

# Getting started with Imperix Power library

PN150 | Posted on May 7, 2024 | Updated on June 13, 2025



**Shu WANG**

Development Engineer

imperix • 



**Luca STROPPIANA**

Engineer

imperix

---

## Table of Contents

- [What is the Imperix Power library?](#)
- [Modeling levels](#)
  - [Modeling of power circuits](#)
  - [Modeling of sensors](#)
- [Transistor model type](#)
- [How to use the Imperix Power library?](#)
  - [Typical use case scenarios](#)
  - [Use the averaged model with duty-cycle output from PWM blocks](#)
  - [Global configuration of the blocks](#)
- [Troubleshooting](#)
  - [Powergui settings in Simulink](#)
  - [Initial states in Simulink](#)
  - [Sensor sensitivity in PLECS ADC blocks](#)
  - [State/source dependence issue](#)
  - [Compatibility with HIL simulation](#)
- [Getting started example: three-phase PV inverter simulation](#)
- [To go further](#)

Imperix Power library is a blockset contained within the imperix [ACG SDK](#) that provides rapid and accurate modeling of imperix power products. It allows practicality and accuracy for the plant model during the offline simulation phase. This blockset is available for both Simulink and PLECS.

This product note summarizes all the useful information to get started with the Imperix Power library and provides an example of simulating a three-phase PV inverter with the Imperix Power library.

Starting from [2025.1](#), the Imperix Power library also supports the modeling of the thermal behaviour of power semiconductors on imperix power modules. To get started with thermal simulation with the Power library, please refer the [Thermal simulation guide](#).

Imperix Power library is available starting from ACG SDK 2024.2. Moreover, thermal simulation is enabled from ACG SDK 2025.1.

Simulink Simscape Electrical or PLECS is also required. The Simulink version is only compatible with Specialized Power Systems. The supported versions are:

- Simulink R2016a or newer.
- Plexim PLECS 4.5 or newer.

The Imperix Power library is only used to build the Plant model in offline simulation. For information on how to build a Controller model in Simulink or PLECS, please refer to the following pages:

[Getting started with ACG SDK on Simulink](#)

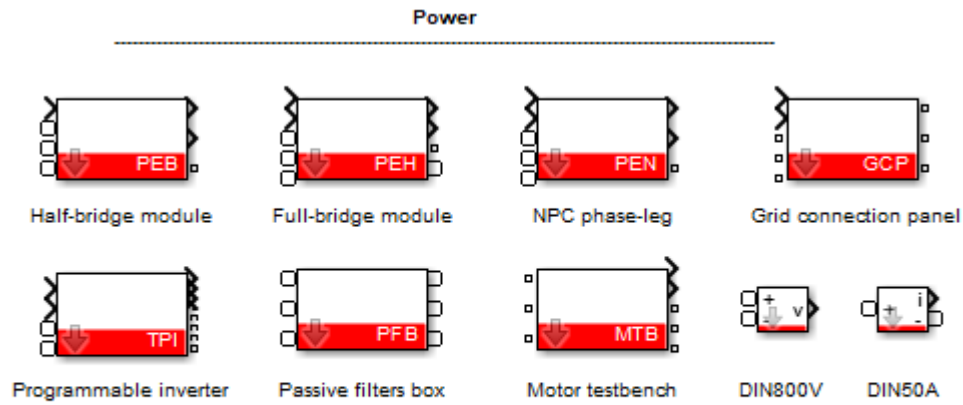
[Getting started with ACG SDK on PLECS](#)

## What is the Imperix Power library?

The Imperix Power library comprises nine blocks, providing simulation models for all the imperix power products. The blocks are classified per circuit topology. For instance, all the [half-bridge power modules](#) are included in one [PEB](#) block with optional variants [PEB 8038](#), [PEB 8024](#), and [PEB 4050](#). The library content is listed below.

