POWERFACTORY

Standardization

in Power Systems Dynamics

POWER SYSTEM SOLUTIONS
MADE IN GERMANY

Why do we need standardization?

- Free tool choice due to interoperability
- The best tool for it’s purpose can be used
- Silos and castle-building can be reduced
- Change of tools is getting easier

Can we have standard formats such as Words docx?

- Todays industry standards are mainly coming from de-facto industry standards (MS-Office, Photoshop, etc.) but allows compatibility
- Other standards are dictated by big vendors (e.g. no flash support on Apple products) and therefore drive the industry to new technology

But what can we expect in the power system world?

Where we are now

Network Models

- Every Tool has their own file format
- Most commonly used are:
  - PSS/E
  - CIM
- Standardization always often is limited to certain use cases
- Double conversions might apply

Dynamic Models

- Various tools come with different modelling approaches
- Often models are not accessible from other tools
- Interoperability is practically not possible

Where we are going

Network Models

- PSS/E or CIM as standard models?
- Both formats are good for their purpose
- This purpose is limited and not always sufficient
- CIM is constantly developing, but is also over-engineered and not resource friendly

Dynamic Models

- Multi-Purpose language models, such as Modelica gain popularity in the industry
- Function Mock-Up Interface/Unit (FMI/FMU) can be used for black-box models

Feasible solutions do exist but need to be used!

What is MODELICA?

Modelica is an open modeling language developed and owned by the Modelica Association

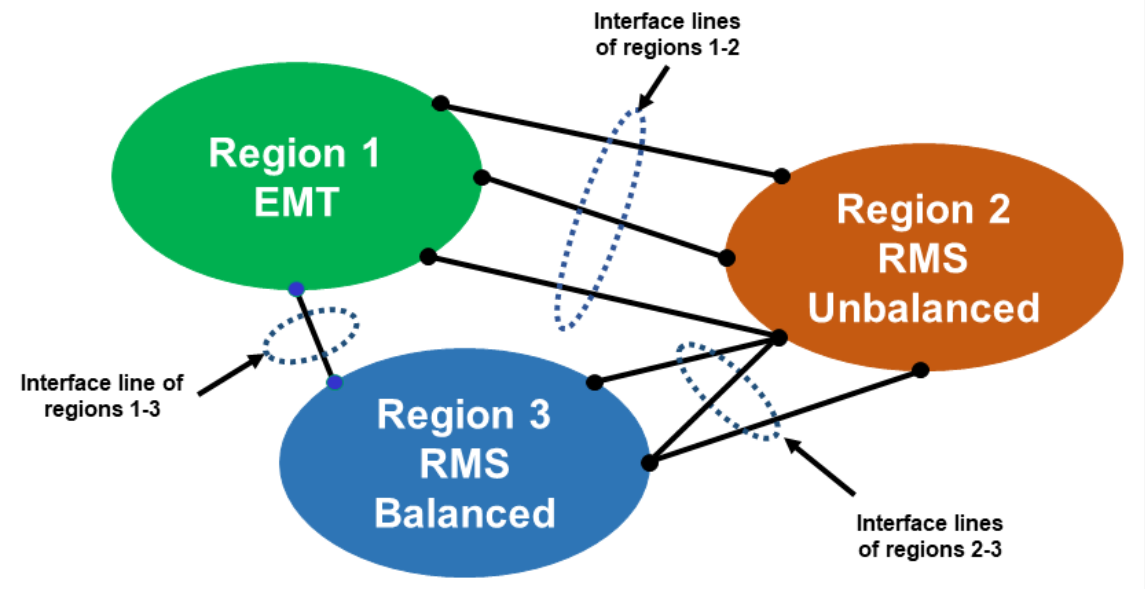
- The Modelica Association is a non-profit, non-governmental organization
- Actively develops and promotes the Modelica modeling language
- **Applications:** mechanical, electrical, electronic, magnetic, hydraulic, thermal, control, electric power or process-oriented subcomponents

The Functional Mock-Up Interface (FMI)

- The Functional Mock-Up Interface (FMI) is a standard Interface by the Modelica Association
- Enables Co-Simulations between different tools
- A variety of tools do support the FMI, including:
  - PowerFactory
  - MATLAB Simulink
  - EMTP-RV
  - dSpace
  - And many more: https://fmi-standard.org/tools/

