RT Box

Tutorial

Solver Engines

Showcasing three different solver engines on the RT Box

Tutorial Version 1.0

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 documentation

1 Introduction

This tutorial is based on the previous "RT Box Tutorial 2: Building a Simple VSI on the PLECS RT Box". In this tutorial you will continue to build on top of the previous VSI model increasing its complexity to a two-stage converter system, but still operating in open-loop.

You will learn step-by-step how to configure a model to use the three different solver engines on the RT Box, the associated time step capabilities and update times. The tutorial is oriented towards users who are already familiar with the classical CPU simulation on the RT Box, and now transition into using the new FlexArray solver and Nanostep® solver.

The tutorial is designed so that it can be completed *only* with **RT Box 2 or 3**. The RT Box 1 and CE are *not primarily considered* for this tutorial because they are *not* equipped with the FlexArray solver. If you only have an RT Box 1 or CE, you can only follow the part of "Your Task: *Optional*" in Section 2, Section 3.1 and 3.2 in this tutorial.

Before you begin You should complete the previous RT Box tutorials before continuing with this exercise. Make sure you have the right hardware setup: you will need a D-SUB 37-pin male-to-female loopback cable, an Analog Breakout Board, an oscilloscope and the necessary cabling. Please connect the loopback cable to drive the RT Box digital inputs from the digital outputs. Plug in the Analog Breakout Board to the analog inputs and outputs of the RT Box. A schematic overview of the hardware connections is given in Fig. 1.

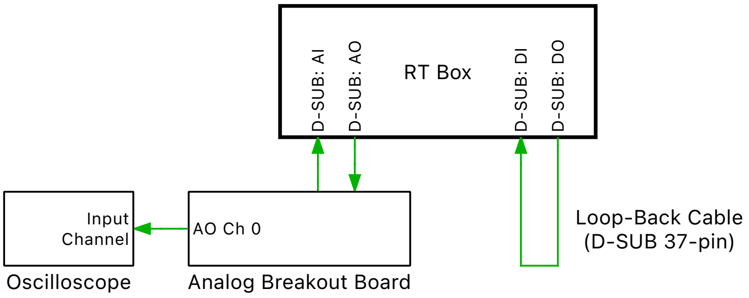

Figure 1: Hardware Connection Diagram

2 Model Preparation

Our starting point is the previously built open-loop VSI tutorial model. We will use it as the second stage of the converter system. In Section. 3 we will connect in front an open-loop Boost converter stage to form a two-stage converter system. Note that the parameters are for demonstrative purpose only.

Your Task:

Locate and open the reference model `solver_engines_prep.plecs`, or use your previously built model in the open-loop VSI tutorial. It should look like Fig. 2 and Fig. 3.

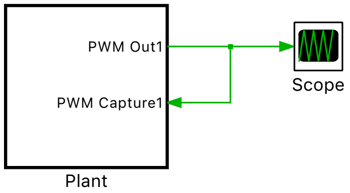

Figure 2: Top-level schematic of the model as the starting task

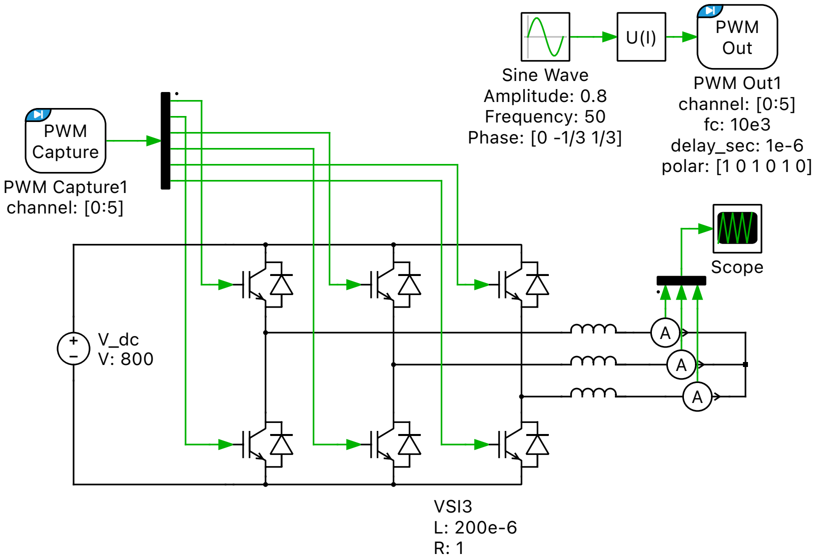

Figure 3: Schematic inside the "Plant" Subsystem of the model as the starting task

