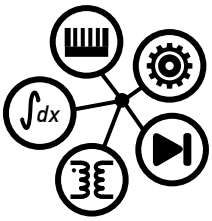

PLECS

DEMO MODEL

Plant Code Generation: Three-Phase 6-Pulse Thyristor Converter

Last updated in PLECS 4.5.1

1 Overview

This example demonstrates PLECS Standalone code generation capabilities for physical systems including electrical circuits. In this model ANSI-C code is generated to represent a three-phase thyristor (SCR) rectifier circuit. The model performance using generated C code is benchmarked against the baseline system using native PLECS components. Code generation for physical systems is essential to model physical plants in a real-time simulation environment, such as with the RT Box family of simulators.

Please note that in order to follow the steps as described below and run the simulation, you will need a PLECS Coder license to enable code generation. To request a trial license for the PLECS Coder, please email Plexim at info@plexim.com.

Note It is important to first create a working copy of this model (**File + Save as...**) before making changes or attempting to generate code.

---

2 Model

This schematic shows a 6-pulse, 3-phase thyristor based converter. In this model, the electrical circuit shown below will be discretized and simulated using generated C code. A detailed explanation of the 6-pulse, 3-phase thyristor converter and the control scheme is given in the demo model “Three-Phase 6-Pulse Thyristor Converter” in the PLECS demo models library.

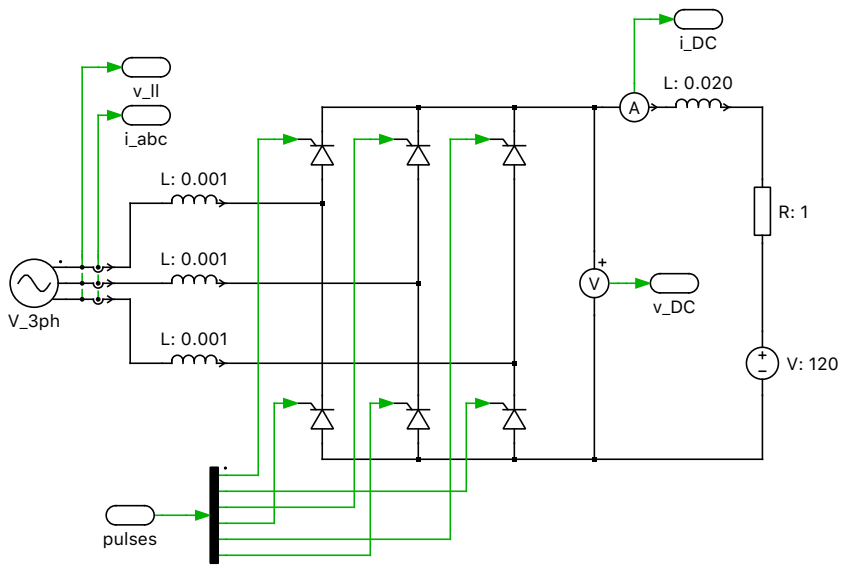

Figure 1: Electrical schematic of a 6-pulse, 3-phase thyristor converter

3 Simulation

3.1 Baseline simulation results

An initial simulation is run using native PLECS components without code generation to record a set of baseline results. The baseline results will be compared with the results using generated code. In the

simulation the DC-side reference current is initially set to zero amps. At 10 ms the reference current is ramped up to 10 A over a duration of 20 ms. At 60 ms the current reference is increased to 25 A. Run the simulation and observe the DC current waveform in the Scope. Notice that the current contains a low-frequency AC-component. Hold the completed simulation trace and label it as “Normal Mode”.

3.2 Discretization and code generation

The next step is to generate ANSI-C code for the plant using the code generation capabilities of PLECS. This requires enabling code generation and specifying the discretization time for the physical system.

To enable code generation, first navigate to the subsystem settings of the electrical circuit contained in the subsystem labeled “Circuit”. This menu is accessed by right-clicking the block labeled “Circuit” and selecting **Subsystem + Execution settings...**. The **Execution settings...** menu can also be accessed by selecting the “Circuit” subsystem, and then selecting **Subsystem + Execution settings...** from the **Edit** drop-down menu . Check the option **Enable code generation**. Note that this option is only shown if you have a license for the PLECS Coder. This will automatically enable the **Treat as atomic unit** option. To allow code generation, the electrical circuit must be treated as an atomic unit. Click on **Apply** and notice how the border of the subsystem is drawn with a thicker line to indicate the “atomic” configuration, as shown in the figure below.