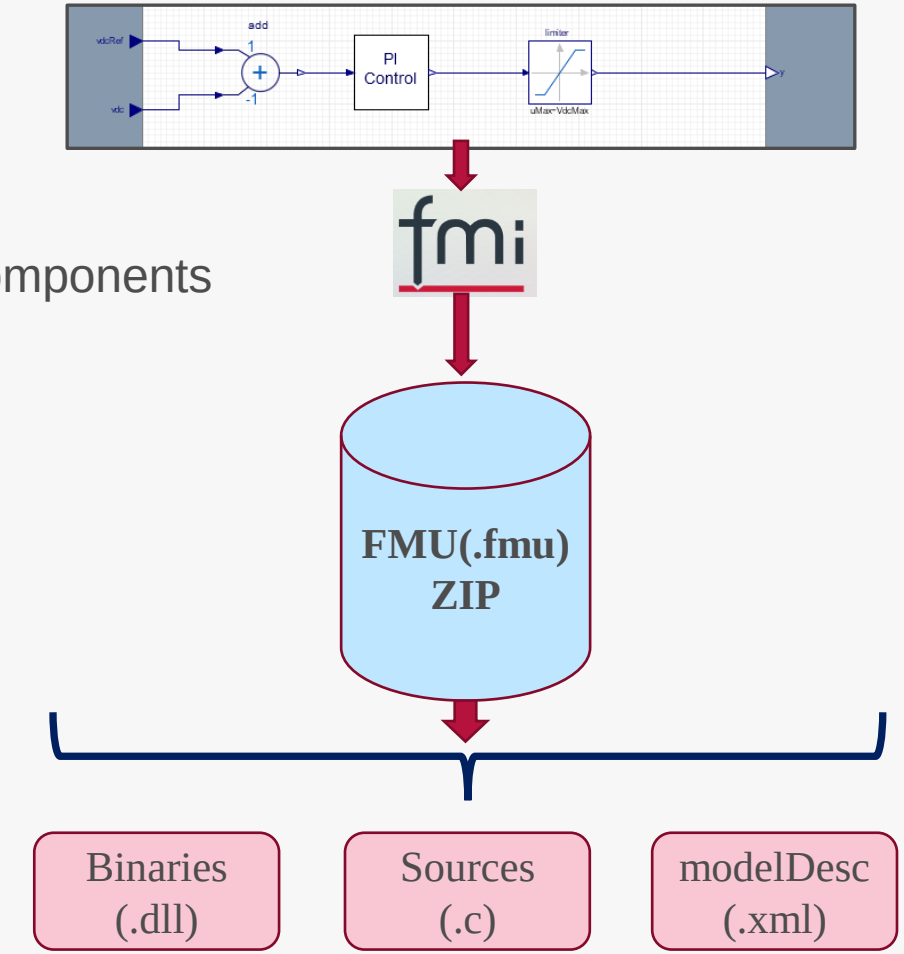
fmi Functional Mock-up Interface

The leading standard to exchange dynamic simulation models

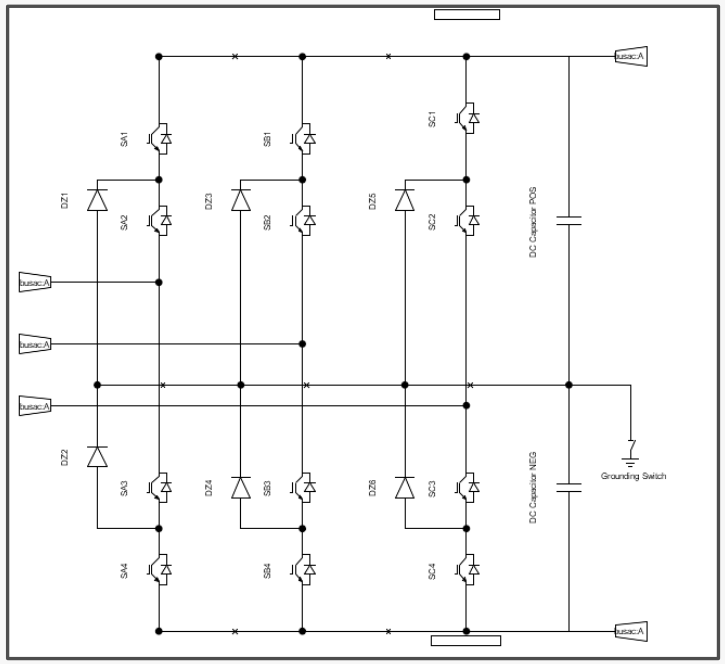


Model Exchange with the Functional Mock-Up Unit (FMU)