Your Task: *Optional* - only in case of an RT Box 1 or CE

For an RT Box 1 or CE, the inverter stage needs to be single-phase instead of three-phase. This is due to the fact that the RT Box 1 and CE have limited capability. In fact, they have only half of the Nanostep® solver capacity compared to the RT Box 2 and 3. Please refer to the RT Box Online Help, Chapter <Running Simulations on the FPGA>, Section "Nanostep Solver". Later in the tutorial, we will use the Nanostep® solver and need to prepare the simplified model for the RT Box 1 or CE at this stage.

Locate and open the reference model `solver_engines_prep_opt.plecs`. The schematic inside the "Plant" Subsystem should look like Fig. 4.

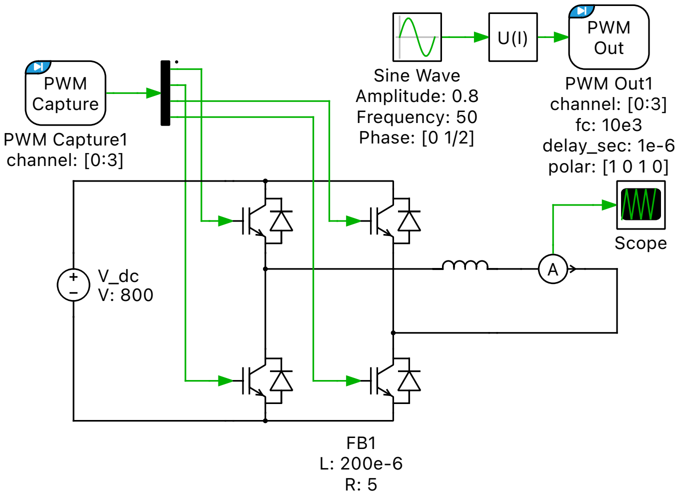

Figure 4: Schematic inside the "Plant" Subsystem of the model as the *optional* starting task

3 Utilizing Different Solver Engines

In this section, you are going to build a two-stage converter. Throughout each task, you will learn how to utilize the different combination of solver engines and their features.

3.1 Adding a Boost stage and using only the CPU Solver

Your Task:

1 Find the Boost Converter component also from the "Electrical", "Nanostep" section of the PLECS component library. Place it in front of the existing VSI stage. Set this component's configuration as **Sub-cycle average**, with the inductance value of $100\,\mu\mathrm{H}$ and the winding resistance of $0.1\,\Omega$.

2 Cut the VSI's DC Voltage Source and place it at the input of the Boost Converter. Set the voltage value from $800\,\mathrm{V}$ to $400\,\mathrm{V}$.

3 Place a Capacitor at the DC link. Set the Capacitance to $5\mathrm{e}{-3}\,\mathrm{F}$, and the initial voltage to $730\,\mathrm{V}$.

4 Next generate a PWM signal to drive the Boost Converter, in the same way as the VSI. Place an RT Box PWM Out block, set to use digital output channel 6 with carrier frequency of $50\,\mathrm{kHz}$. Change the carrier limit to [0 1]. Change the turn-on delay to $0\,\mathrm{s}$. Feed this PWM Out block with a Constant value of 0.5 as duty cycle.

5 Place a PWM Capture block, set to use digital input channel 6, keep the default **Offline behavior: Sub-cycle average**. Connect the PWM Capture block to the gate of the Boost converter switch. At the top-level schematic, properly connect the PWM signal path in order to feed the "PWM Out" to the "PWM Capture" in an open-loop fashion to drive the switch in the Boost Converter.

6 Place an Ammeter to measure the Boost inductor current. Connect an RT Box Analog Out block to output the Boost inductor current. Set to use analog output channel 0, with a **Scale** of 0.1 and **Offset** of -30.

7 Use a Scope to show three plots: Boost inductor current, DC link voltage, and load currents.

Note: By default, building onto the RT Box runs a CPU based simulation, unless the **Electrical Model Settings** block is included. For now, we are not using this block and we will add it in a later step when setting up the FPGA simulation.

8 Refer to Fig. 5 and Fig. 6 for the schematic at the current step. Run the simulation once in offline simulation, the scope waveforms should look like in Fig. 7.

