

RT Box

DEMO MODEL

Modular Multilevel Converter

Simulation of a grid-connected modular multilevel converter on the RT Box

Last updated in RT Box TSP 4.0.1

www.plexim.com

- Request a PLECS and PLECS Coder trial license
- Get the latest RT Box Target Support Package
- Check the PLECS and RT Box TSP documentation

1 Overview

This RT Box demo model features a grid-connected modular multilevel converter (MMC) with open-loop controls. The demo model can be simulated either on the FlexArray solver or on the CPU in single-tasking or multi-tasking mode. The model does not need to be split when solved on the FlexArray Solver on the FPGA of the RT 2, 3 or 4. To allow multi-tasking mode, the physical model has to be split into different parts. This can be done with the Task Frame component from the PLECS library. This block associates the enclosed components with a specified task in a multi-tasking environment. For real-time simulation on the RT Box 2, 3 or 4, each specified task is then executed on a different CPU core to reduce the overall discretization step size. For the PLECS RT Box 1 only single-tasking mode is available. The chosen discretization step size and average execution times for the two different tasking modes on the CPU are shown in Tab. 1. The FlexArray step time is summarized in Tab. 2.

Table 1: CPU discretization step size and average execution time of real-time models with both tasking modes for 5 submodules per arm on RT Box 2

Discretization Step Size	Average Execution Time Single-Tasking	Average Execution Time Multi-Tasking
5 μs	4.2 μs	2.3 μs

Table 2: FlexArray discretization step size of real-time model for 5 submodules per arm on RT Box 2

CPU discretization Step Size	FlexArray Step Size
3 μs	1.5 μs

1.1 Requirements

To run this demo model, the following items are needed (available at www.plexim.com[1]):

- One PLECS RT Box CE, 1, 2, 3 or 4[2] and one PLECS Coder[3] license
- One or two 37 pin Sub-D cable to connect the digital I/Os of the box front-to-front
- The RT Box Target Support Package[4]
- Follow the step-by-step instructions on configuring PLECS and the RT Box in the Quick Start guide of the RT Box User Manual[5].

Note

This model contains model initialization commands that are accessible from:
PLECS Standalone: the menu **Simulation > Simulation Parameters... > Initializations**
PLECS Blockset: right click in the **Simulink model window > Model Properties > Callbacks > InitFcn***

2 Model

Two subsystems are placed on top level schematic of the demo model as shown in Fig. 1. The **CPU Multitasking** subsy is configured for single- and multi-tasking execution on the CPU of the RT Box. The

[1] http://www.plexim.com
[2] https://www.plexim.com/products/rt_box
[3] https://www.plexim.com/products/plecs/plecs_coder
[4] https://www.plexim.com/download/rt_box
[5] https://www.plexim.com/sites/default/files/rtboxmanual.pdf

FlexArray subsystem is configured for running on the FlexArray solver of the RT Box 2, 3 or 4. Since the demo model runs in open-loop, the PWM generation and the power circuit run on the same RT Box. To run the subsystem on an RT Box, the subsystem has to be configured as **atomic** and enabled for **code generation** by right-clicking on the subsystem and choosing **Subsystem + Execution settings...**.

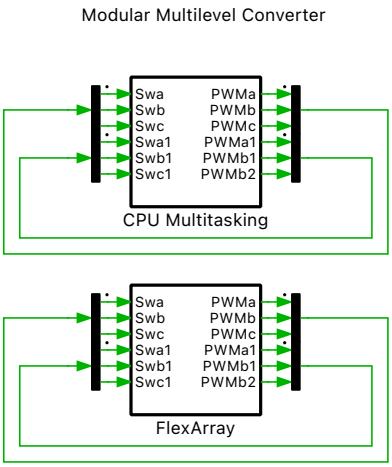


Fig. 1: Top level schematic of MMC demo model

2.1 Power Circuit

Fig. 2 shows the circuit model of the "Plant", which comprises an MMC connecting the AC system and the DC system for execution on the CPU in single- or multi-tasking mode. The MMC has a configurable number of submodules per arm with a default value of 5. Every submodule is composed of one full-bridge and a DC-link capacitor and each single-phase pair of converter arms, together with their arm inductors, is then connected to the AC grid. The converter arms are implemented with the Full Bridges (Series Connected) power module library component. This component has two configurations: a Switched implementation where ideal switches represent the semiconductors, and an sub-cycle averaged configuration that uses controlled voltage and current sources. This model is configured to use the Sub-cycle averaged implementation of the power module components which is suitable for offline and real-time simulation.