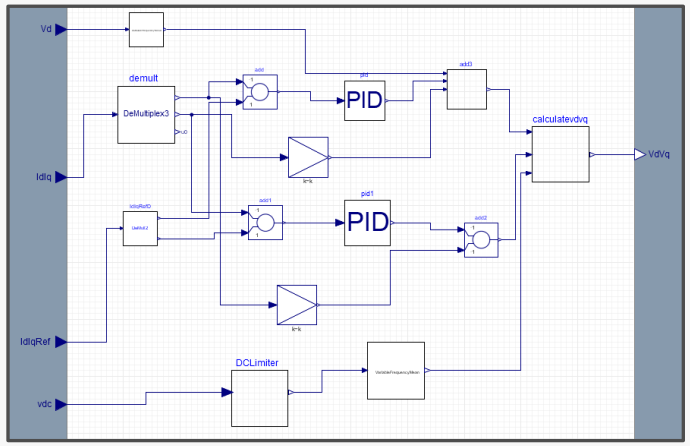
- Model exchange based on the FMI via the Functional Mock-Up Unit (FMU)
- Black-box models that can be used between different tools
- **Advantages:**
  - Convenient way of producing, sharing and using simulation components
  - Efficiently couple multi-disciplinary simulations
  - One FMU: exchange between different tools



Key Benefits of Modelica Models and FMU



Need for Discrete Controllers



Diagram

```
if IntMethod == 1 then
x := previous(y)+k*Ts*0.5*previous(u);
y := if y > uMax then uMax else if y < uMin then uMin else x+k*Ts*0.5*u;
elseif IntMethod == 2 then
y := if y > uMax then uMax else if y < uMin then uMin else previous(y)+k*Ts*previous(u);
end if;
```

Algorithms

Sequential Code Generation

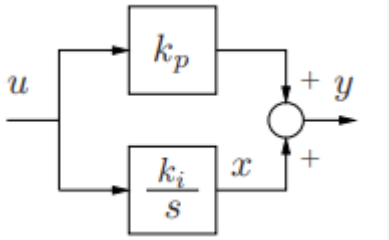
Simulate in PF

Functional Mock-up Interface

Portable model to external tools, e.g., MATLAB

Modelica Examples

A Simple Modelica Model



Hybrid Model

```
3  model Example1StandAlone
4  parameter Real kp (min=0, max=10) = 0.1;
5  parameter Real ki (min=0,max=100) = 1.5;
6  parameter Real uref = 1;
7  Real u; //input to the controller
8  Real x; //Integrator state variable
9  output Real y;//output
10 initial equation
11 der(x) = 0;
12 equation
13 u = if time < 0.1 then 1 - uref else 1.01 - uref;
14 y - kp*u - x = 0;
15 der(x) - ki*u = 0;
16 end Example1StandAlone;
```

Additional equations

```
inc(y)=0
inc(x)=0
u = select(time() < 0.1,1 - uref,1.01 - uref)
y = kp*u + x
x.= ki*u
```