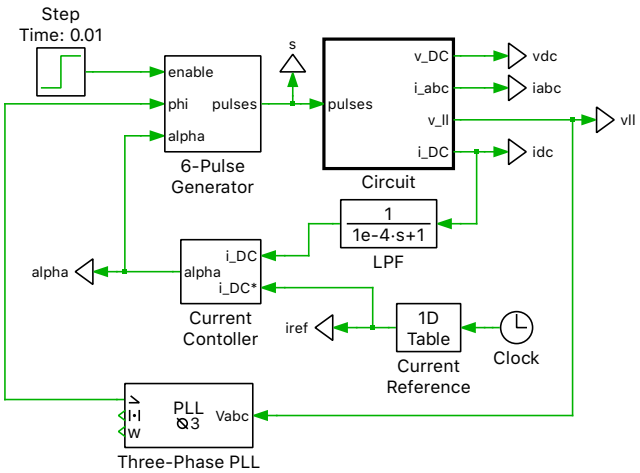

Figure 2: Control schematic with the “Circuit” subsystem modeled as an atomic subsystem

Checking the option **Enable code generation** also causes the subsystem to be added to the list of systems in the **Coder Options** dialog. Open this dialog by choosing **Coder options...** from the **Coder** drop-down menu. In the list on the left hand side of the dialog choose the entry **Circuit**. The right hand side shows various configuration settings for the code generation. Please refer to the PLECS Coder documentation for detailed information about these settings.

The parameter **Discretization step size** specifies the sample time used for the discretization of the electrical model in the “Circuit” subsystem. Begin by discretizing the model with a step size of $1e{-}3\,s$. Then, generate code by clicking the **Build** button. Notice the C files that appear in the specified output directory. The output folder contains one header file and one implementation file. Inspect both files with a text editor. The files are named after the subsystem for which the code was generated (“Circuit”) and have extensions .h and .c file, respectively. The PLECS Coder produces C code according to an API that also permits easy integration with the real-time simulation framework

Checking the option **Enable code generation** in the **Execution Settings** dialog also enables the **Simulation mode** option. This setting is accessed by right-clicking the “Circuit” subsystem and then selecting **Subsystem + Simulation mode**. The **Simulation mode** selection is also conveniently available from the **Edit** drop-down menu. When this option is set to **Normal**, which is the default,

the subsystem is simulated using the contained native PLECS blocks. When the option is set to **CodeGen**, the generated code is automatically compiled by PLECS and executed in place of the subsystem during a simulation. This functionality, combined with the **Hold current trace** feature of the Scope, allows for an effective verification of the behavior of the generated code, by comparing the results against those obtained from a normal simulation.

Select **CodeGen** as the simulation mode from the drop-down menu and apply the changes. The border of the subsystem is now highlighted with dashed lines to indicate that this portion of the model will execute using the generated code. Rerun the simulation and observe the (large) deviation of the simulation results from the previous run. Clearly, a discretization step size of $1\mathrm{e}{-3}\,\mathrm{s}$ is not acceptable. Save the current trace and label it as “CodeGen 1e-3”.

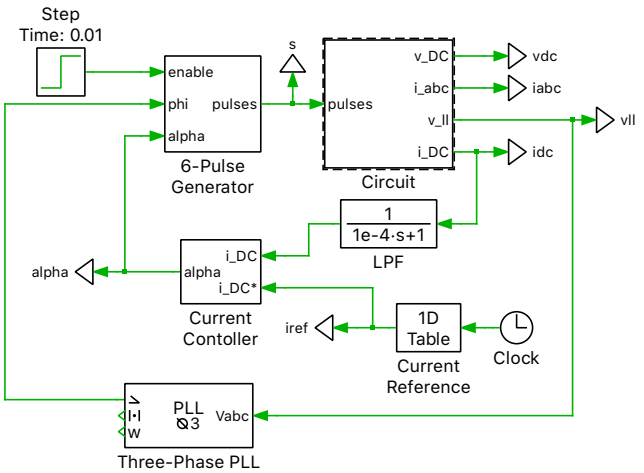

Figure 3: Control schematic with the “Circuit” subsystem modeled using generated code

Now, modify the discretization step size to $1\mathrm{e}{-4}\,\mathrm{s}$. Generate the code for this new step size and apply the changes to the model. Rerun the simulation and save the new trace as “CodeGen 1e-4”. Notice the improved precision of the results when using the generated code with the reduced discretization step size. The figure below shows a comparison of the “Normal Mode” results and the “CodeGen 1e-4” results. Reduce the step size further if desired.

4 Conclusion

The PLECS Coder can generate C code for physical systems including electrical circuits. In this model, a 6-pulse, 3-phase thyristor converter was simulated using native PLECS components and using generated code to represent the electrical subsystem. Using the PLECS Scope, the simulation result using generated code was compared against a benchmark simulation. Setting a discretization time of $1\mathrm{e}{-4}\,\mathrm{s}$ for the electrical circuit shows a good correlation between the native PLECS components and the generated code. The discretization step size could be further reduced for improved precision. Using the PLECS Coder and selecting an appropriate discretization time are important steps in developing real-time simulation models.