The implementation of both the power module and the PWM generation is such that the number of cells can be configured with a variable `num_sm_x` in the Model initialization commands without having to extend the model with additional wiring or components. This concept is called implicit vectorization of the model structure and is further explained in the tutorial "Implicit Model Vectorization" available in the tutorials section of the Plexim website[6].

Tasking modes

The following section is only relevant for the plant in the **CPU Multitasking** subsystem. In order to distribute the physical model on different CPU cores of the RT Box 2, 3 or 4, the model has to be split with the Task frame component from the PLECS library. The tasking mode can be configured in the Coder Options in the **Scheduling** tab of the **Coder Options** window, as shown in Fig. 3 on p. 5.

Single-tasking If the **Tasking mode** is configured as `single-tasking`, all task frame components are ignored and the physical system is executed in a single base task. This configuration is needed for real-time simulation of the demo model on the RT Box 1.

Multi-tasking If the **Tasking mode** is configured as `Multi-tasking` the physical model is split into three different tasks and those tasks are associated with the three available CPU cores of the RT Box 2, 3 or 4, as shown in Fig. 3 on p. 5.

[6] https://www.plexim.com

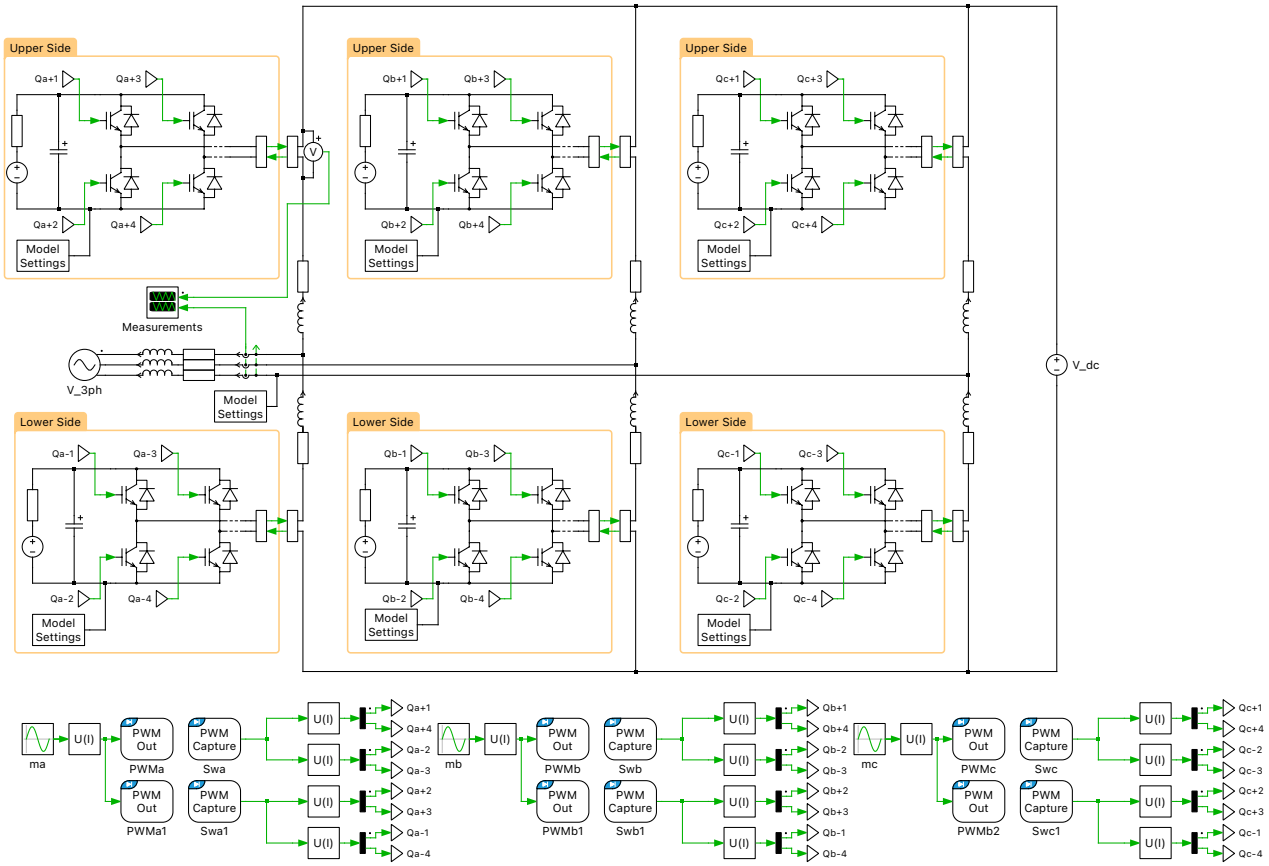


Fig. 2: Schematic of the grid-connected MMC inverter for CPU execution

The physical model is split as follows:

- Core 0: AC Grid and phase inductance/resistance and DC bus
- Core 1: All high-side sub-cells and DC-link capacitors
- Core 2: All low-side sub-cells and DC-link capacitors

To allow this system splitting, the physical model has to be split at strategic locations by using a coupling circuit. To do so, a controlled current source is placed in one part of the model and a controlled voltage source in the other part. Both sources are controlled by the respectively measured voltage/current state variables from the other part of the model. To avoid state/source dependence, one measured signal has to be delayed by one discretization step. This approach is shown in Fig. 4 on p. 6.

The individual CPU load for each core becomes smaller and due to this, the average execution time can be reduced considerably. A reduced discretization step size leads to a better frequency resolution, and therefore fidelity of the real-time simulation is improved.

FlexArray

The **FlexArray** subsystem is configured to be executed on the FlexArray solver available on the RT Box 2, 3 and 4. This configuration has the big advantage, as shown in Fig. 5 on p. 7, that the power circuit does not need to be split.

2.2 Controls

The demo model is operated in open-loop. The PWM generation is executed in the same box and routed to the digital output be means of the PWM Out block. Those PWM signals are then feed back to the dig-