Modelica

DSL

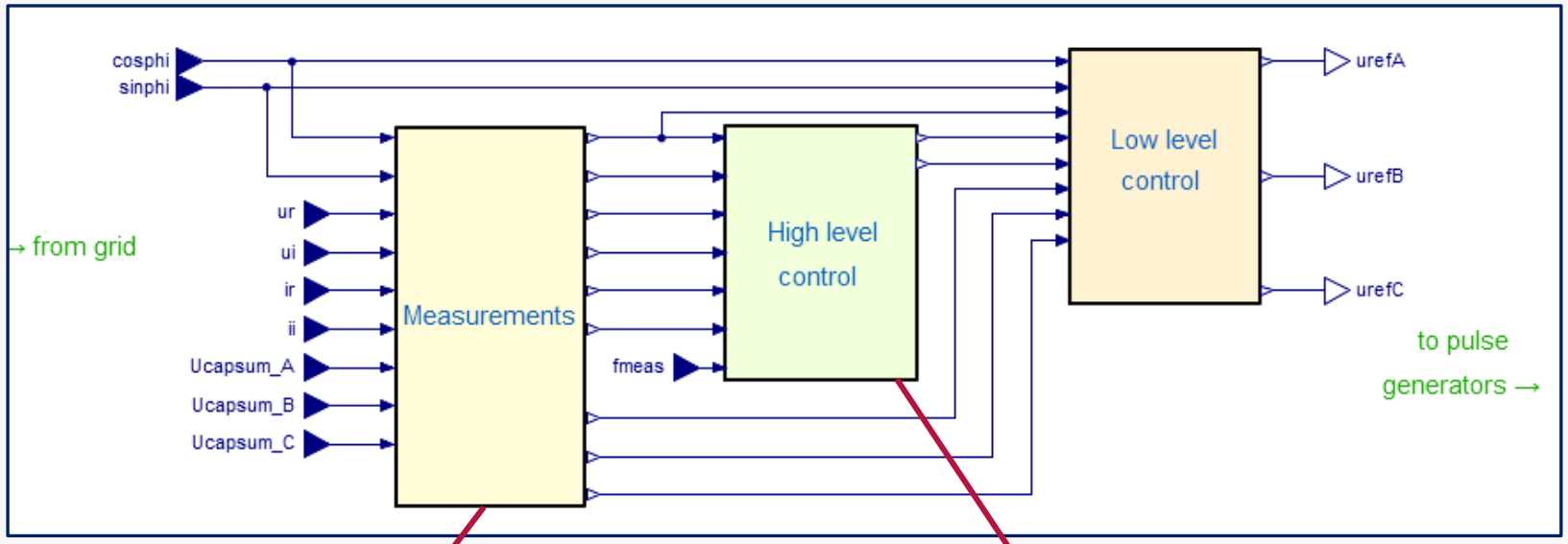
Clocked Model

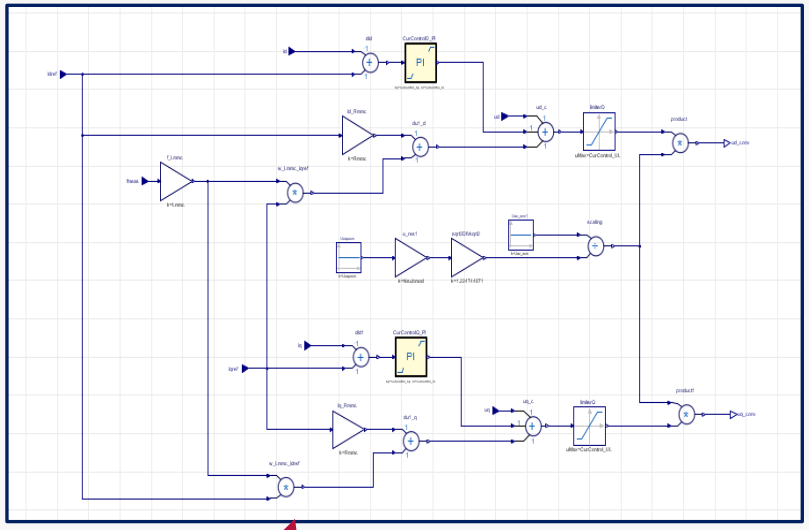
```
18 model Example1StandAloneAlg
19 parameter Real kp (min=0, max=10) = 0.1;
20 parameter Real ki (min=0,max=100) = 1.5;
21 parameter Real uref = 1;
22 Real u; //input to the controller
23 Real x; //Integrator state variable
24 output Real y;//output
25 Real y_clocked;
26 parameter Real Ts =0.001;
27 Clock c = Clock(Ts);
28 equation
29 y=hold(y_clocked);
30
31 algorithm
32 if (firstTick(c)) then
33 u := 0;
34 x := 0;
35 y_clocked := 0;
36 else
37 y_clocked:=previous(y_clocked);
38 x:=previous(x);
39 u:=previous(u);
40 u := if time < 0.1 then 1 - uref else 1.01 - uref;
41 x := previous(x)+ki*interval()*previous(u);
42 y_clocked := kp*u+x;
43 end if;
44
45 end Example1StandAloneAlg;
```

