RT Box

Tutorial

System Splitting for Distributed Real-Time Simulation

Learn how to split a system model to execute it on multiple RT Boxes

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

Large power electronic systems often include many switching devices so that the real-time deployment on a RT Box can be challenging. The main reasons are:

- Large systems will result in increased calculation times, and therefore, discretization step sizes.
- The real-time executable code might not fit into the available memory of the RT Box.
- The number of required Analog/Digital signals exceeds the physical number of available I/Os of the RT Box.

In this tutorial you will learn how to split a given PLECS model into two parts. The different parts of the model exchange information using Am- and Volt-meters and controlled voltage and current sources. The split of the system is implemented at a slowly varying state variable (i.e. the DC-Link). In the end different options for synchronization between RT Boxes using SFP ports are explained.

Before you begin This tutorial requires two RT Boxes and a PLECS + PLECS Coder license, as well as knowledge about basic RT Box operation and usage. Please check www.plexim.com for introductory tutorials and how to get started with the RT Box.

The tutorial is designed so that it can be completed with:

- 2 RT Boxes
- 2 D-SUB 37-pin male-to-female loopback cables
- 1 SFP cable

Make sure that the file `distributed_realtime_simulation_start.plecs` is located inside your working directory along with the reference solutions for each tutorial task.

2 System Splitting

In this exercise the system of interest is based on a three-phase diode rectifier together with an interleaved boost and voltage source inverter (VSI), as shown in Fig. 1. Note that for simplicity, both the boost and VSI stages are controlled in an open-loop manner.

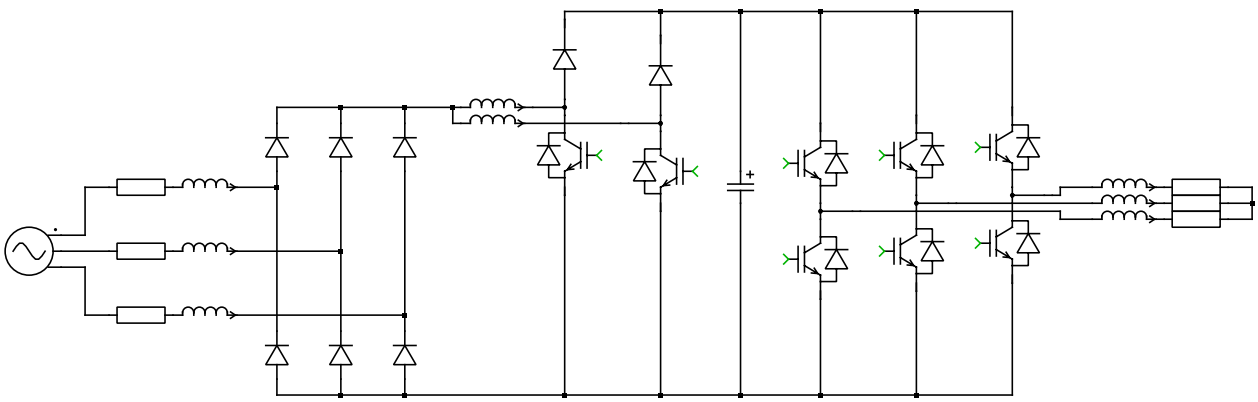

Figure 1: Rectifier and inverter circuit as target model for system splitting

Start from the provided PLECS model `distributed_realtime_simulation_start.plecs`, which is shown in Fig. 2. The circuit parameters and open-loop operating point are defined in the **Model initialization commands** field in the **Initialization** tab of the **Simulation Parameters** (**Ctrl + E**) window. Note that the initial currents of the line and boost inductors, as well as the initial DC capacitor voltage, are set such that the circuit reaches steady state faster. The boost legs and VSI half-bridge legs are all implemented using Power Module blocks in the PLECS component library. Further, they are configured to use the option, `Sub-cycle average`, instead of `Switched` in the **Configuration** field. Please refer to the RT Box tutorial “Model Optimizations” to get more information about this topic.