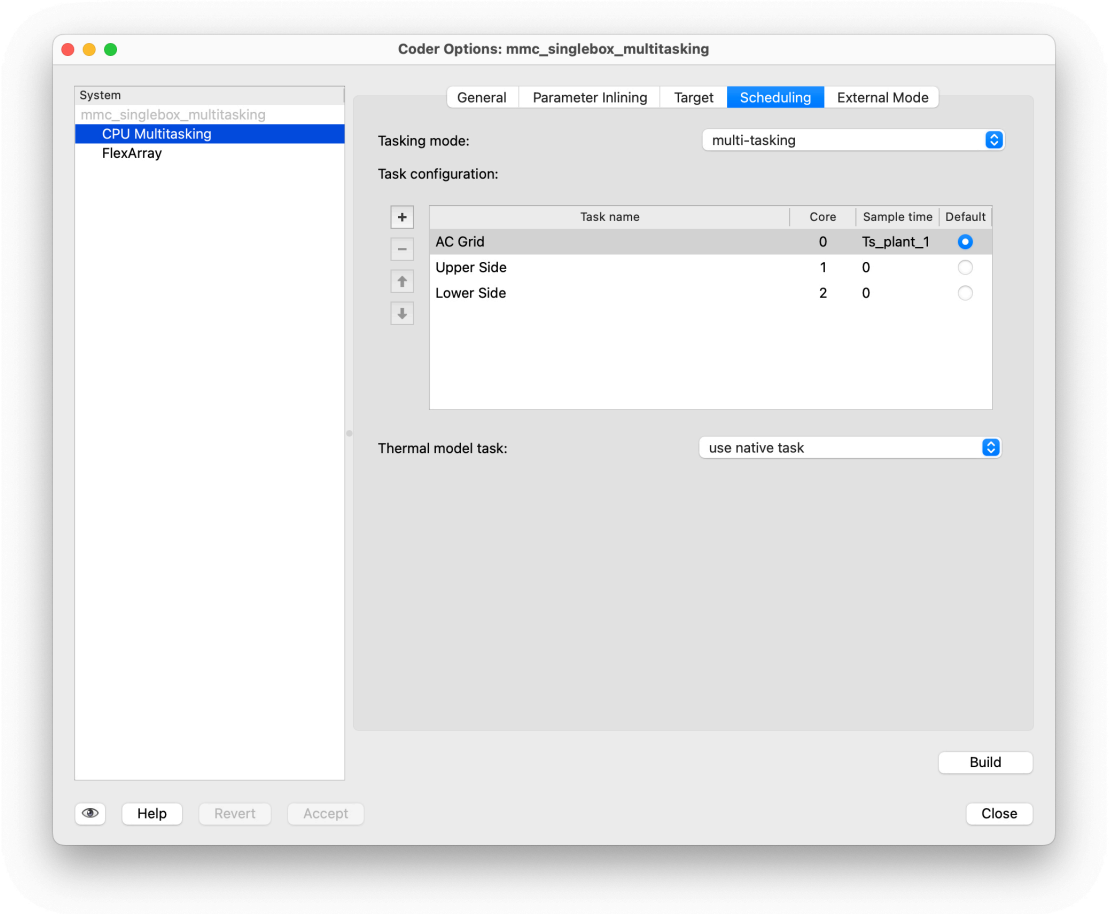


Fig. 3: Scheduling task in the Coder Options

ital inputs by using on or two physical loop-back cable(s). The PWM signals are then brought into the real-time simulation with the PWM Capture block.

3 Simulation

This model can run both in offline mode on a computer or in real-time mode on the PLECS RT Box.

Please follow the instructions below to run in real-time mode on an RT Box:

- Connect the Digital In interface to the Digital Out interface of the RT Box, i.e. by using a 37 pin Sub-D cable as shown in Fig. 6 on p. 7. When using an RT Box 3, two Sub-D cables must be connected.
- From the **System** tab of the **Coder options...** window, select one of the subsystems and go to the **Scheduling** tab. Chose either `single-tasking` or `multi-tasking` as the tasking mode and accept your choice. Please note that multi-tasking mode is only available for RT Box 2, 3 and 4.
- Return to the **System** tab, select one of the subsystems and **Build** it onto the RT Box.
- Once the model is uploaded, from the **External Mode** tab, **Connect** to the RT Box and **Activate autotriggering**.

During the real-time operation under **External Mode** the measurements can be observed using the PLECS Scope “Measurements”. The arm voltage and AC grid currents are shown in Fig. 7 on p. 8.

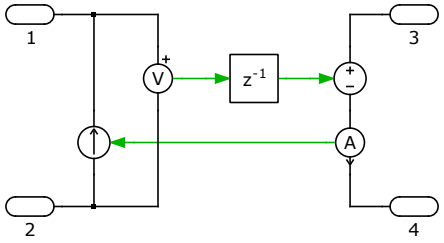


Fig. 4: Circuit to allow physical system splitting

4 Conclusion

This RT Box demo model demonstrates a grid-connected MMC inverter under open-loop control. The demo model can run both in single-tasking mode on one CPU core of the RT Box 1, 2, 3 or 4, or in multi-tasking mode on three CPU cores of the RT Box 2, 3 or 4. In addition, the MMC model can also be executed on the FlexArray solver on the RT Box 2, 3 or 4. Multi-tasking has the benefit that the average execution time can be reduced considerably. Execution on the FlexArray solver has the big advantage that the physical system does not need to be split.