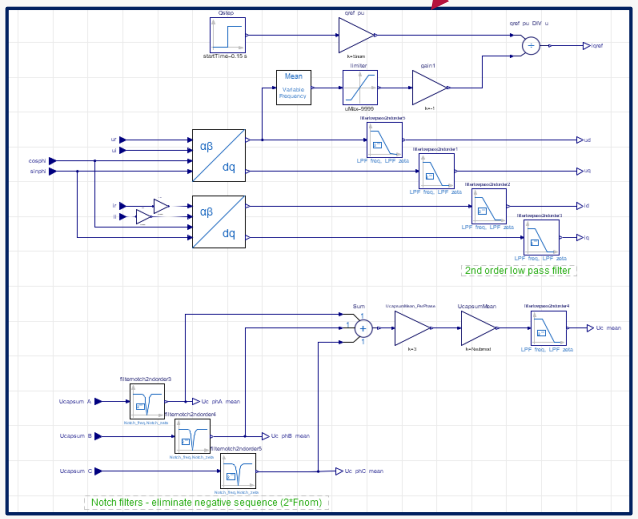
Algorithm after first tick

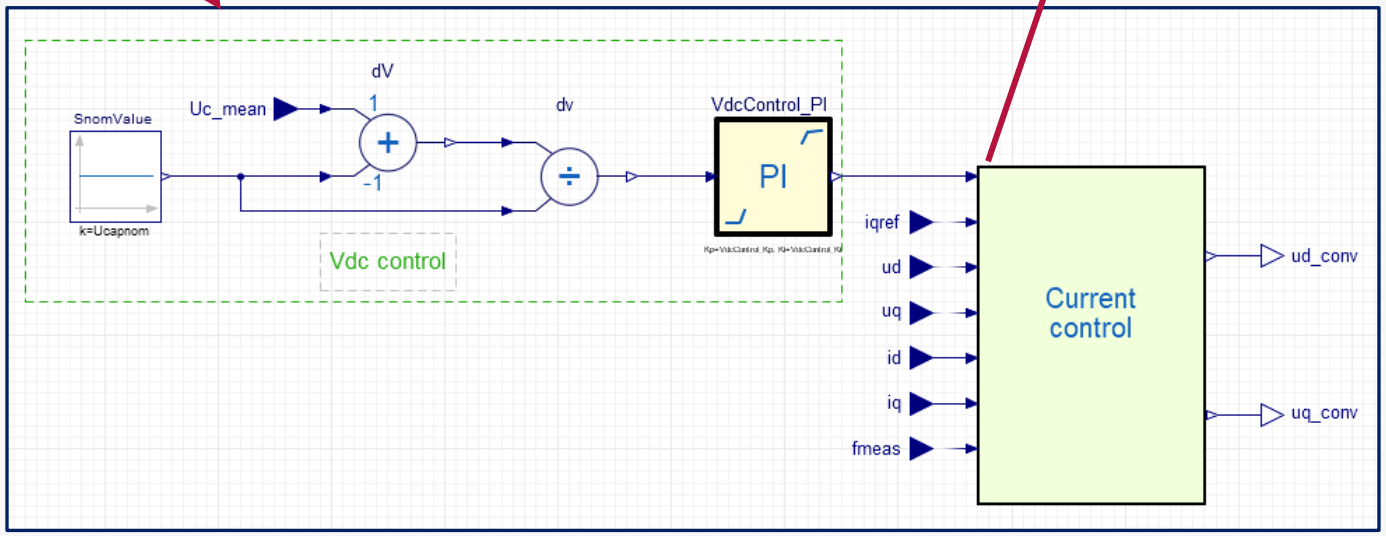
```
u := if time < 0.1 then 1 - uref else 1.01 - uref;
x := previous(x)+ki*interval()*previous(u);
y := kp*u+x;
```