9 Keep the discretization time step of $2.5\mathrm{e}{-6}\,\mathrm{s}$. Build the model onto your RT Box 2 or 3, enable external mode to view the real-time waveform inside the Scope.

? What is the processor loading in percentage? These performance metrics can be accessed from the RT Box Web Interface, or at the RT Box front panel display. Fill the values into the Table 1 at the end of this document. Use "n/a" when the table entry is not applicable.

A The average processor loading is around $90\,\%$, based on a step size of $2.5\,\mu\mathrm{s}$.

? Use the oscilloscope to probe the Analog Out channel 0 on the Analog Breakout Board, which outputs the Boost inductor current. What is the update rate of this physical signal? Fill this value into the Table 1 at the end of this document.

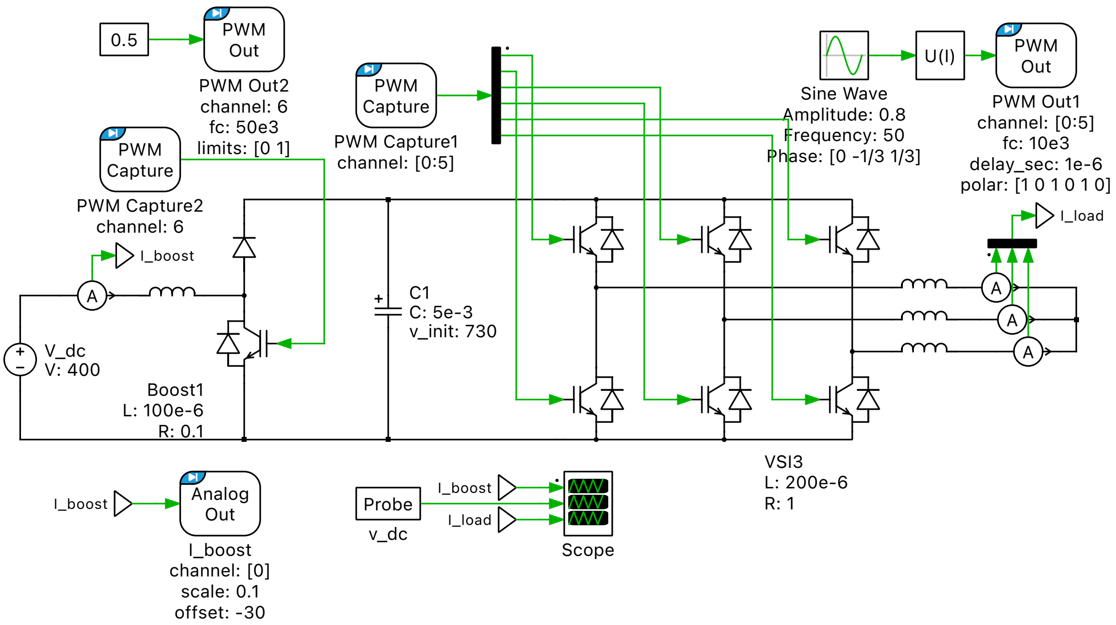

Figure 5: Schematic inside the "Plant" Subsystem for the cascaded Boost Converter and VSI, using the CPU solver

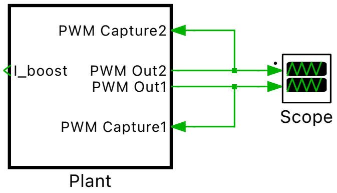

Figure 6: Top-level schematic of the cascaded Boost Converter and VSI, using the CPU solver

