OpenModelica for Power Systems at TU Delft

Digvijay Gusain
Dr. Peter Palensky

Intelligent Electrical Power grids
Electrical Sustainable Energy
EEMCS, TU Delft

d.gusain@tudelft.nl; p.palensky@tudelft.nl

TU Delft

About me

- 2014 - BTech (EEE) from Delhi Technological University, India
- 2016 - MSc (ESE) from TU Delft, Netherlands
- 2016 - Interned at EPRI, USA
- 2017 - PhD student at TU Delft, Netherlands

Overview

- IEPG push for open source tools, Modelica and Python.
- OpenModelica for dynamic electric power systems simulations.
- Python (PyPSA, pandapower) for steady state power systems.
- Multi-energy systems.

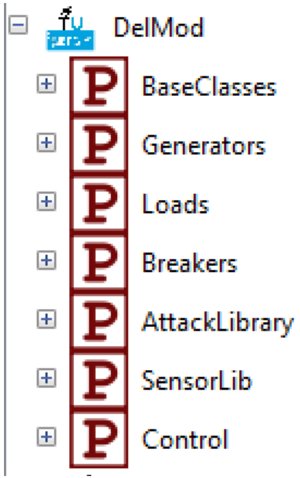
HAPSISH
Heat and Power Systems at
Industrial Sites and Harbours

TU Delft

OpenModelica for power systems @ TUD

- Course on Intelligent Electrical Power Grids
  - Modeling and simulation of power systems.
  - Intelligent controls: local and central.
  - Cyber attacks on power systems.

- Developed a library: DelMod
  - Heavily based on OpenIPSL library.
  - Attacks library, Load models, Sensors etc.