Graphically Structured Model





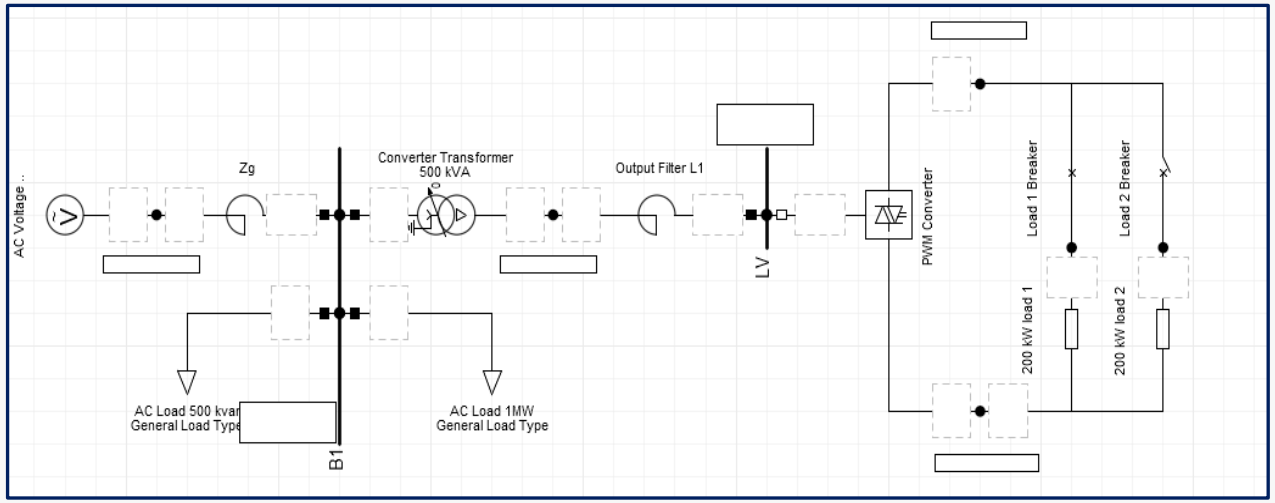


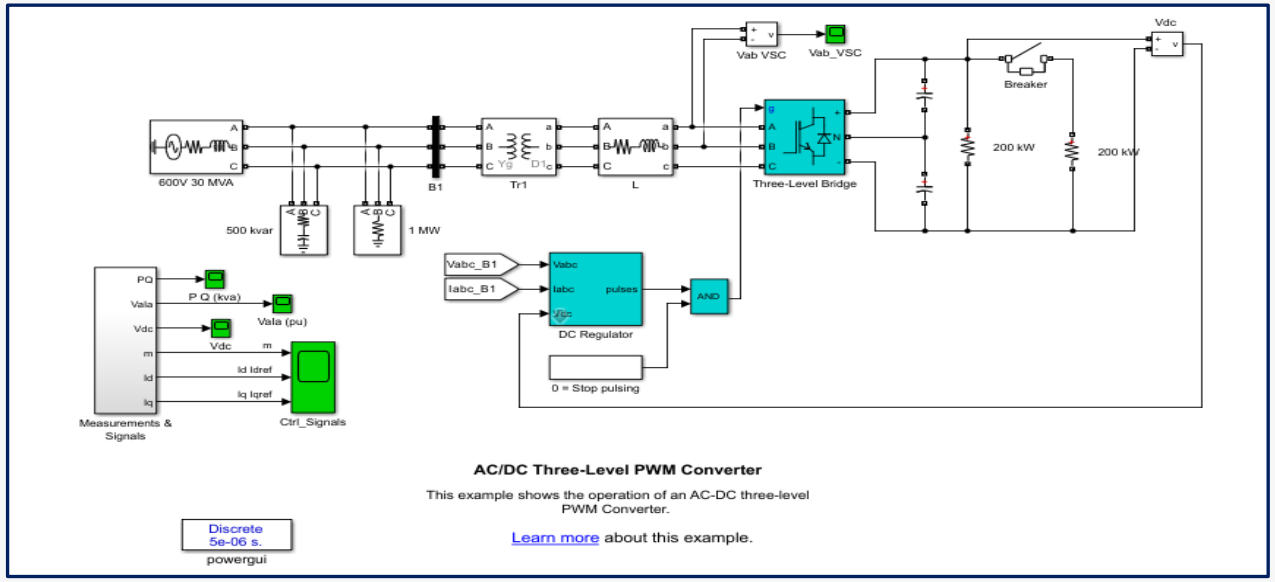


Example for Interoperability

FMU Exchange between PowerFactory and Simulink

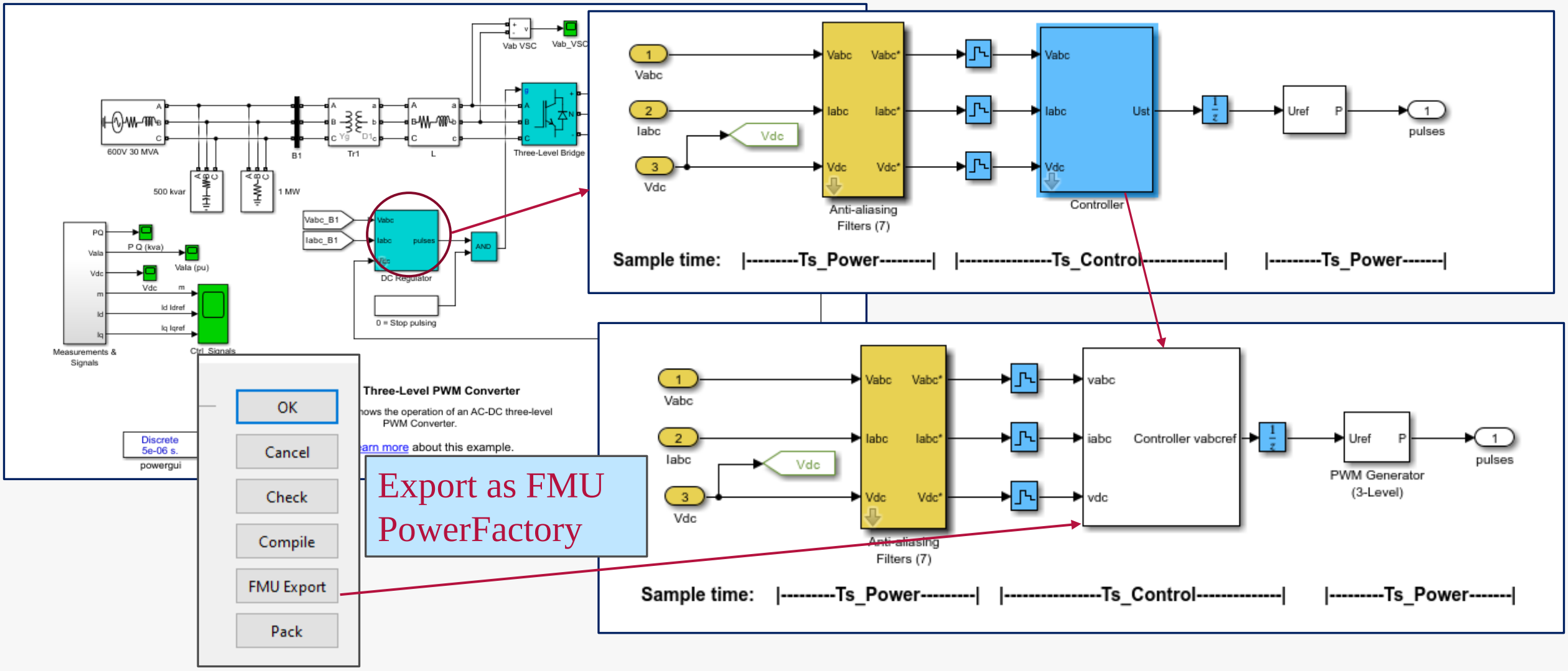
Comparison of Discrete Control