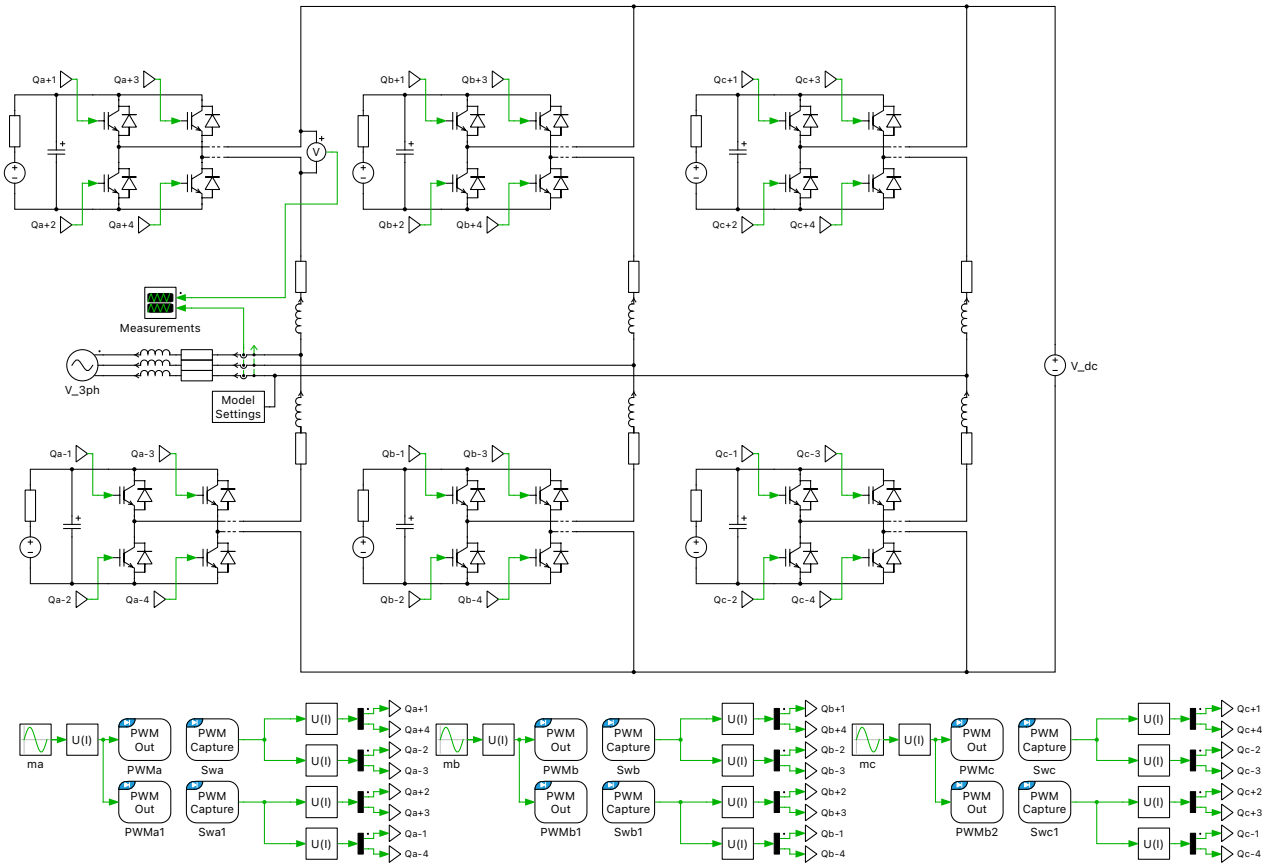


Fig. 5: Schematic of the grid-connected MMC inverter for execution on the FlexArray solver