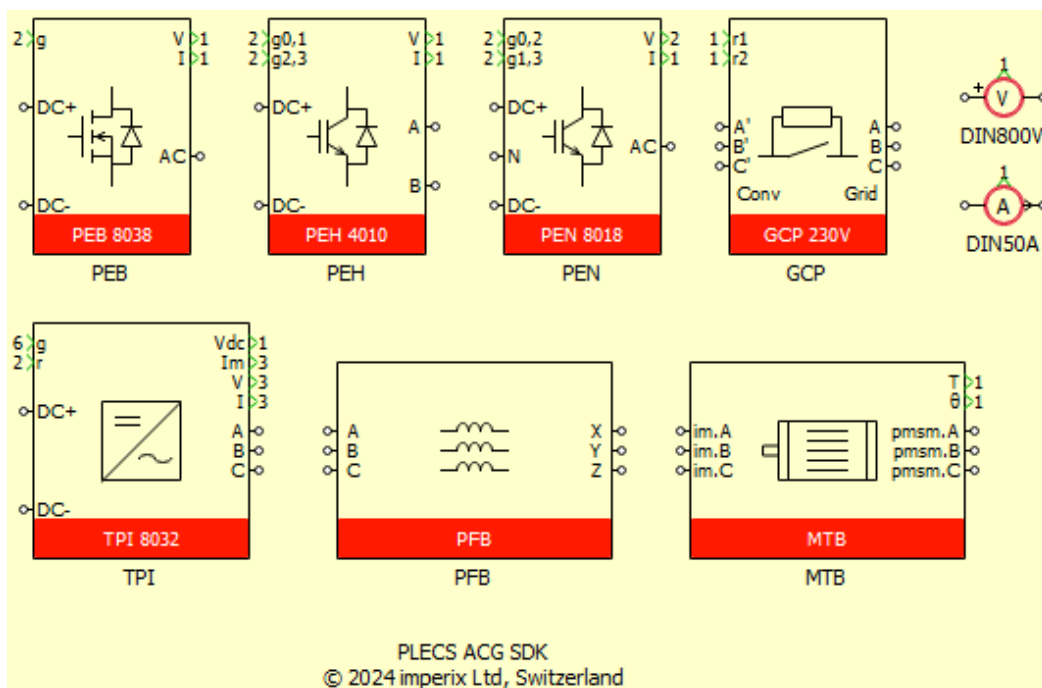
Product name	Block name	Variants
<a href="#">Half-bridge module</a>	<a href="#">PEB</a>	PEB8038, PEB8024, PEB4050
<a href="#">Full-bridge module</a>	<a href="#">PEH</a>	PEH2015, PEH4010
<a href="#">NPC Phase-leg</a>	<a href="#">PEN</a>	PEN8018
<a href="#">Grid connection panel</a>	<a href="#">GCP</a>	AC 230V, AC 110V
<a href="#">Programmable inverter</a>	<a href="#">TPI</a>	–
<a href="#">Passive filters box</a>	<a href="#">PFB</a>	–
<a href="#">Motor testbench</a>	<a href="#">MTB</a>	–
<a href="#">±800 V voltage sensor</a>	<a href="#">DIN800V</a>	–
<a href="#">±50 A current sensor</a>	<a href="#">DIN50A</a>	–

Content of Imperix Power library



Simulink ACG SDK 2024.2  
© 2024 imperix Ltd, Switzerland

### Imperix Power library for Simulink



### Imperix Power library for PLECS

Overall, the modeling procedure of an imperix power product can be divided into two individual tasks, namely:

- Modeling of the peripheral circuits, including the power circuits (DC bus, filters, other passive components, ...) and the sensors.
- Modeling of the power transistors, either modeled by switched devices or controlled voltage/current sources.

The blocks in the Imperix Power library provide two configuration parameters, *Modeling level* and *Transistor model type*, for flexible switching between different models. These configurations are addressed in detail in the following sections.

# Modeling levels

The Imperix Power library offers two options for modeling levels for all the blocks:

- **(A) Simple**
- **(B) Detailed**

As the name indicates, the (A) Simple level is a simplified plant model that only considers the main components in the circuit. Most of the characteristics at the high-frequency range are ignored to achieve a faster simulation speed. Therefore, the Simple level is mostly preferable for system-level modeling. On the other hand, the (B) Detailed level ensures higher modeling accuracy at the cost of increased simulation time. Thus, it is more suitable for circuit-level modeling that requires higher accuracy.

The simulation model is relevant within a given frequency range. The frequency validity range is 0-4 kHz for the (A) Simple level and 0-100 kHz for the (B) Detailed level. In that range, the impedance and transfer function of the model are reasonably close to the modeled system.

Some models only have the (A) Simple level available, such as [GCP](#) and [MTB](#).

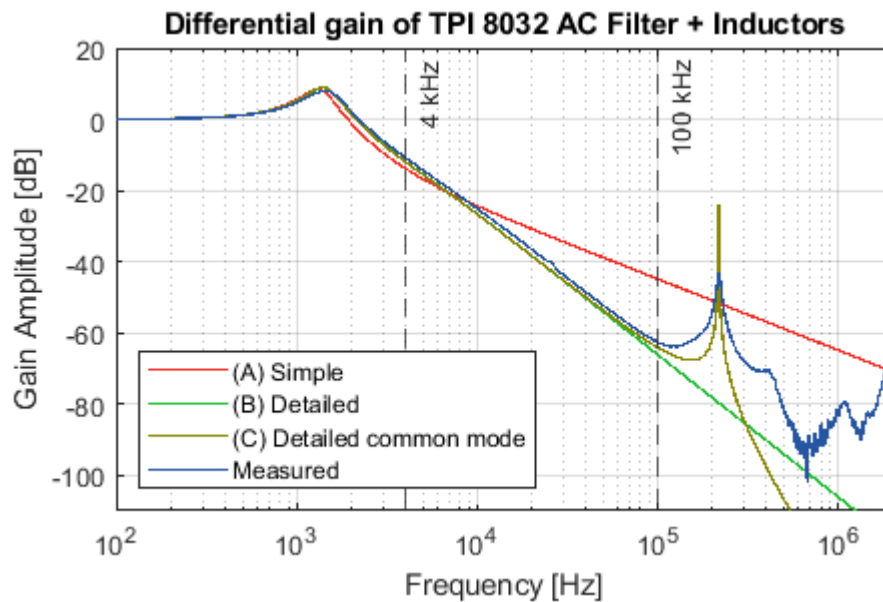
Please note that both modeling levels only include models in the differential mode. This is because the common-mode characteristics mostly lie in the frequency range of up to hundreds of kilohertz to megahertz, which is far above the analog bandwidth of an imperix controller. However, there is only one exception in the [TPI](#) model, which provides an extra modeling level:

- **(C) Detailed common mode**

The frequency validity range of (C) Detailed common mode is also 0-100 kHz. Compared with level (B), it includes the common-mode feedback path between the AC EMC filter and the DC bus. For a detailed explanation of why common-mode modeling can be important in the TPI, even in the low-frequency range, please refer to the dedicated software page for the [TPI](#).

## Modeling of power circuits

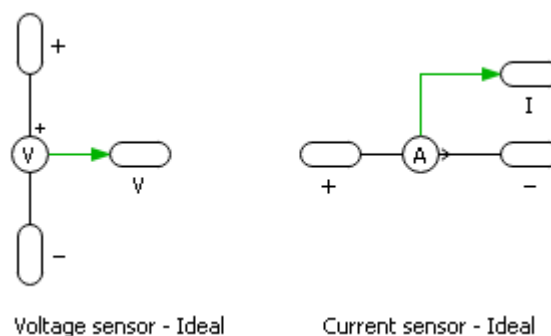
As previously explained, the parameters of the plant models are chosen so that the difference between the model and measured results is minimized. The figure below shows the transfer function of the TPI AC EMC filter model versus measured results as an example. For more detailed information on the model of a certain power product, please read the dedicated software reference page.



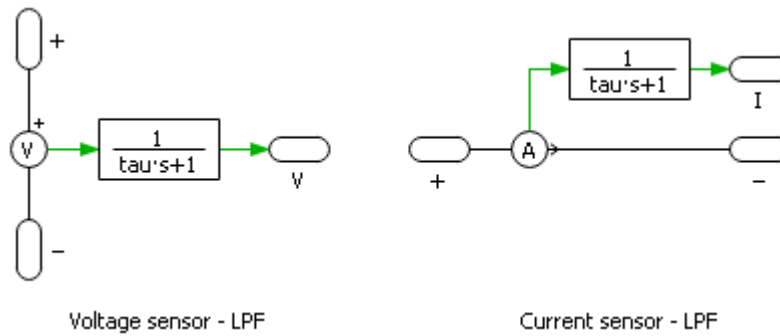
It is important to distinguish “the frequency validity range” in which the simulation model of the plant is reasonably close to reality and the control or switching frequencies of the controller. It is usually not necessary that the converter control and switching frequencies are within the frequency validity range of the plant model. Modeling inaccuracies at 40dB or below are usually negligible. With most of the imperix knowledge base examples that use 20kHz controller frequency, the (A) Simple model provides sufficient accuracy.

## Modeling of sensors

A sensor can be approximately modeled by an ideal sensor in series with an optional 1st-order Low-Pass Filter (LPF). The relationship between sensor’s bandwidth  $f_{BW}$  and the time constant  $\tau$  of the LPF follows  $\tau = \frac{1}{2\pi f_{BW}}$ .



Ideal model



1st-order LPF model

A sensor is modeled as a first-order LPF only if its bandwidth lies within the frequency range of a given modeling level. Otherwise, it is modeled as an ideal sensor. The following table summarizes the information on all the sensors in the Imperix Power library.

Sensor	Bandwidth [kHz]	(A) Simple	(B) Detailed
DIN800V	100	Ideal	LPF
DIN50A	200	Ideal	Ideal
PEB voltage	8	Ideal	LPF
PEB current	280	Ideal	Ideal
PEH voltage	25	Ideal	LPF
PEH current	120	Ideal	Ideal
NPC voltage	25	Ideal	LPF
NPC current	450	Ideal	Ideal
TPI DC voltage	10	Ideal	LPF
TPI module current	360	Ideal	Ideal
TPI AC voltage/current	200	Ideal	Ideal

Modeling of the sensors

## Transistor model type

The Imperix Power library offers two options for modeling the power transistors:

- **Switched** The power transistors are modeled by individual switches. Their control inputs are instantaneous logical gate signals.

- **Averaged** The switches are modeled by controlled voltage and current sources. Their control inputs are the relative on-times of the transistors with values between 0 and 1. The gate signals can be either instantaneous (using only values 0 and 1) or time-averaged.

Although the functionality is practically the same, these options are named differently in PLECS and Simulink Simscape to be consistent with their naming conventions.