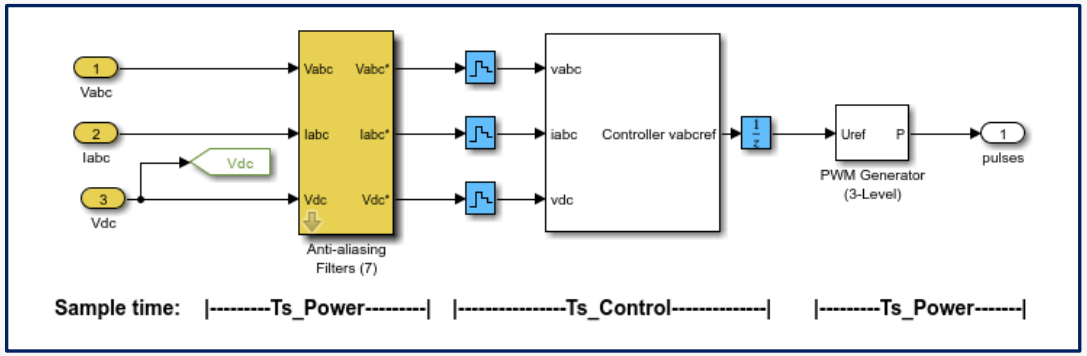


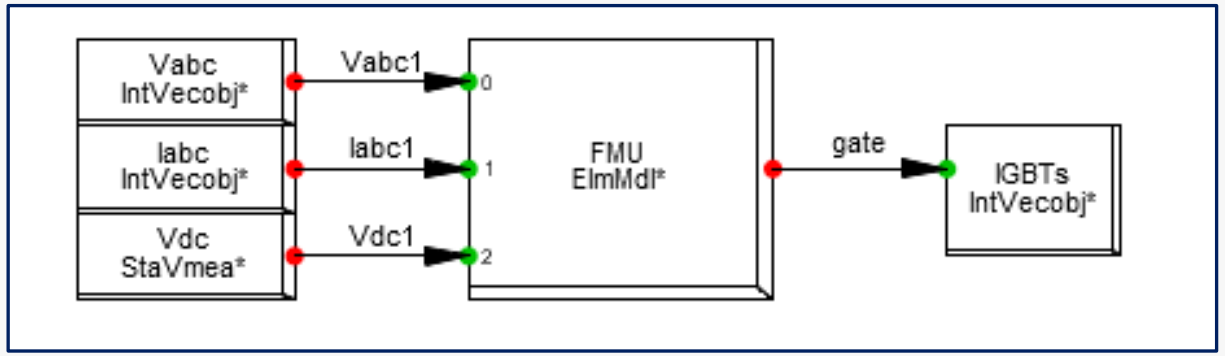
- Comparison of results between PowerFactory (EMT) and Matlab-Simulink
- AC/DC Three-Level PWM converter controllers are implemented using Modelica diagram modelling