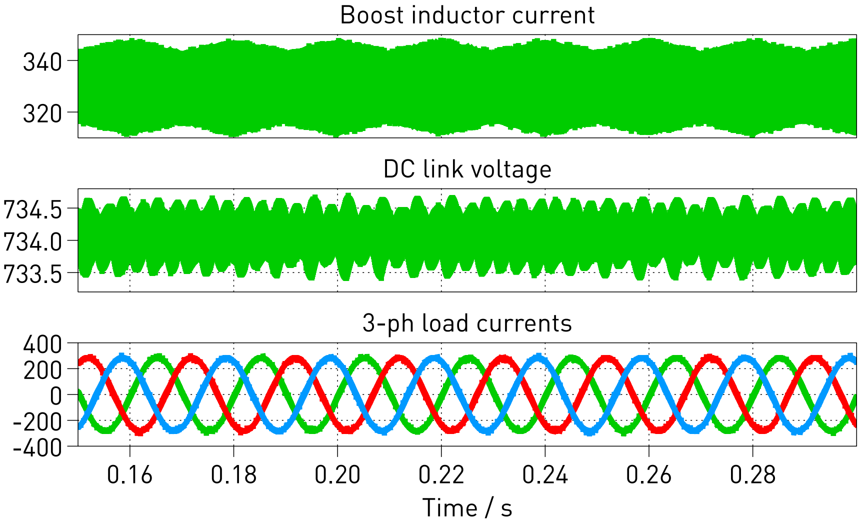

Figure 7: Offline simulation result of task 3.1

The signal updates at a step size of $2.5\,\mu s$, which is the same as the CPU discretization step size.

After completing this task, your model should be the same as the reference model `solver_engines_1.plecs`.

Your Task: *Optional* - only in case of an RT Box 1 or CE

1. Read through the standard task in this Section 3.1. Add the Boost Converter in front of Full-Bridge Inverter. The only difference is the winding resistance of the Boost Converter component, $0.5\,\Omega$ instead of $0.1\,\Omega$. The schematic inside the "Plant" Subsystem should look like the Fig. 8.

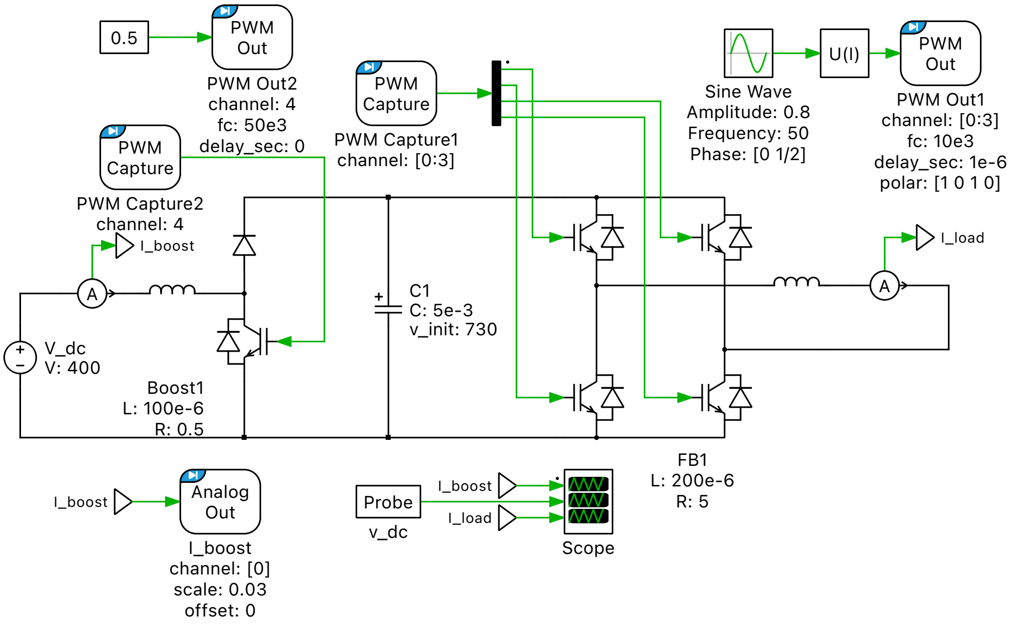

Figure 8: Schematic inside the "Plant" Subsystem in the *optional* task of Section 3.1

2. Adjust the Analog Out block to have a **Scale** of 0.03 and keep the **Offset** as 0.
3. Keep the discretization time step of $2e{-}6\,\mathrm{s}$. Build and activate the external mode.
4. Finish the Q&As in the standard task.

After completing this task, your model should be the same as the reference model `solver_engines_1_opt.plecs`.

3.2 Nanostep Solver with CPU Solver

So far we have been running the "Nanostep® Power Modules", VSI and Boost Converter, as sub-cycle averaged models on the CPU. In the next task we will run these modules as switched models on the Nanostep® solver. Users can keep the passive components, such as input voltage source and load, to be solved on the CPU, and only bring the Boost Converter and the VSI onto the Nanostep® solver. This is not the option that gives the least latency on an RT Box 2 and 3. However, for RT Box 1 or CE it is indeed the option that results in the fastest possible real-time simulations.

