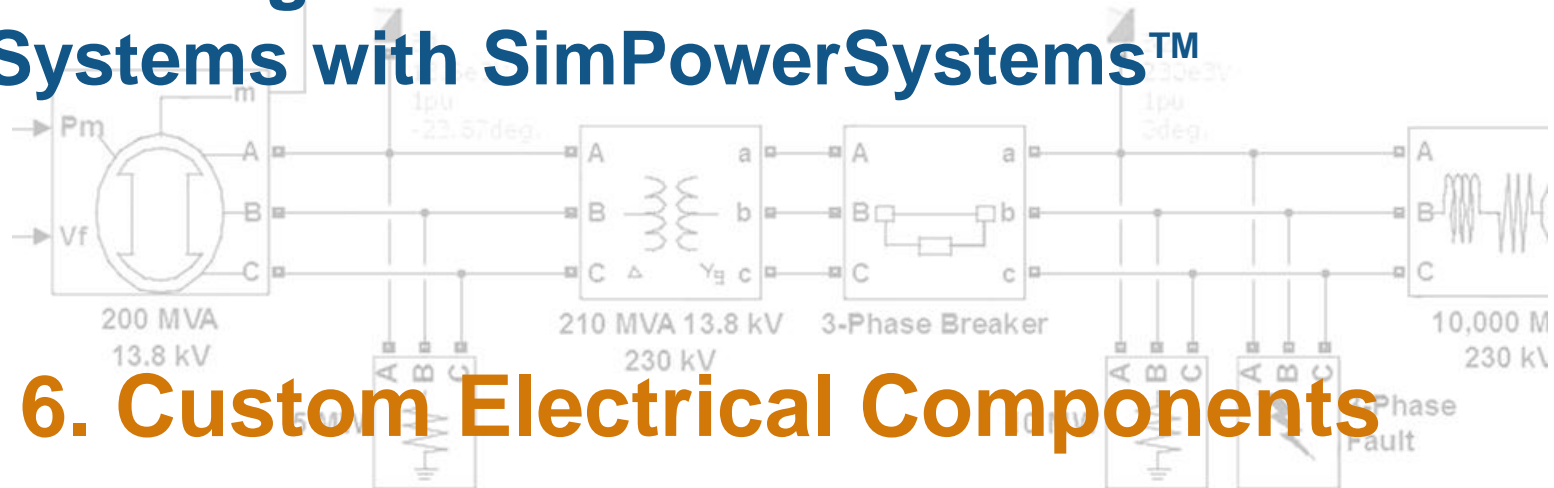


SimPowerSystems Hands-on Workshop: Modeling and Simulation of Electrical Power Systems with SimPowerSystems™



6. Custom Electrical Components



Carlos Osorio

Principal Application Engineer
MathWorks – Natick, MA

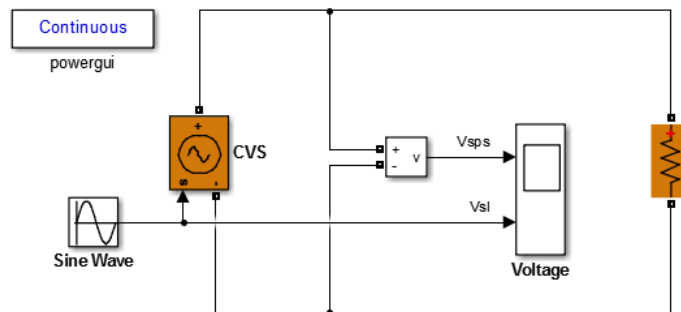
Outline

- Simulink-based electrical components
- Modifying Specialized Technology library components
- Simscape Language and Simscape Components library
- Simscape and SimPowerSystems interfaces
- Model sharing and IP protection

Simulink-based electrical components

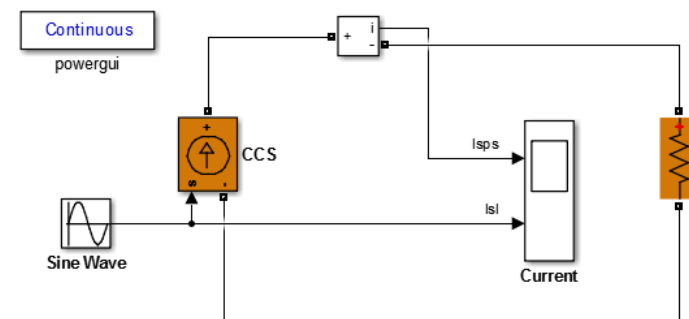
Controlled voltage and current sources

```
>> controlled_voltage_source
```



The **voltage** across the source terminals equals the input Simulink signal, regardless of the current requirement