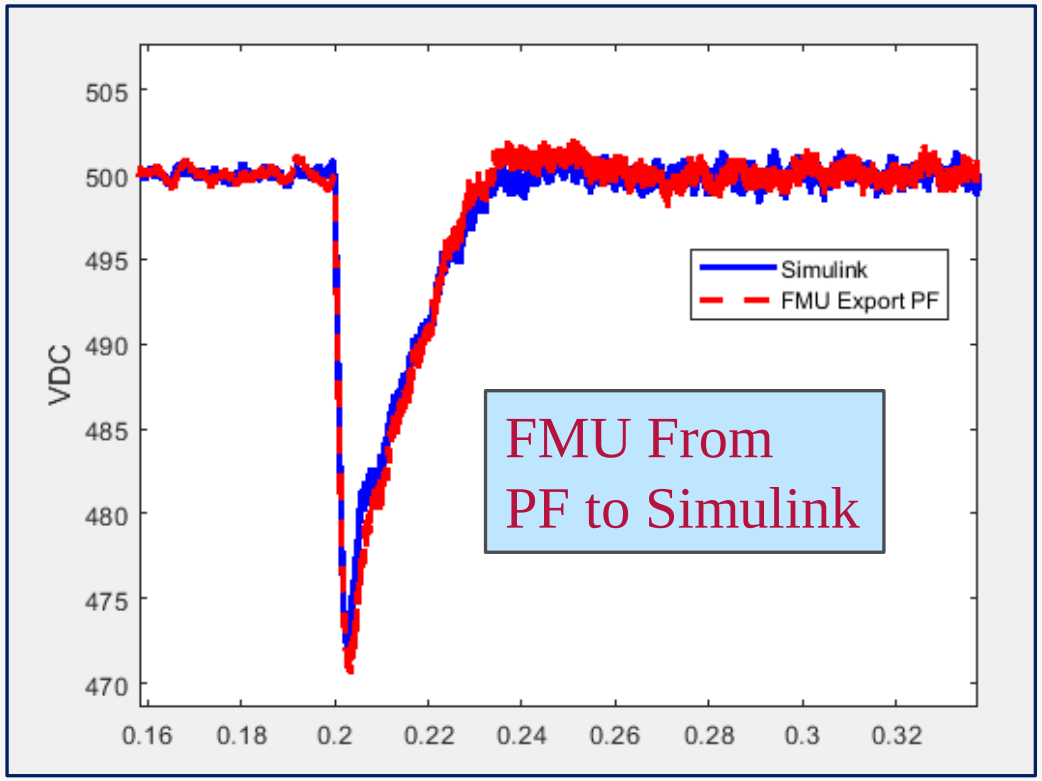
Interoperability of Modelica Models

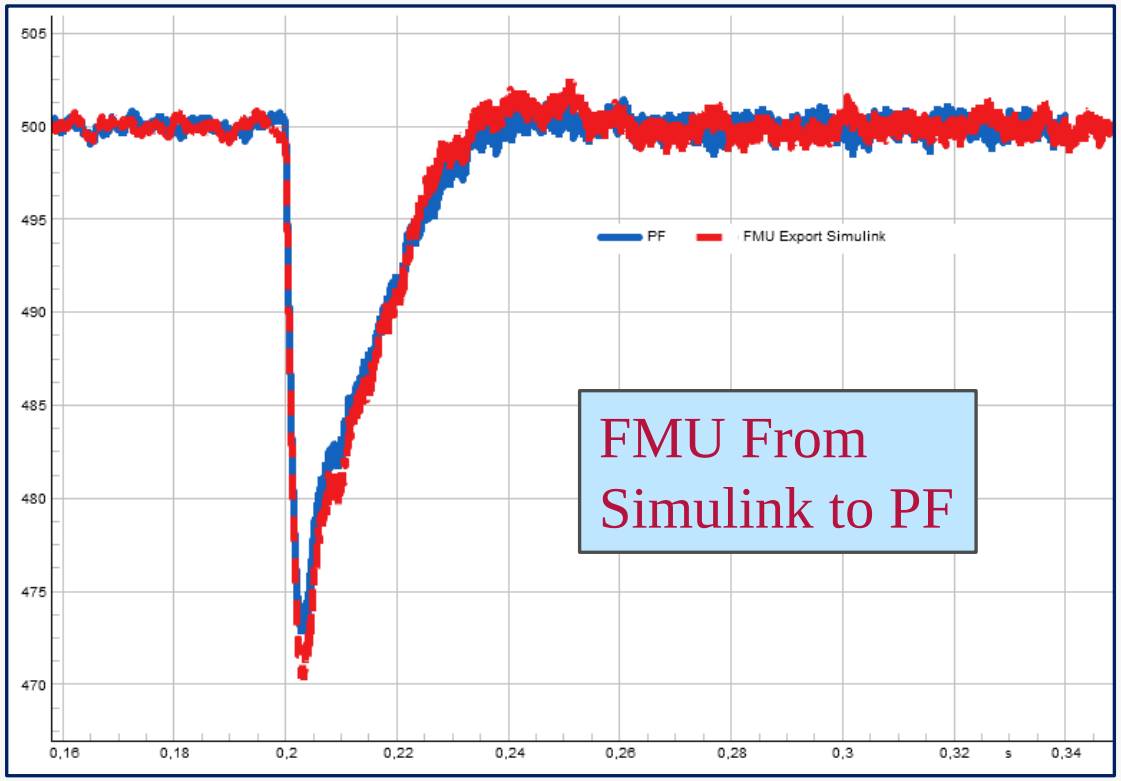


Interoperability of Modelica Models









An Outlook from DIgSILENT

Outlook and Conclusion

- **Conclusion**
  - ❑ Structured discrete drawn/algorithmic dynamic model using Modelica language
  - ❑ Modelica models are interoperable as model exchange and co-simulation using FMI standard
  - ❑ Manufacturer sourced control model import using FMI standard
  - ❑ Solutions for standardized model exchange between tools do exists and should be forced by the industry

- **Outlook**
  - ❑ Enhanced initialization of discrete/clocked Modelica models
  - ❑ Guidelines of importing of FMUs to initialize in PF from power flow solution

POWERFACTORY

Thank you!

POWER SYSTEM SOLUTIONS
MADE IN GERMANY