Your Task:

1 Change the configuration of both the Boost Converter and the 3-Phase VSI to **Nanostep**.

2 If you wish to run an offline simulation before building the model on to the RT Box, you need to change both PWM Capture blocks to the matching **Offline behavior: Nanostep**. Additionally you need to add **Pulse Delay Blocks** with $4\mathrm{e}{-9}\,\mathrm{s}$ delay on the inputs of the PWM capture blocks, to avoid an algebraic loop. The top-level schematic of your model is now as shown in Fig. 9.

3 Keep the discretization time step of $2.5\mathrm{e}{-6}\,\mathrm{s}$. Build the model onto your RT Box 2 or 3, enable external mode to view the real-time waveform inside the Scope.

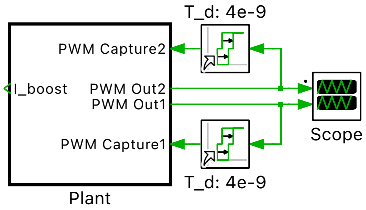

Figure 9: Top-level schematic in the task of Section 3.2

What about the RT Box performance now? Fill the values into the Table 1 at the end of this document.

The average processor loading is around $60\,\%$, based on a step size of $2.5\,\mu\mathrm{s}$. The two converter modules are calculated at $4\,\mathrm{ns}$ step with the two Nanostep® solvers. Meanwhile, the remaining part of the circuit is calculated on the CPU with a step size of $2.5\,\mu\mathrm{s}$.

Use the oscilloscope to probe the Analog Out channel 0 on the Analog Breakout Board. What is the update rate of this physical signal? Fill this value into the Table 1 at the end of this document.

The signal updates at a step size of $2.5\,\mu\mathrm{s}$ now, which is the same as the current CPU step size.

After completing this task, your model should be the same as the reference model `solver_engines_2.plecs`.

Your Task: *Optional* - only in case of an RT Box 1 or CE

1 Read through the standard task in this Section 3.2. Adjust to **Nanostep** configuration for the corresponding blocks. Add **Pulse Delay Blocks** with $7.5\mathrm{e}{-9}\,\mathrm{s}$ delay on the PWM signal path at the top-level schematic.

2 Keep the discretization time step of $2\mathrm{e}{-6}\,\mathrm{s}$. Build and activate the external mode.

3 Finish the Q&As in the standard task.

After completing this task, your model should be the same as the reference model `solver_engines_2_opt.plecs`.

Note: The next two sections involve the utilization of the RT Box **FlexArray Solver**, which is only available on RT Box 2 and 3. If you are using an RT Box 1 or CE, the tutorial ends here.

3.3 FlexArray Solver

Whether an electrical circuit is simulated on the CPU or the FPGA is controlled by the **Electrical Model Settings** block. In this task, we keep the same circuit model and connect an Electrical Model Settings block to the circuit in order to change from CPU simulation to the FlexArray FPGA simulation.

Your Task:

1 Connect and set the Electrical Model Settings block as **Target: FPGA**. Refer to Fig. 10 for the schematic of the current model.