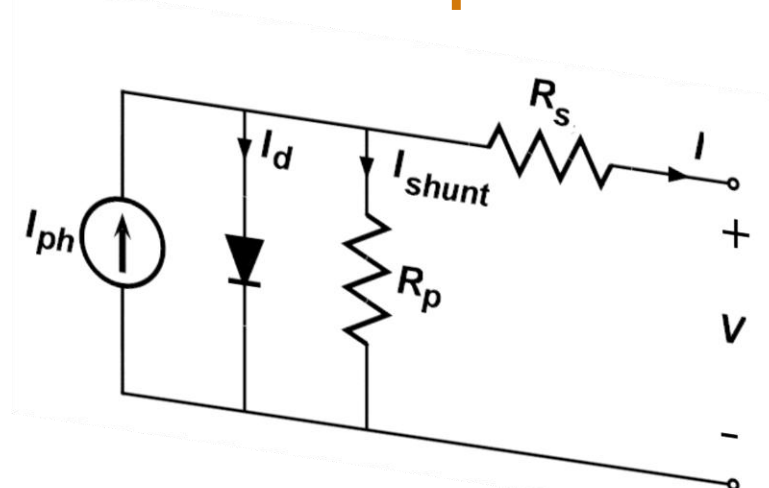
The **current** through the source terminals equals the input Simulink signal, regardless of the voltage requirement



```
>> controlled_current_source
```

Simulink-based electrical components

Photovoltaic solar cell example



$$I = I_{ph} - I_s \left(e^{\frac{V + IR_s}{NV_t}} - 1 \right) - \frac{V + IR_s}{R_p}$$

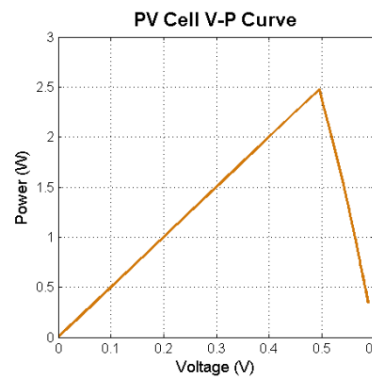
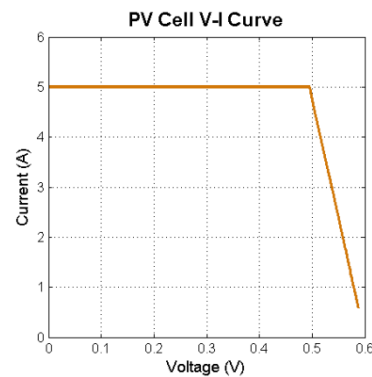
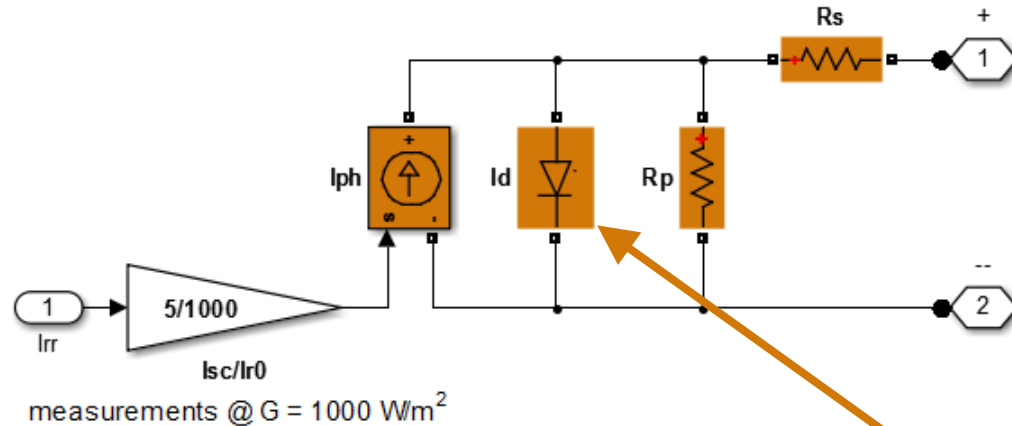
exponential diode

Where:

- I_{ph} Solar induced current (proportional to irradiance)
- I_s Diode saturation current (exponential behavior)
- N Diode quality factor (emission coefficient)
- V_t Thermal voltage kT/q (k : Boltzmann constant, T : device temperature)
- R_p Shunt resistance (models leakage currents, primarily due to defects)
- R_s Series resistance (models bulk and contact resistances)