In Simulink Simscape they are named:

- **Switching devices**
- **Switching function**

In PLECS they are named:

- **Switched**
- **Sub-cycle average**

This option is only available for the models that contain power transistors, including [PEB](#), [PEH](#), [PEN](#), and [TPI](#). In particular, the [PEN](#) block in Simulink only has the Switched model.

## How to use the Imperix Power library?

### Typical use case scenarios

In practice, a high modeling accuracy is usually achieved at the cost of increased simulation time. The typical use case of the Imperix Power library is shown below.

Typical use case	(A) Simple <i>Preferable for power system users</i>	(B) Detailed <i>Preferable for power electronics users</i>	(C) Detailed common mode <i>Only available in TPI</i>
(A) Switched <i>Preferable in switch-level simulation</i>	<ul style="list-style-type: none"> <li>• Voltage / current control at low frequency, typically 20kHz or below;</li> <li>• Phase modulation;</li> </ul>	<ul style="list-style-type: none"> <li>• Voltage / current control at high frequency;</li> <li>• Filter resonances;</li> </ul>	<ul style="list-style-type: none"> <li>• TPI AC precharge;</li> <li>• Space-Vector PWM;</li> <li>• Harmonics injection;</li> </ul>

<b>(B) Averaged</b> <i>Preferable in system-level or HIL simulation</i>	<ul style="list-style-type: none"> <li>• Power flow control;</li> <li>• Energy management;</li> <li>• System-level operation (energizing, discharge, state machines);</li> <li>• Primary, secondary, and tertiary control;</li> </ul>	/	/
--	---	---	---

Typical use case scenarios

Please note that this table is more of a recommendation than a hard limit.

## Use the averaged model with duty-cycle output from PWM blocks

The Averaged model can also accept duty-cycle inputs between 0 and 1. In this case, the PWM modulator can be removed from the simulation model and the simulation time can be significantly reduced. To use this feature, open the [CB-PWM](#) block in Simulink or PLECS, and find the *Simulation output type* option.

PWM carrier

Carrier shape

TRIANGLE

Main parameters

Duty-cycle (0 to 1)

0

☒ Block input

Phase (0 to 1)

0

☐ Block input

PWM activation

☐ Show "activate" input

Update rate

PWM parameters update rate

Single-rate

Only applicable to triangle and inverse triangle carriers.

Simulation outputs

Simulation output type

PWM signals

CB-PWM block in Simulink

CB-PWM - Carrier-based PWM (mask) (link)

Generates carrier-based PWM signal(s).

- The input signal 'D' is the duty-cycle(s).
- The input signal 'P' is the carrier phase-shift(s).
- The input signal '>' is the clock input.
- The last input 'A' allows the activation (1) or deactivation (0) of the PWM output(s).

Addressing    Modulation parameters    Complementary signal parameters

Carrier type:  ☐ PWM parameters update rate:  ☐

Duty-cycle value:  ☐ Duty-cycle(s) [0 to 1]:  ☐

Carrier phase-shift:  ☐ Phase-shift(s) [0 to 1]:  ☐

PWM activation:  ☐ Simulation output type:  ☐

CB-PWM block in PLECS

When the *simulation output type* is set to *Duty-cycle*, the output of the CB-PWM block will be updated at the end of each controller step, and the PWM phase will be ignored. Therefore, this option is not suitable for phase modulation techniques, such as [TN115](#).

Please note that this option only applies to simulation, and doesn't affect the controller's physical output in code generation.

## Global configuration of the blocks

The options of *Modeling level* and *Transistor model type* can be set individually for each block. However, for a large plant model with numerous blocks from the Imperix Power library, a global configuration for all the blocks is usually preferred. The global configuration is available in both Simulink and PLECS.

### Simulink

In Simulink, the global configuration of the Imperix Power library blocks is managed by the [Config](#) block. The detailed procedure is as follows:

1. In the Simulink model, open the block where the global configuration is to be applied. Tick the box *Use global configurations*. Note that the global configuration will only be applied to blocks with this box ticked. The other blocks will have their local configuration unchanged.

Parameters

☒ Use global configurations

Note: global configurations are only applied when clicking the button in CONFIG block.

Modeling level (A) Simple ▼

Transistor model type Switching devices (switched) ▼

### Global configuration in Simulink power library blocks

2. In the Simulink model, open the *Config* block, go to the *Power library* tab, and select the proper options for *Modeling level* and *Transistor model type* here. As the averaged model is usually used together with the duty-cycle outputs from the CB-PWM blocks, the simulation output type of [CB-PWM](#) and [TPI CB-PWM](#) is also globally configured here.

Control task Advanced sampling Simulation **Power library**

Modeling level (A) Simple ▼

Transistor model type Switching devices (switched) ▼

CB-PWM output type PWM signals ▼

The changes are only applied when clicking the button below:

Apply changes to CB-PWM and power library blocks

Use Imperix Power library for simulation of imperix products.  
[Open Imperix Power library](#)

### Global configuration in the Simulink Config block

3. Click on the button to apply changes to CB-PWM and power library blocks. The changes will be applied **only when the button is clicked** and **not automatically updated** each time at the beginning of the simulation.