- Python based tool for auto-initialisation of OpenIPSL models

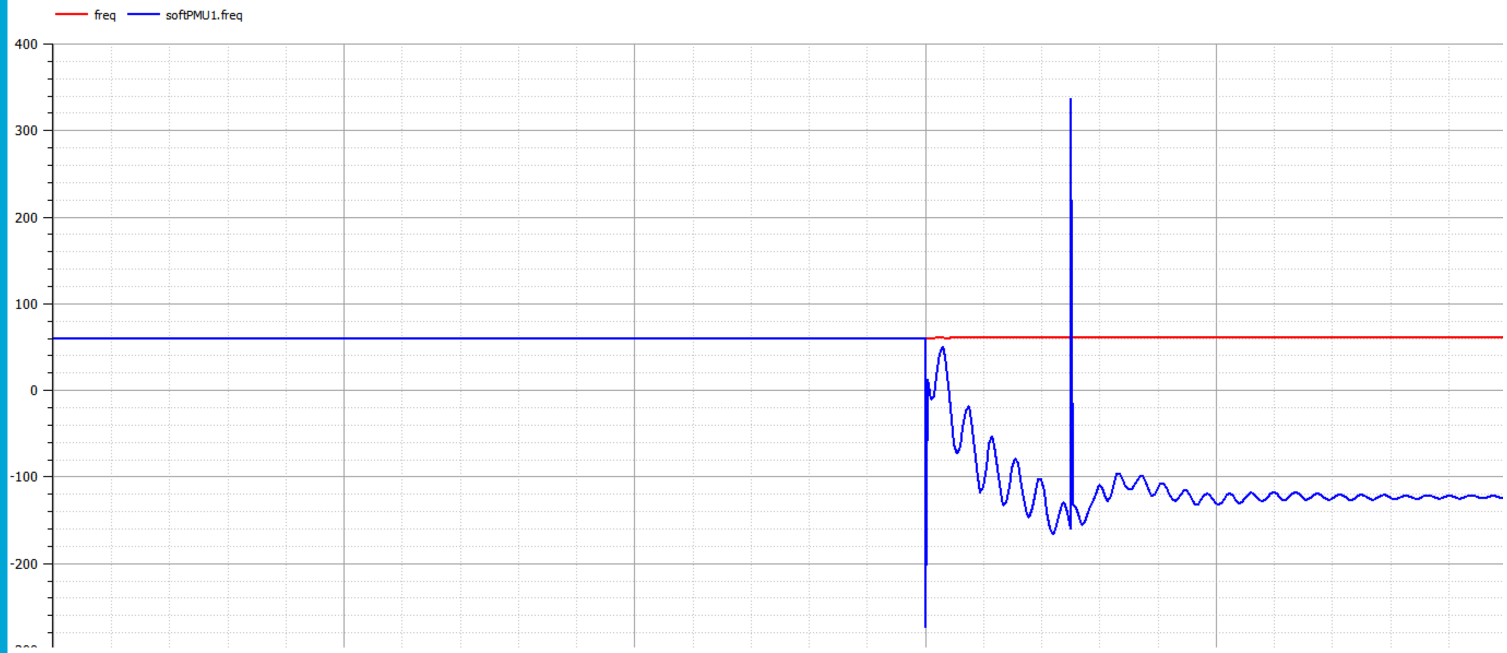
TUDelft

Takeaways

+ Generally positive.
+ Students appreciate availability of high quality open-source tools.
+ IEPG group push for more open source involvement.
+ Contributions made to OpenIPSL library.
- Many libraries don’t (extensively) support OpenModelica.
- Electrical domain library components still not mature.

TU Delft

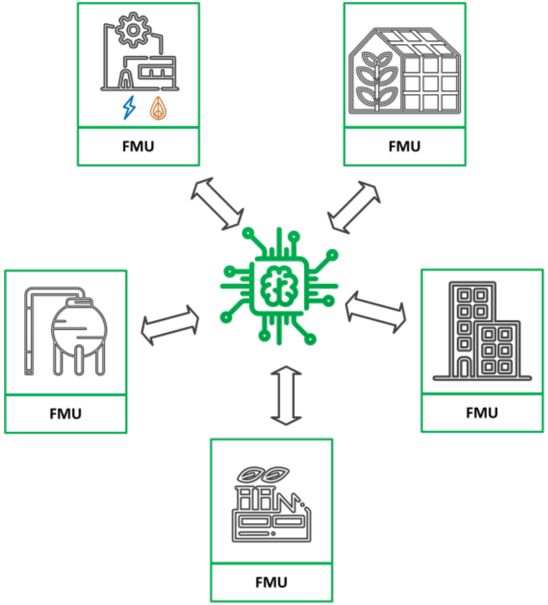
Example: OpenIPSL PMU



PMU: Phasor Measurement Unit

Multi-Energy Systems

- Analysing energy flexibility in industrial parks using multi-energy systems.
- Modeling components and networks:
  - Boilers
  - CHPs
  - Heat Pumps, heat networks
  - Electric distribution grid, loads, controls
- Libraries such as AixLib, Buildings, OpenIPSL, IPBSA, etc..
- FMU based co-simulation approach



Takeaways

+ Mature open-source libraries and examples to get started.
+ Great documentation and GitHub support.
+ Easy export of FMUs.
- Libraries such as Buildings can take lot of time to load.
- Model compatibility is not 100% with OpenModelica.

TU Delft

IEPG Tools

- To encourage FMU based (co-)simulation:
  - fmuAdapter
    - init(), step(), set_input(), get_output().
  - FMUWorld
    - Simple co-simulation.
    - World(), add_fmu(), add_connections(), simulate().
    - sensitivity analysis and parameter optimizations.

TU Delft

FMUWorld

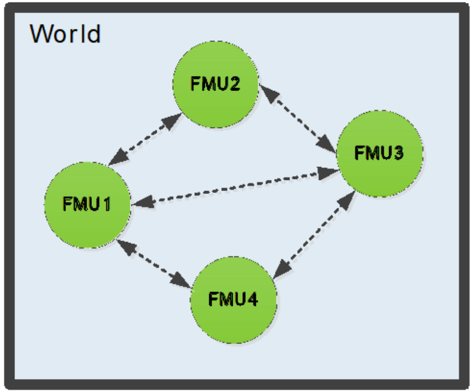
```
#import World object from FMUWorld
from FMUWorld import World

#create an instance of World as my_world
my_world = World(start_time = 0.0,
          stop_time = 100.0,
          logging = True,
          exchange = 1)
```