Simulink-based electrical components

Photovoltaic solar cell example

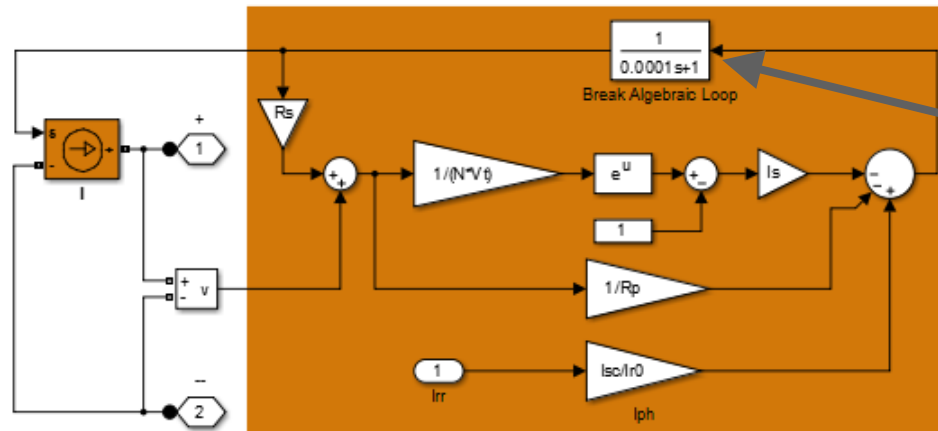


SPS diode is modeled as a piecewise linear switch

```
>> sps_solarcell
```

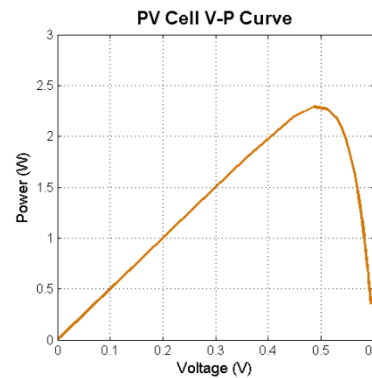
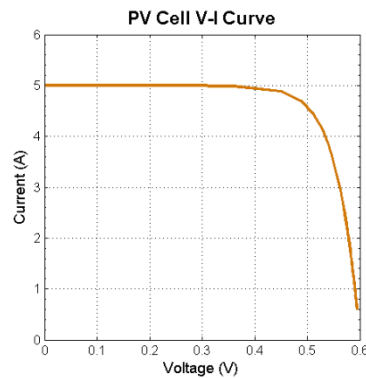
Simulink-based electrical components

Photovoltaic solar cell example



Always need to properly break the algebraic loop

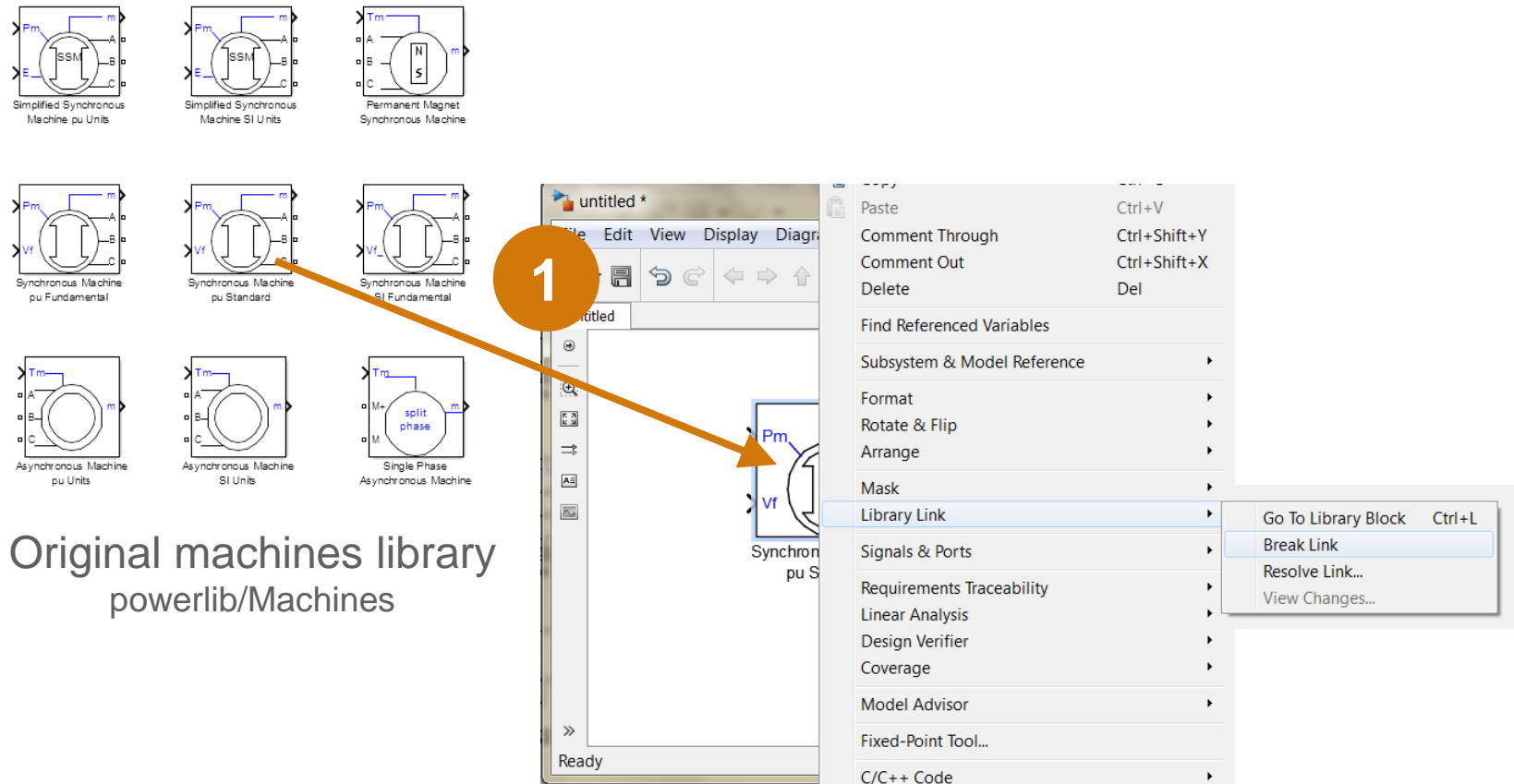
$$I = I_{ph} - I_s \left(e^{\frac{V + I R_s}{N V_t}} - 1 \right) - \frac{V + I R_s}{R_p}$$



```
>> sl_solarcell
```

