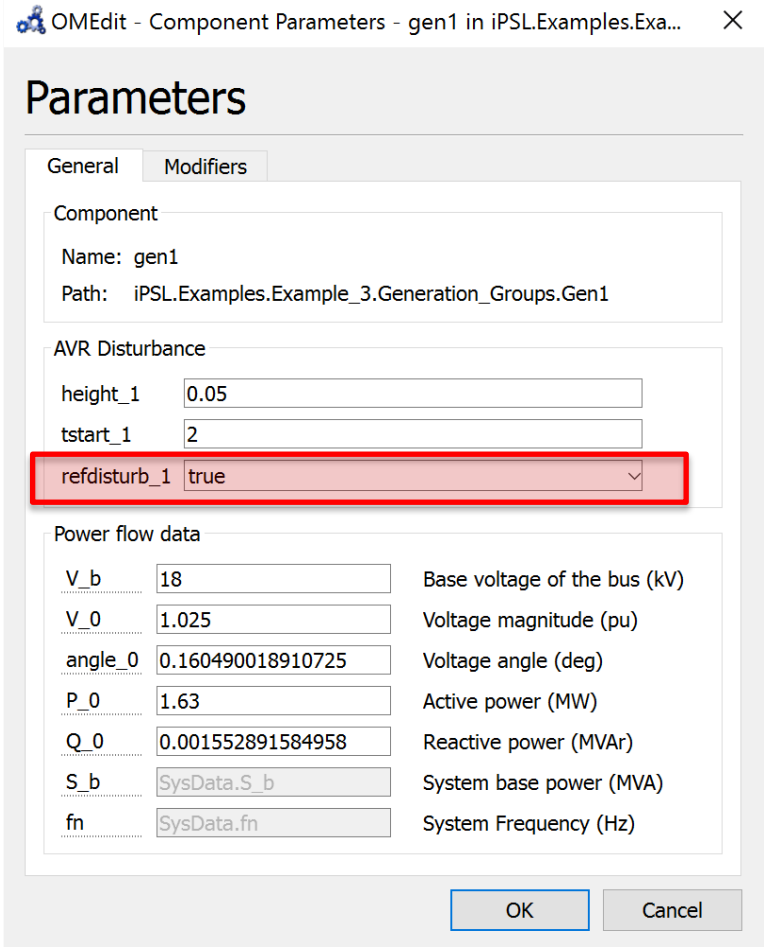
- ... or open the line at the given time instant*



*Model of the line with opening is PowerSystems.Electrical.Branches.PwLine2Openings

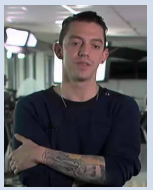
Example 3

- Step disturbance to the voltage reference of the generators can be introduced by setting the desired `refdisturb_x parameter to true`



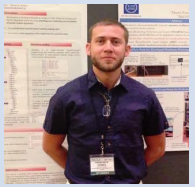
iPSL Developers

KTH Developers

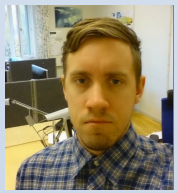

Luigi Vanfretti


Achour Amazouz


Mohammed Ahsan Adib Murad


Francisco José Gómez


Giusseppe Laera


Jan Lavenius


Le Qi


Maxime Baudette


Mengjia Zhang


Tetiana Bogodorova


Tin Rabuzin


Joan Russiñol Mussons