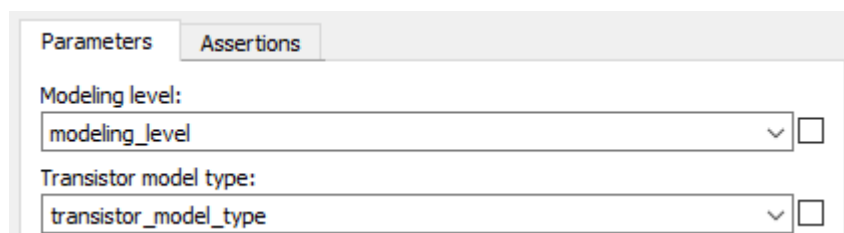
## PLECS

In PLECS, the global configuration can be easily controlled by a global variable using the <reference> option in the masked block. The detailed procedure is as follows:

1. In the PLECS model, open the *Simulation parameters* window (Ctrl + E) and go to the *Initialization* tab.
2. Define two variables `modeling_level` and `transistor_model_type` in the Model initialization commands, so that these two variables will be initialized during compilation. The values of these two variables are as follows:

Variable	= 1	= 2
modeling_level	(A) Simple	(B) Detailed
transistor_model_type	Switched	Sub-cycle average

- Open the PLECS block where the global configuration is to be applied, choose <reference> for *Modeling level* and *Model of power module* options, and then put the two variables inside.



Global configuration in PLECS blocks

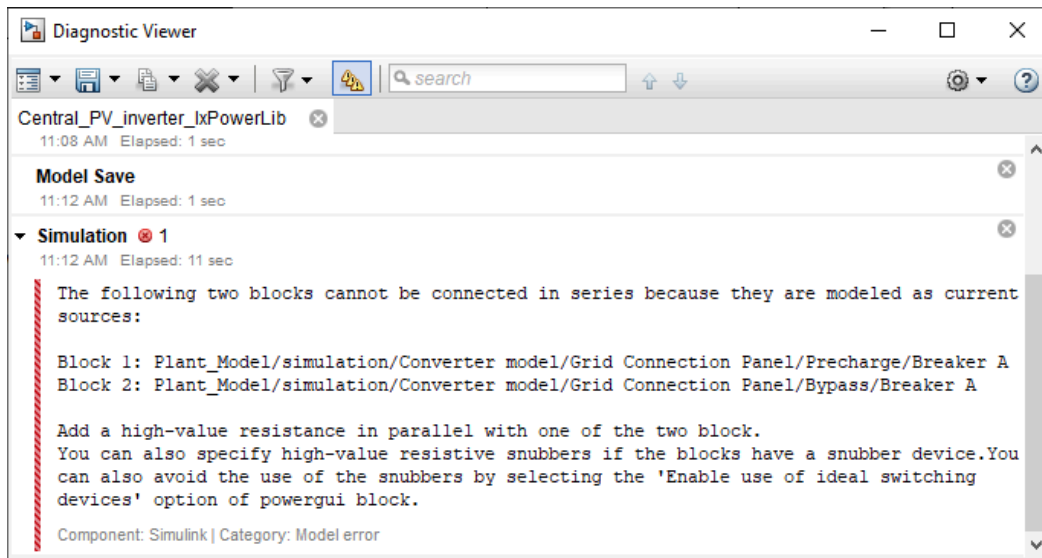
- Change the values of the two variables to apply a different global configuration. The changes will be **automatically applied** each time at the beginning of the simulation.

## Troubleshooting

This section summarizes some possible issues and solutions when using the Imperix Power library.

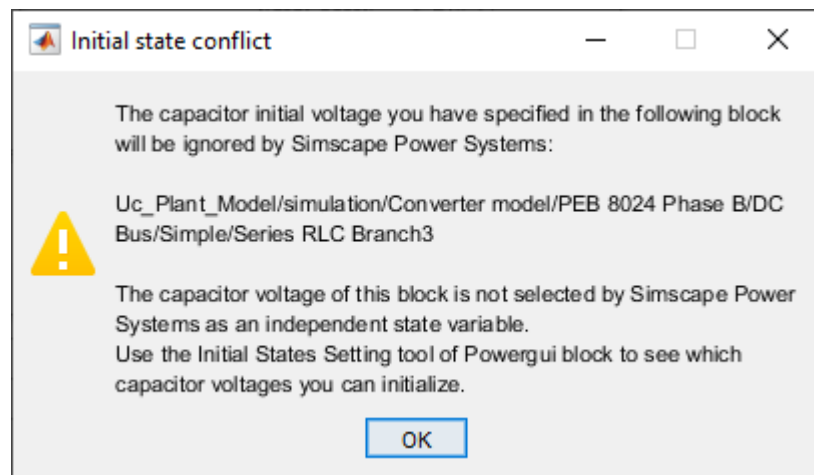
## Powergui settings in Simulink

As introduced in [PN135](#), it's recommended to use a **Continuous** plant model for an accurate and efficient simulation. Particularly, when using Imperix Power library blocks in MATLAB R2016a, it's also recommended to tick the box *Use ideal switching devices*. Otherwise, Simulink may report an error when simulating the relays included in the [TPI](#) and [GCP](#) blocks.