Modifying Specialized Technology library components

Custom synchronous machine example



Original machines library
powerlib/Machines

1

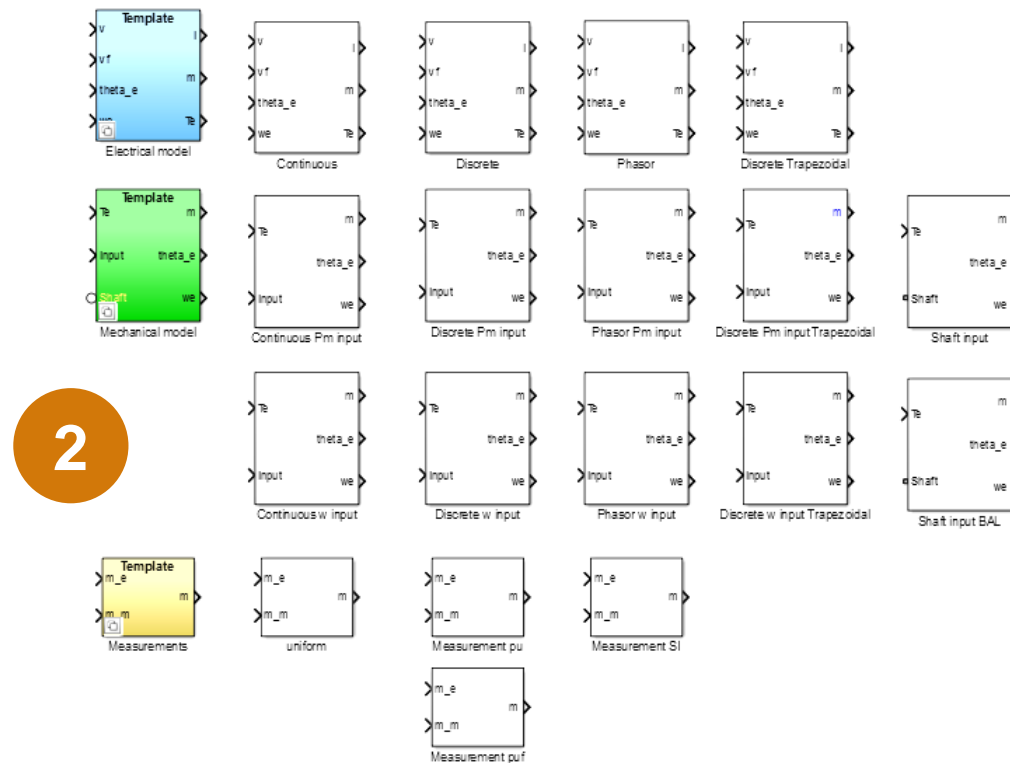
Library Link

Break Link

Break the link and save the original block to a user defined library
i.e. mySMBlocks_userDefinedLib

Modifying Specialized Technology library components

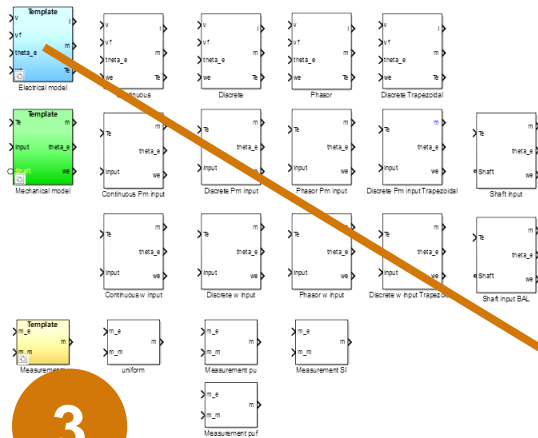
Custom synchronous machine example



Copy the underlying machine model library to a new user defined library
i.e. `spsSynchronousMachineModel` → `mySMMModels_userDefinedLib`