For more information on code generation in PLECS, please consult the PLECS User Manual or the PLECS documentation accessed from the Help menu.

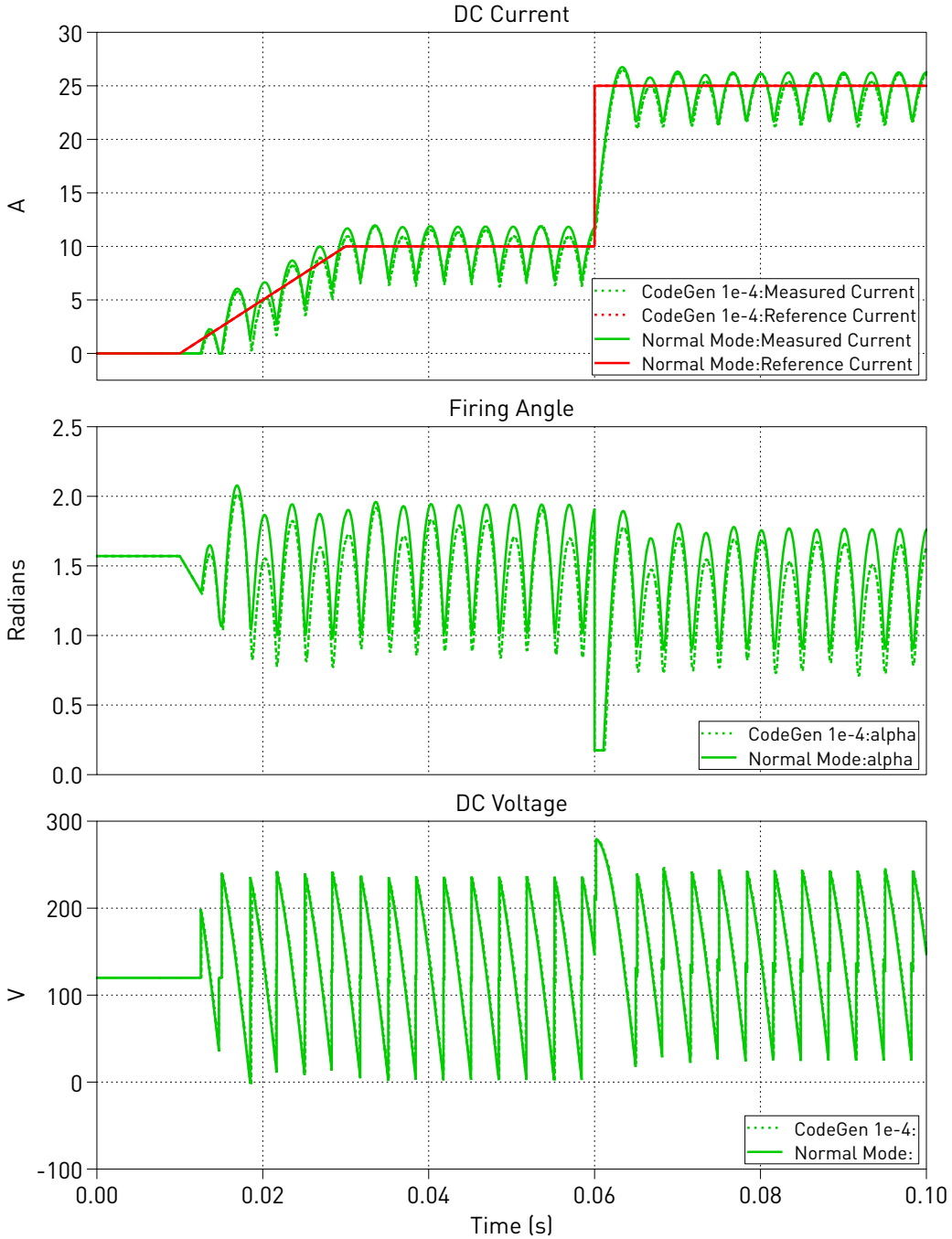

Figure 4: Comparison of "Normal Mode" and "CodeGen 1e-4" Results

Revision History:

PLECS 4.3.1	First release
PLECS 4.5.1	Updating PLL component with new library block

How to Contact Plexim:

☎	+41 44 533 51 00	Phone
	+41 44 533 51 01	Fax
✉	Plexim GmbH Technoparkstrasse 1 8005 Zurich Switzerland	Mail
@	`info@plexim.com`	Email
	`http://www.plexim.com`	Web

PLECS Demo Model