## Initial states in Simulink

For those blocks including DC bus capacitors, the initial voltage across the DC bus can be defined from the block's mask. However, when multiple DC bus capacitors with initial voltages are connected in parallel, Simulink may report a warning of the initial state conflict. This warning usually occurs when simulating with the (A) Simple modeling level, where the DC bus is modeled by ideal capacitors without Equivalent Series Resistance (ESR).



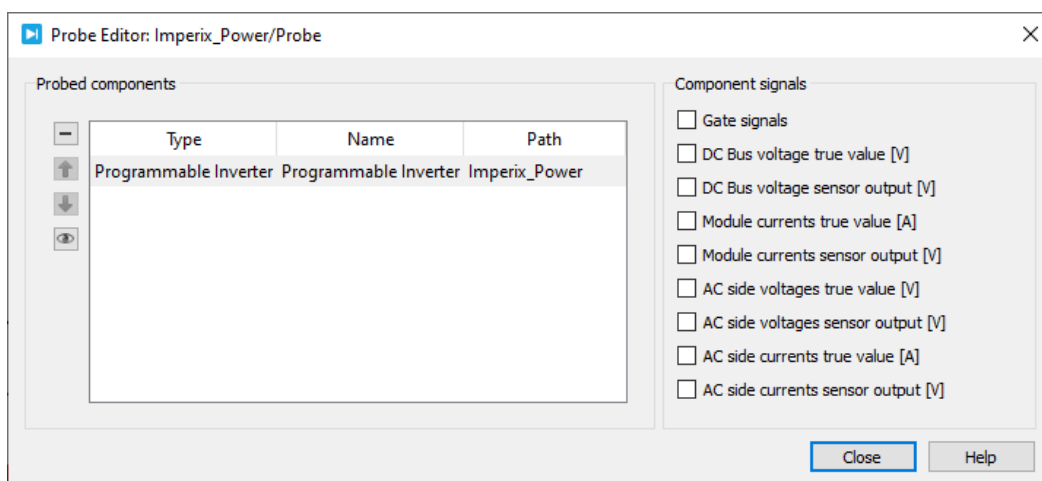
The initial state conflict will not affect the simulation results, as long as all the DC bus capacitors in parallel have the same initial voltage. However, it can be quite annoying to see these pop-up windows at the beginning of each simulation. Unfortunately, there is no good way to actively hide this warning message. As a possible workaround, the initial state conflict can be resolved by simulating the ESRs of DC bus capacitors using (B) Detailed modeling level.

## Sensor sensitivity in PLECS ADC blocks

Starting from ACG SDK 2024.2, the [ADC](#) and [TPI ADC](#) blocks in PLECS have the option to **Simulate sensor(s) sensitivity(ies)**. If this option is ticked, the ADC block expects its input to be the physical value of a sensor's output, and the reciprocal of the sensor sensitivity is applied to the data flow to get the true value in the circuit.

For most of the simulation users, the sensor sensitivity is irrelevant information. It's much easier to simply ignore the sensor sensitivity in simulation and use true values everywhere. As such, all the Imperix Power library blocks have the real values as block output. When using the Imperix Power library with PLECS ADC blocks, please don't forget to untick the **Simulate sensor(s) sensitivity(ies)** option.

On the other hand, the sensor sensitivity can be useful for Hardware-in-the-loop (HIL) simulation, when the true values must be scaled to RT Box's analog output range. In this case, a PLECS Probe block can be used to get the sensor output values. Drag the Probe block from the PLECS library into the model that contains the circuit. Double-click the icon to open the probe editor window, and drag the Imperix Power library blocks into the window as shown below.