Modifying Specialized Technology library components

Custom synchronous machine example

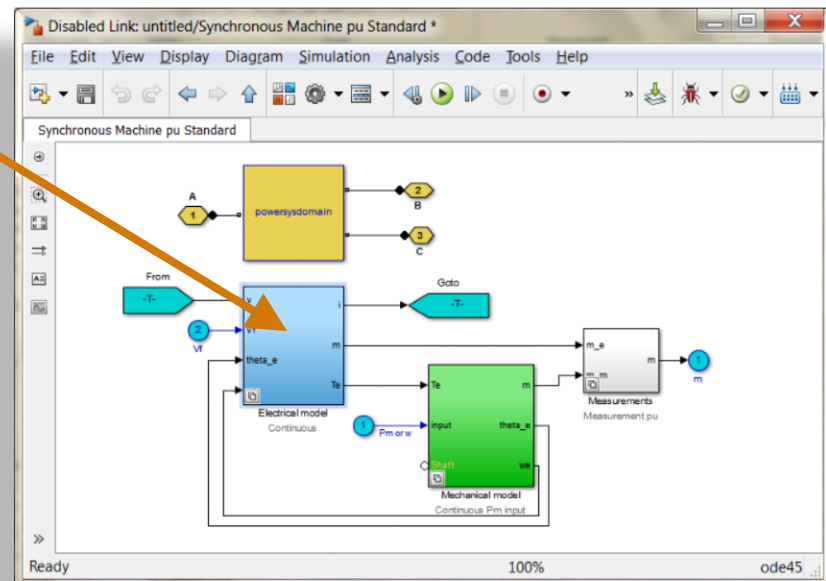


3

Modify any of the blocks in this new library and update the configurable subsystem template blocks
i.e. mySMMModels_userDefinedLib

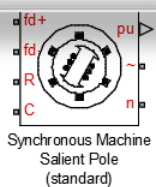
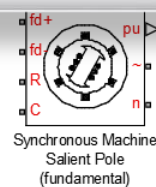
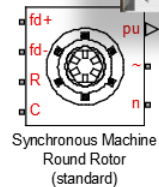
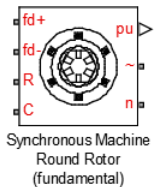
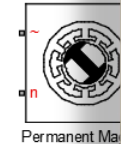
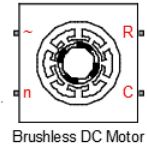
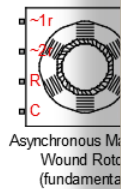
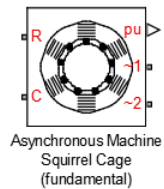
4

Replace the configurable subsystem templates under the mask of the original block with your modified templates
i.e. mySMBlock