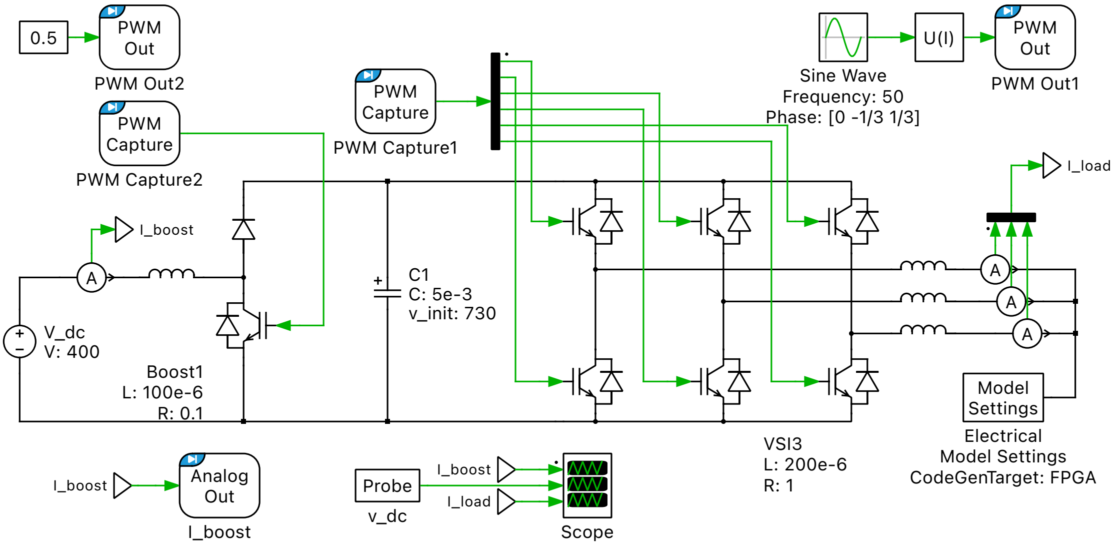

Figure 10: Schematic inside the "Plant" Subsystem for the cascaded Boost Converter and VSI, using the FlexArray Solver

2 Change the corresponding components (i.e. Boost Converter, 3-Phase VSI and two PWM Capture blocks) configuration from Nanostep back to **Sub-cycle average**, as in Section 3.1.

3 **Right-click** and **Comment through** the **Pulse Delay Blocks** with $4\mathrm{e}{-9}\,\mathrm{s}$ delay, at the top-level schematic.

4 Keep the discretization time step of $2.5\mathrm{e}{-6}\,\mathrm{s}$. Build the model onto your RT Box 2 or 3, enable external mode to view the real-time waveform inside the Scope.

? What about the RT Box performance now? Fill the values into the Table 1 at the end of this document.

A The average processor loading is at around $55\,\%$, based on a step size of $2.5\,\mu\mathrm{s}$. Meanwhile, the whole converter electrical circuit is simulated on the FPGA with a step size of $417\,\mathrm{ns}$.

? Use the oscilloscope to probe the Analog Out channel 0 on the Analog Breakout Board. What is the update rate of this physical signal? Fill this value into the Table 1 at the end of this document.

A The signal updates at a step size of $417\,\mathrm{ns}$, which is the same as the FlexArray FPGA step size. When a simulation is running using the FlexArray solver, the analog outputs of the RT Box are updated after each FPGA step.

After completing this stage, your model should be the same as the reference model `solver_engines_3.plecs`.

3.4 Nanostep Solver with FlexArray Solver

The Nanostep® solver can bring the simulation time step down to just $4\,\mathrm{ns}$ on an RT Box 2 or 3. The power modules supported by the Nanostep® solver can be found in the “Electrical”, “Nanostep” category of the PLECS components library. We are already using these components since the beginning of this tutorial. With the Nanostep® solver, only the time-critical part of a circuit is calculated with Nanostep® resolution, i.e., $4\,\mathrm{ns}$.

Note: For all the Nanostep® Power Modules in their default “Nanostep” configuration, there is internally an Electrical State-Space Connector when using the Nanostep® solver of the RT Box. This ensures that both the input and the output circuitry of the Nanostep® Power Module stay inside the same state-space system.

According to the target chosen by the connected **Electrical Model Settings** block, the remaining part of the system can be configured to calculate either on the CPU or the FlexArray solver.

In the previous Task 3.2, we already configured the rest of the circuit to run (by default) on the CPU by *not* connecting an Electrical Model Setting block. This gives the same behavior as connecting an Electrical Model Setting block and setting its target as CPU.

In this task, we keep the rest of the circuit (i.e., input DC source, input current Ammeter, DC link capacitor and three-phase load current Ammeters) to be executed on the FlexArray solver, which is faster than on the CPU.

Your Task:

1 Change the configuration of both the Boost Converter and the 3-Phase VSI to **Nanostep**.

2 If you wish to run an offline simulation before building the model on to the RT Box, you need to change both PWM Capture blocks to the matching **Offline behavior: Nanostep**.

3 **Right-click** and **uncheck** the Comment through option for the **Pulse Delay Blocks** with $4\mathrm{e}{-9}\,\mathrm{s}$ delay, at the top-level schematic.

4 Keep the Electrical Model Settings block as **Target: FPGA**.

5 Keep the discretization time step of $2.5\mathrm{e}{-6}\,\mathrm{s}$. Build the model onto your RT Box 2 or 3, enable external mode to view the real-time waveform inside the Scope.

What about the RT Box performance now? Fill the values into Table 1 at the end of this document.