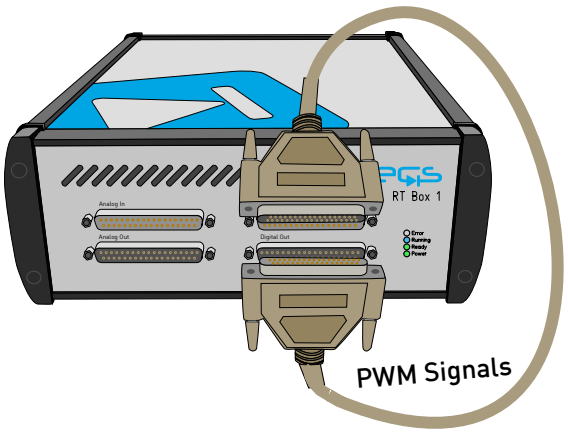


Fig. 6: Hardware configuration for the real-time operation of the demo model

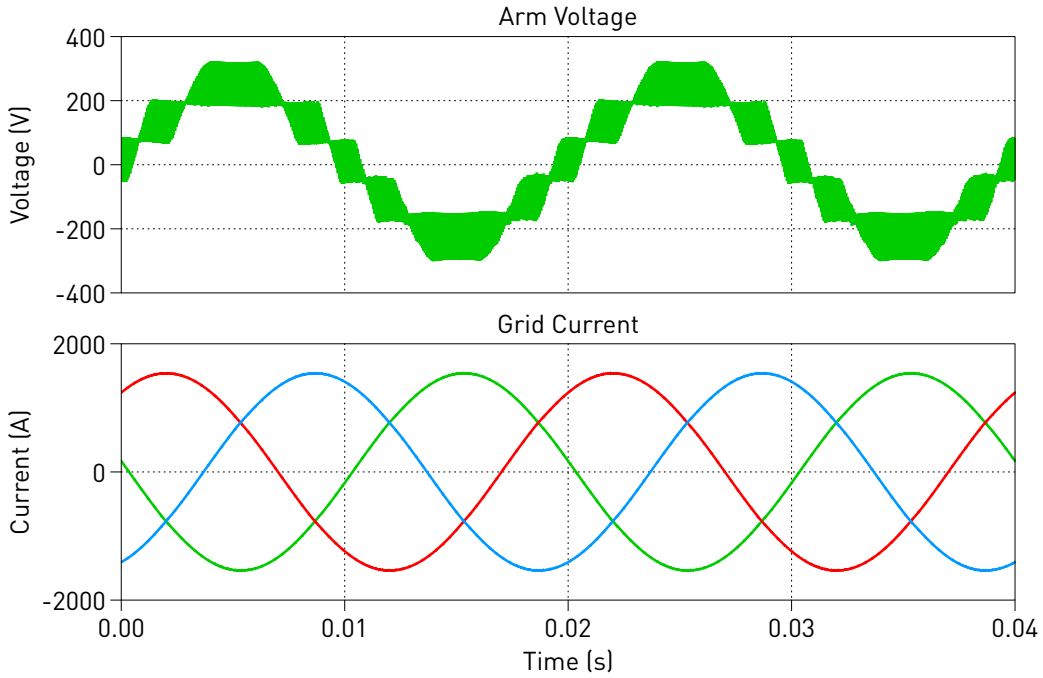


Fig. 7: Real-time measurements obtained on the RT Box in multi-tasking mode

Revision History:

RT Box TSP 2.0.5	First release
RT Box TSP 2.1.5	Turn on Assertions in IGBT Full Bridges and add deadtime in PWM Out blocks
RT Box TSP 2.2.1	Add grid impedance
RT Box TSP 4.0.1	Use FlexArray solver for RT Box 2,3 and 4

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
	`https://www.plexim.com`	Web

RT Box Demo Model

© 2002–2026 by Plexim GmbH

The software PLECS described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from Plexim GmbH.

PLECS is a registered trademark of Plexim GmbH. MATLAB, Simulink and Simulink Coder are registered trademarks of The MathWorks, Inc. Other product or brand names are trademarks or registered trademarks of their respective holders.