Simscape Language and Simscape Components library

Three-phase electrical domain



```

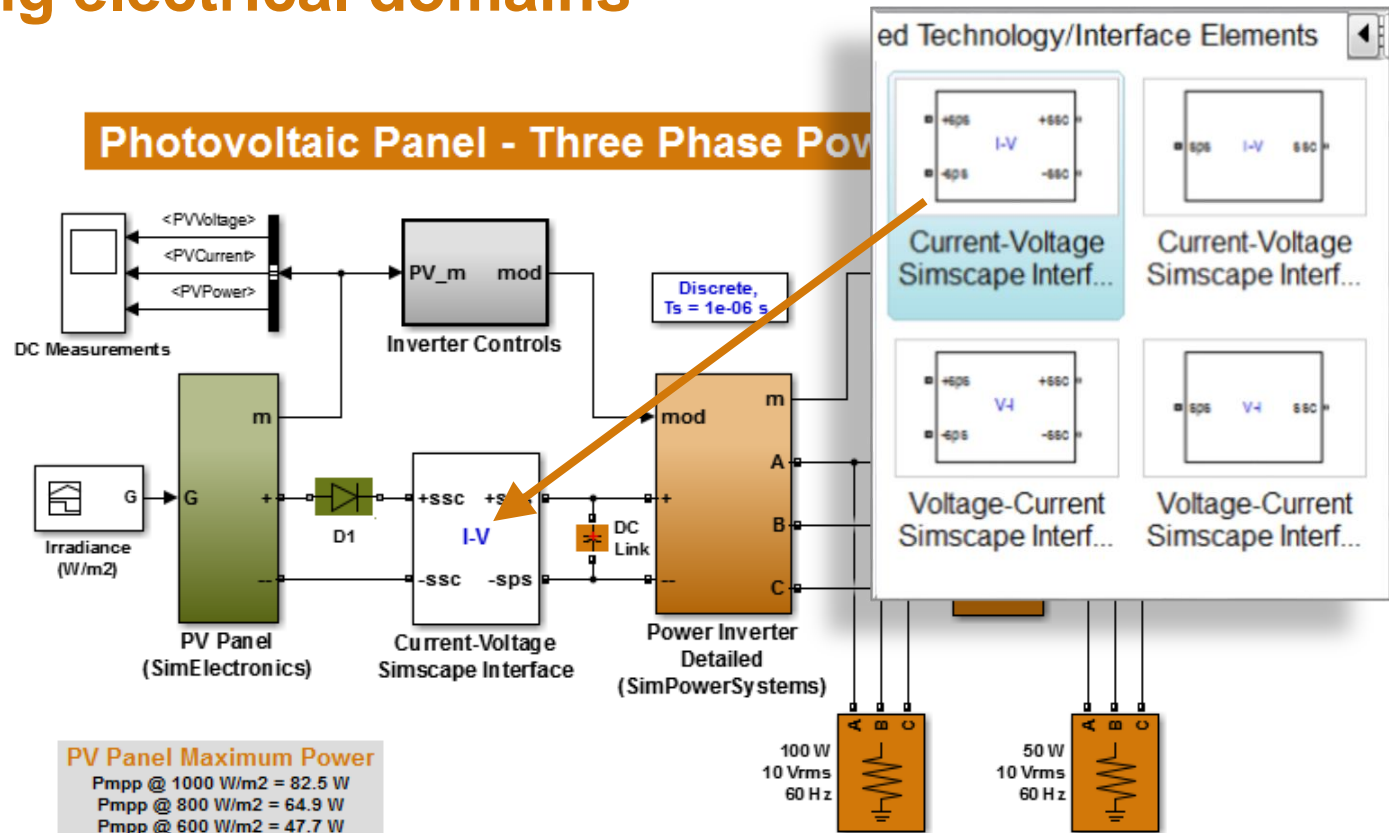
171      0 == oneOverOmega*pu_psi2q.der + R2q*pu_i2q;
172
173      % Per unit stator flux linkage equations
174      pu_psid == -Ld*pu_id + Lad*pu_rotor_circuit_ifd + Lad*pu_ild;
175      pu_psiq == -Lq*pu_iq + Laq*pu_ilq + Laq*pu_i2q;
176
177      % Per unit rotor flux linkage equations
178      pu_psidf == Lffd*pu_rotor_circuit_ifd + Lfld*pu_ild - Lad*pu_id;
179      pu_psidl == Lfld*pu_rotor_circuit_ifd + Ll1d*pu_ild - Lad*pu_id;
180      pu_psilq == Ll1q*pu_ilq + Laq*pu_i2q - Laq*pu_iq;
181      pu_psi2q == Laq*pu_ilq + L22q*pu_i2q - Laq*pu_iq;
182
183      % Mechanical equations
184      torque == ( pu_psid*pu_iq - pu_psiq*pu_id ) * base_torque;
185      angular_velocity == angular_position.der;
186
187      % Per-unit physical signal output
188      pu_velocity == angular_velocity / base_wMechanical;
189      pu_voltage == sqrt( pu_ed^2 + pu_eq^2 );
190  end
  
```

Leverage the **Simscape Language** and the three-phase electrical domain to create custom components

```
>> simscape_custom_generator
```

Simscape and SimPowerSystems interfaces

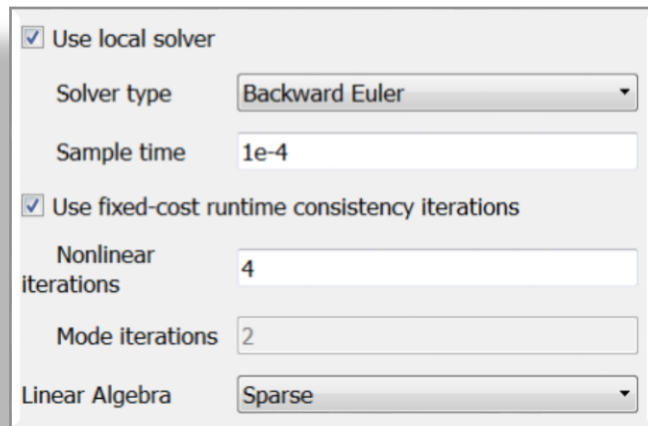
Interfacing electrical domains



```
>> sps2se_pvinverter_avg
>> sps2se_pvinverter_det
```