The average processor loading is still at around $55\,\%$, based on a step size of $2.5\,\mu\mathrm{s}$. The two converter modules are calculated at $4\,\mathrm{ns}$ step with the Nanostep solvers. Meanwhile, the remaining part of the circuit is calculated with a reduced step size of $250\,\mathrm{ns}$ using the FlexArray solver in the FPGA.

Use the oscilloscope to probe the Analog Out channel 0 on the Analog Breakout Board. What is the update rate of this physical signal? Fill this value into the Table 1 at the end of this document.

The signal updates at a step size of $250\,\mathrm{ns}$ now, which is the same as the current FlexArray FPGA step size.

After completing this stage, your model should be the same as the reference model `solver_engines_4.plecs`.

Note: After clicking Build button, users can find more info about which Nanostep® module is calculated on which Nanostep® solver with what weight number, by opening the PLECS menu tab “Window” and then “Show Console”.

4 Conclusion

We recommend you take a moment to look at the filled Table 1 with values. Also, we have included a Fig. 11 to highlight the utilization of different solver engines (distinguished by colors) in each corresponding task of this tutorial.

Up till now, you have learnt how to utilize different solver engines and the corresponding step sizes. With an understanding of this topic, you should now be able to assign model calculation in different solvers in order to achieve the best performance on the RT Boxes.

Table 1: Time step comparison table for different solver configurations

Task	CPU Solver exec./disc. step	FlexArray Solver step	Nanostep® Solver step	AO-0 step
3.1				
3.2				
3.3				
3.4				

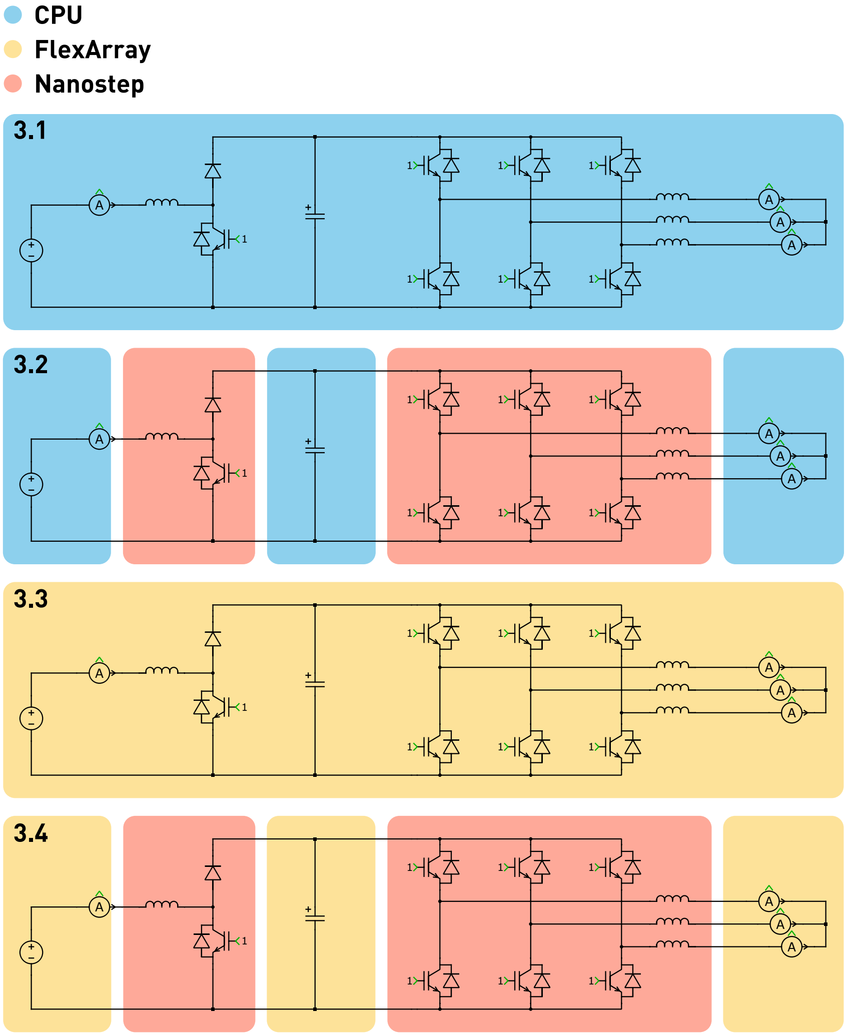

Figure 11: Utilization of different solver engines in each task of this tutorial

Revision History:

Tutorial Version 1.0	First release

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

RT Box Tutorial