Probe signals of the TPI block

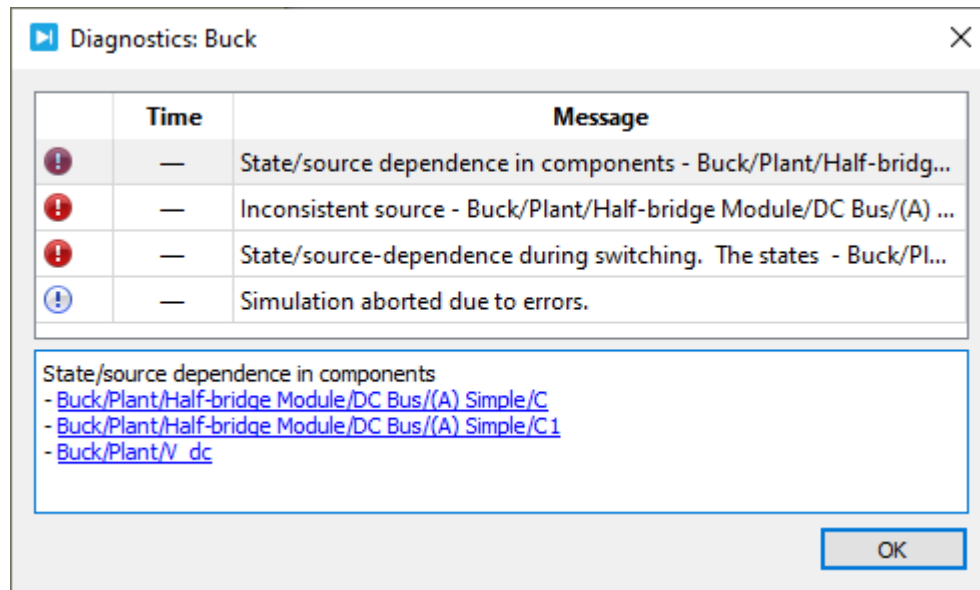
In the next section, a PLECS model example is provided to show how to deal with the true values and sensor outputs when using the Imperix Power library.

## State/source dependence issue

As introduced in [this note from Plexim](#), Simulink and PLECS report the state/source dependence issue when a capacitor is directly in parallel with a voltage source, or an inductor is directly in series with a current source.

When using the Imperix Power library, this problem usually happens when the plant model contains a power module block directly connected to a DC source. This is because the Simple level ignores the Equivalent Series Resistance (ESR) of the DC

Bus capacitors. For example, in a plant model of a Buck converter, an error message will pop up in the simulation if the PEB block uses the Simple level.



PLECS error message

There are two ways of solving this issue:

1. Use the (B) Detailed level instead for the power modules directly in parallel with a voltage source.
2. Insert a small resistor (typically a few milliohms) in series with the DC source. However, the resistance must be chosen carefully so that it does not influence the system dynamics too much (if it is too large) or create numerical problems (if it is too small).

## Compatibility with HIL simulation

The PLECS blocks in the Imperix Power library are compatible with the Hardware-in-the-loop (HIL) simulation using the [B-Box RCP](#) and [Plexim RT-Box](#) (using Plexim's sub-cycle average model). The PLECS\_RT\_Box target support package is also required for HIL use.

## Getting started example: three-phase PV inverter simulation

The Simulink and PLECS models provided below implement a three-phase PV inverter, where the plant model is built with blocks from the Imperix Power library. The controller model is the same as the one in [AN006](#).

## Simulink model

[Download Central\\_PV\\_inverter\\_Simulink\\_powerlib\\_v2](#)

## PLECS model

[Download Central\\_PV\\_inverter\\_PLECS\\_powerlib\\_v2](#)

The file below contains the PLECS model with a **Hardware-In-the-Loop (HIL)** configuration that can be used with the [B-Box RCP](#) together with a [Plexim RT-Box](#).

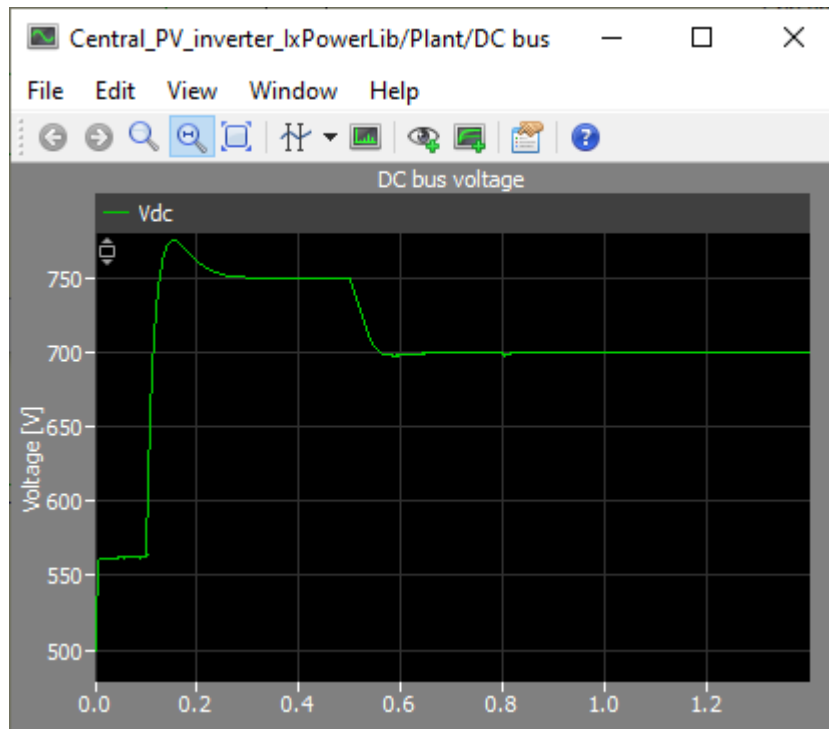
[Download Central\\_PV\\_inverter\\_PLECS\\_HIL\\_powerlib\\_v2](#)