Simscape and SimPowerSystems interfaces

Interfacing electrical domains

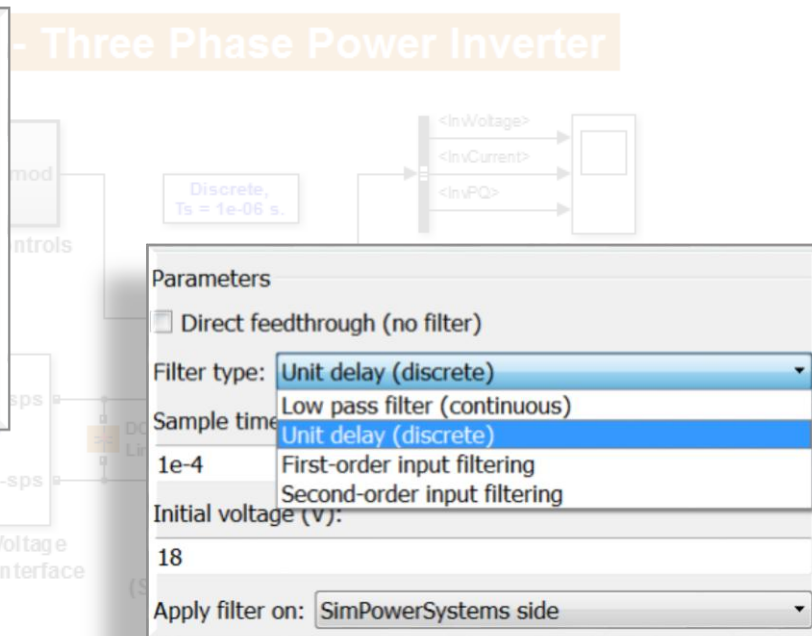


1

Use the Simscape local solver to simulate slower dynamics

2

Use the appropriate filter in the interface block to break the algebraic loop



```
>> sps2se_pvinverter_avg
>> sps2se_pvinverter_det
```

Model sharing and IP protection

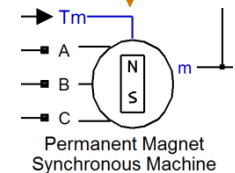
Simscape Editing Mode

- Share SimPowerSystems models with other Simscape users
- Simulate, analyze and generate code without requiring an extra SimPowerSystems license



Function	Full Mode	Restricted Mode
Add or delete regular Simulink blocks	Yes	Yes
Change Simulink solver, simulate	Yes	Yes
Change numerical parameters	Yes	Yes
Access PowerGUI functions, settings	Yes	Yes
Generate code	Yes	Yes
Add/delete blocks from SimPowerSystems	Yes	No
Make or break physical connections	Yes	No
Change block parameterization options	Yes	No

Model Developer
Simscape license
SimPowerSystems license



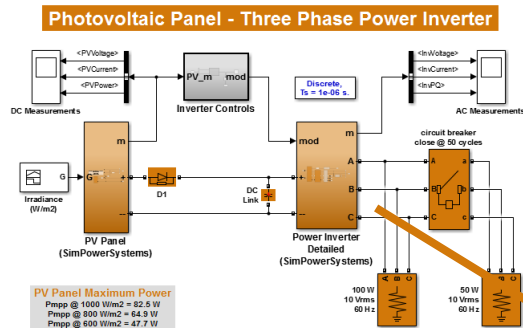
Model using Simscape
and SimPowerSystems



Model Users
Simscape license
SimPowerSystems *installed*,
but no license required

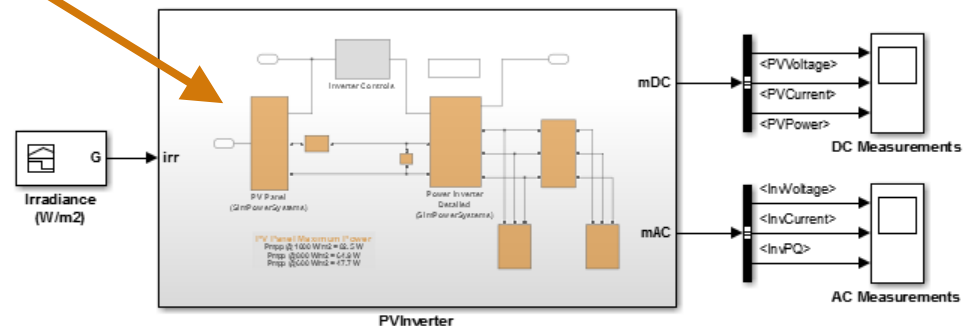
Model sharing and IP protection

Protected model reference



1

Group the part(s) of the model you want to protect in a subsystem – note that only normal Simulink signals can be allowed to cross the subsystem boundaries



```
>> pvinverter_det_vsub
```

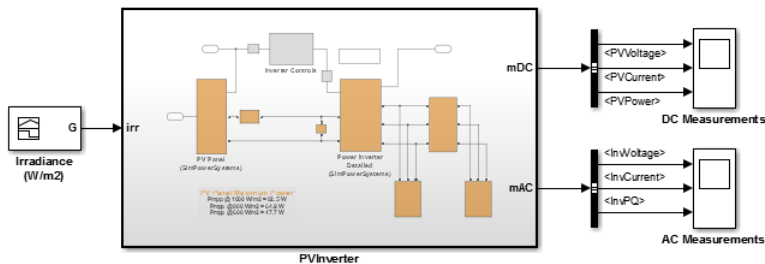
Model sharing and IP protection

Protected model reference

2

Convert the subsystem to an **Atomic** unit
Block Parameters (Subsystem)

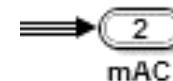
☒ Treat as atomic unit



PV Panel Maximum Power
Pmpp @ 1000 W/m2 = 82.5 W
Pmpp @ 800 W/m2 = 64.9 W
Pmpp @ 600 W/m2 = 47.7 W

3

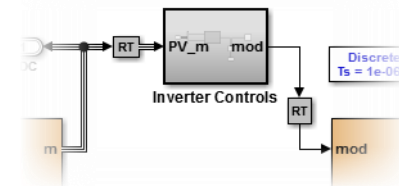
Specify the input/output signal attributes
– data type, port dimensions, sample time, etc. – of the atomic unit explicitly



Create the appropriate <bus object>
data type for any bus ports

4

Change the solver to **FixedStepDiscrete**
Add **Rate Transition** blocks where needed



Check the **Inline parameters** option
Configuration Parameters → Optimization → Signals...