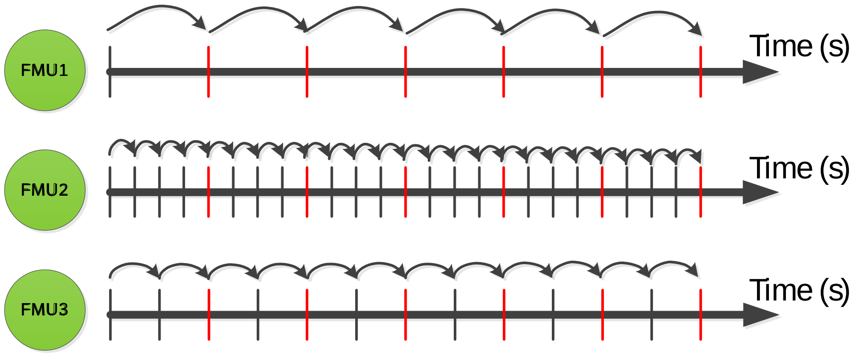
```
#add FMU to my_world
my_world.add_fmu(fmu_name = 'fmu1',
        fmu_loc = fmuLoc2,
        step_size = 1e-3,
        inputs = ['input_connector'],
        outputs = ['component.variable'])
```

```
#constant signal of value x1
my_world.add_signal('signal1': [x1])
#step input. if time > t1 value = x2 else x1
my_world.add_signal('signal2': [x1, t1, x2])
#pulse input. if t1<time<t2 value = x2 else x1
my_world.add_signal('signal3': [x1, t1, t2, x2])
```

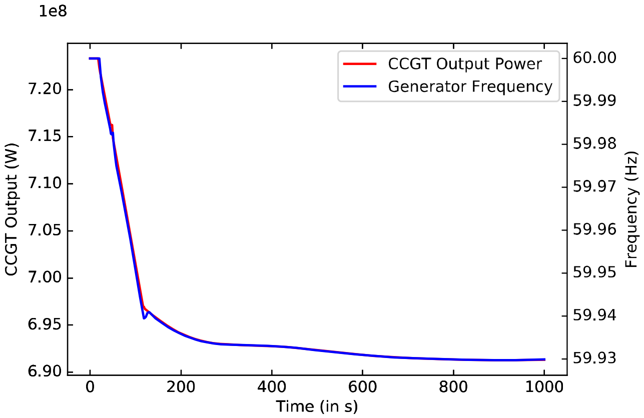
```
#define connections
connections = {'fmu1.output':'fmu2.input',
    'signal1':'fmu3.input',
    ('fmu2.output', 'fmu3.output'):'fmu4.input'}

#add connections to the world
my_world.add_connections(connections)
```

```
#simulate my_world
results = my_world.simulate()
```



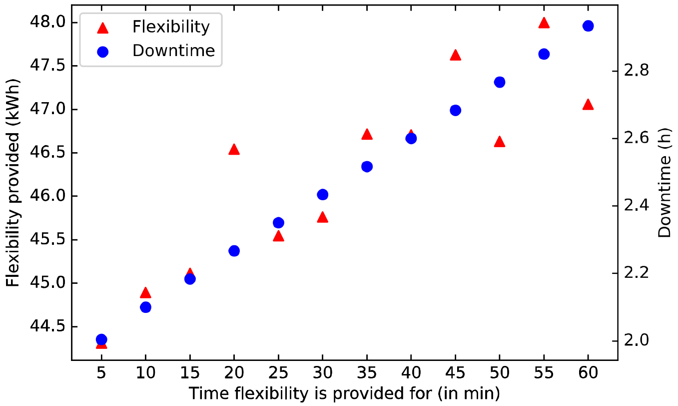


Results



Multi-Energy Co-simulation

Sensitivity analysis for energy flexibility



Future Plans

- Creating MOOC on EdX for IEPG using OpenModelica (4TU).
- Incorporating OpenModelica in courses and MSc thesis.
- Contributing models developed to OpenIPSL library.
- Multi-energy systems co-simulation.
- GUI for FMUWorld.

TU Delft

Thank you!

Digvijay Gusain
d.gusain@tudelft.nl