### Minimum requirements:

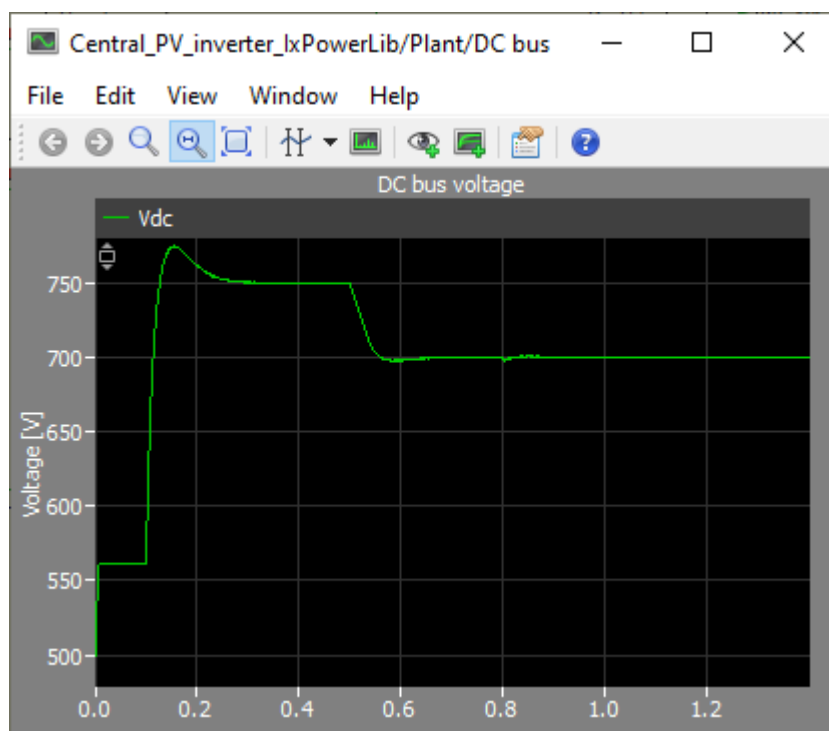
- Imperix ACG SDK 2024.2 or newer.
- For control code development and simulation in Simulink:
  - MATLAB Simulink R2016a or newer.
  - Simscape Electrical
- For control code development and simulation in PLECS:
  - Plexim PLECS 4.5 or newer. The HIL version also required the PLECS\_RT\_Box target support package.

The example models have the following simulation scenario, which is illustrated by the simulation results below:

- At t=0, the DC bus is already precharged at 500V and the PWMs are not yet activated.
- At t=0.1s, the “activate” variable is set to 1. The PWMs are immediately activated (FSM is bypassed in simulation), the DC bus voltage is regulated at 750V, and the MPPT algorithm converges gradually to the MPP (corresponding to a PV current reference around 18A).  
After the MPPT algorithm has converged (after t=0.22s), the Perturb and Observe algorithm gives a PV current reference that oscillates around the MPP current.
- At t=0.5s, the DC voltage reference is reduced to 700V.
- At t=0.8s, the solar irradiance is reduced (simulation of passing clouds). The MPPT algorithm gradually decreases the PV current reference and reaches the new MPP at around t=1s (corresponding to a PV current reference around 15A).



DC bus voltage, (A) Simple, averaged



DC bus voltage, (B) Detailed, switched

The simulation time in PLECS is shown below. For a 20kHz control, there is almost no visible difference between the waveforms under different configurations. On the other hand, the simulation time can be significantly reduced by choosing an appropriate plant model. Overall, for most of the 20kHz control applications, it's recommended to use (A) Simple modeling level and averaged model together with duty-cycle outputs of CB-PWM to achieve fast and accurate simulation.

Model configuration	Simulation time [s]
(A) Simple, switched	38.7736
(B) Detailed, switched	62.1835
(A) Simple, averaged	8.42253
(B) Detailed, averaged	8.66998

Comparison of simulation time in PLECS

This note only compares the simulation time and results in PLECS, but similar conclusions can also be drawn if Simulink is used for simulation. Please just be aware that, since the MPPT plot in Simulink costs quite a lot of computational resources, it's recommended to comment out the MPPT plot for better comparison.

## To go further

Thermal modeling of power semiconductors can be useful in estimating the losses and optimizing the controller. To start thermal simulation with Imperix Power library, please read:

- [PN132 – Thermal simulation with Imperix Power library](#)

More power library examples are available on the Knowledge Base, covering a wide range of application scenarios.

- [AN015 – Solid-state transformer \(SST\) for MV-LV smart grids](#)
- [TN132 – Getting started with an NPC converter](#)
- [TN138 – V/f control of an induction machine](#)
- [TN167 – Grid-Following Inverter \(GFLI\)](#)