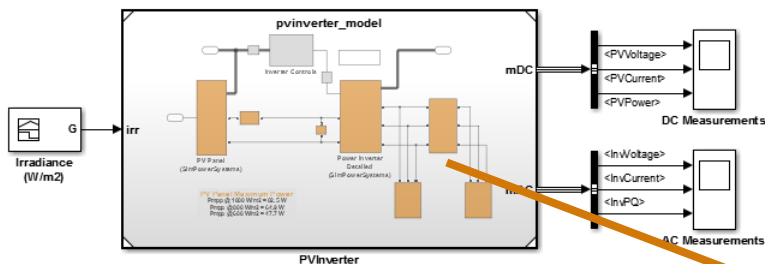
```
>> pvinverter_det_asub
```

Model sharing and IP protection

Protected model reference

5

Convert the subsystem to a **Reference Model**
Subsystem & Model Reference → Convert Subsystem to...

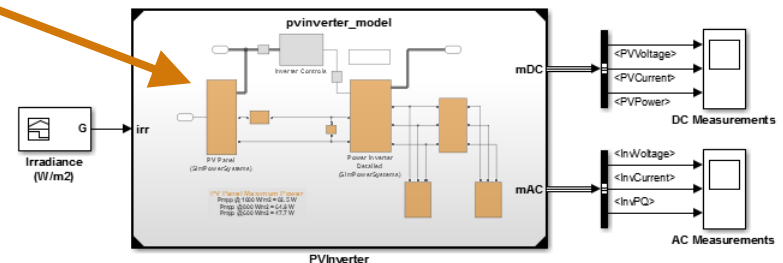
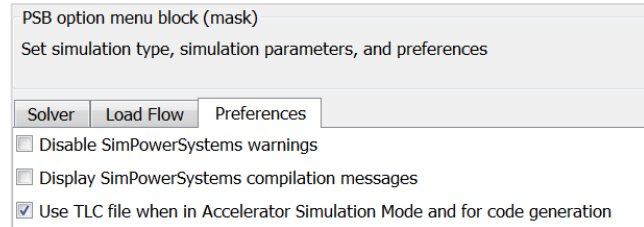


PV Panel Maximum Power
Pmpp @ 1000 W/m2 = 82.5 W
Pmpp @ 800 W/m2 = 64.9 W
Pmpp @ 600 W/m2 = 47.7 W

6

Switch the **Simulation Mode** of the referenced model to **Accelerator**
Block Parameters (ModelReference)

On the referenced model **powergui**
→ **Configure parameters** menu
check the **Use TLC file...** option



```
>> pvinverter_det_mrefacc
```


Model sharing and IP protection

Protected model reference

7

Create the **Protected Model**

Subsystem & Model Reference → Create Protected Model ...

Description
Create a protected model(.slxp) that allows read-only view, simulation, and code generation of the model with optional password protection.

Allow user of protected model to

☐ Open read-only view of model Enter password (optional) Verify password

☒ Simulate Enter password (optional) Verify password

☐ Generate code Enter password (optional) Verify password

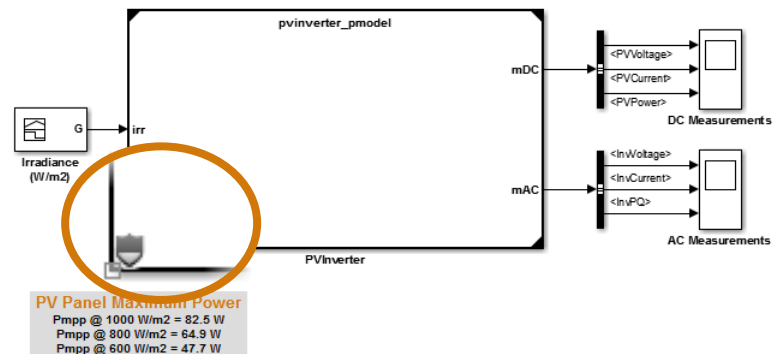
Generated code content type: Binaries

Create protected model in: op\files\06_CustomElectricalComponents Browse...

☐ Create harness model for protected model

Create Cancel Help

Photovoltaic Panel - Three Phase Power Inverter Protected Model Reference



```
>> pvinverter_det_pmref
```