Your Task:

1. Before you run the simulation, ensure the simulation parameters of the PLECS solver are configured as follows:
   - Stop time: $0.5\,\mathrm{s}$
   - Max. step size: $1\mathrm{e}{-3}\,\mathrm{s}$
   - Relative tolerance: $1\mathrm{e}{-3}\,\mathrm{s}$
   - Solver type: `DOPRI (non-stiff)`
2. Run the simulation and observe the key waveforms in the Scope. They should look like in Fig. 3.

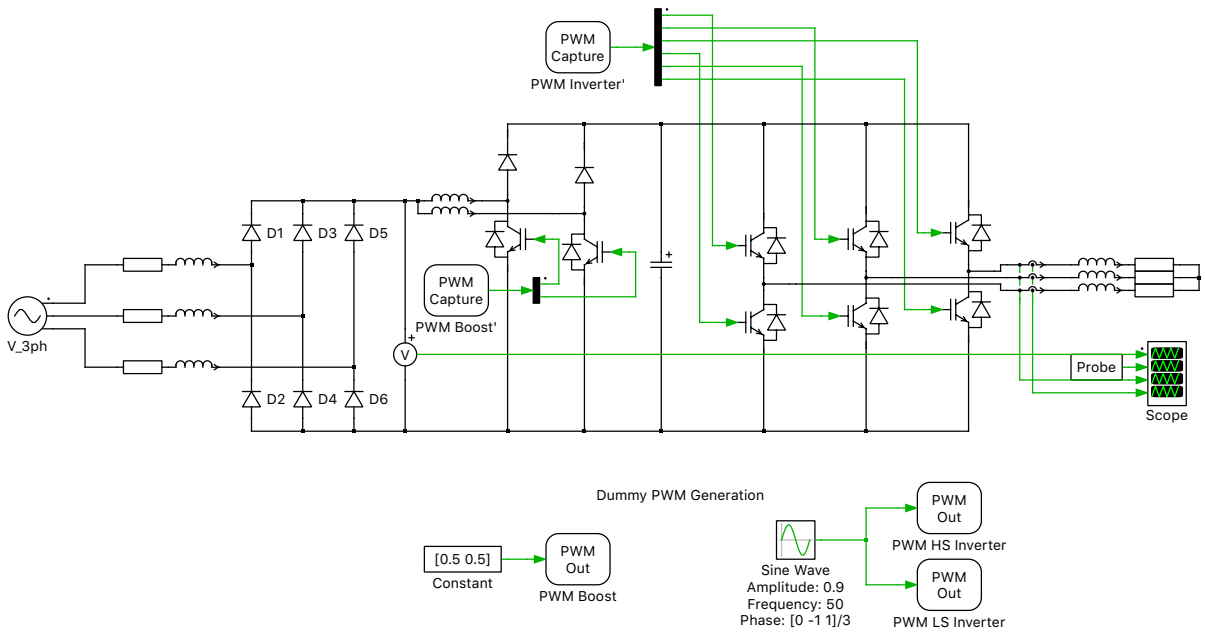

Figure 2: Rectifier and inverter system with open-loop PWM signal generation

2.1 Enable State-space Splitting and Deployment on One RT Box

In this task the system in Fig. 1 is not yet split into parts, but rather deployed as a whole on a single RT Box. Even though this is feasible, splitting the system will later allow decreasing the necessary discretization time step size.

Your Task:

1. In the **Options** tab of the **Simulation Parameters** window, when the **Enable state-space splitting** option is checked, PLECS will attempt to split the state-space model for a physical domain into smaller independent models that can be calculated and updated individually. This can reduce the calculation effort at runtime, which is particularly advantageous for code generation in real-time simulations. Next, by checking **Display state-space splitting**, PLECS will issue diagnostic messages that highlight the components that make up the individual state-space models after splitting.

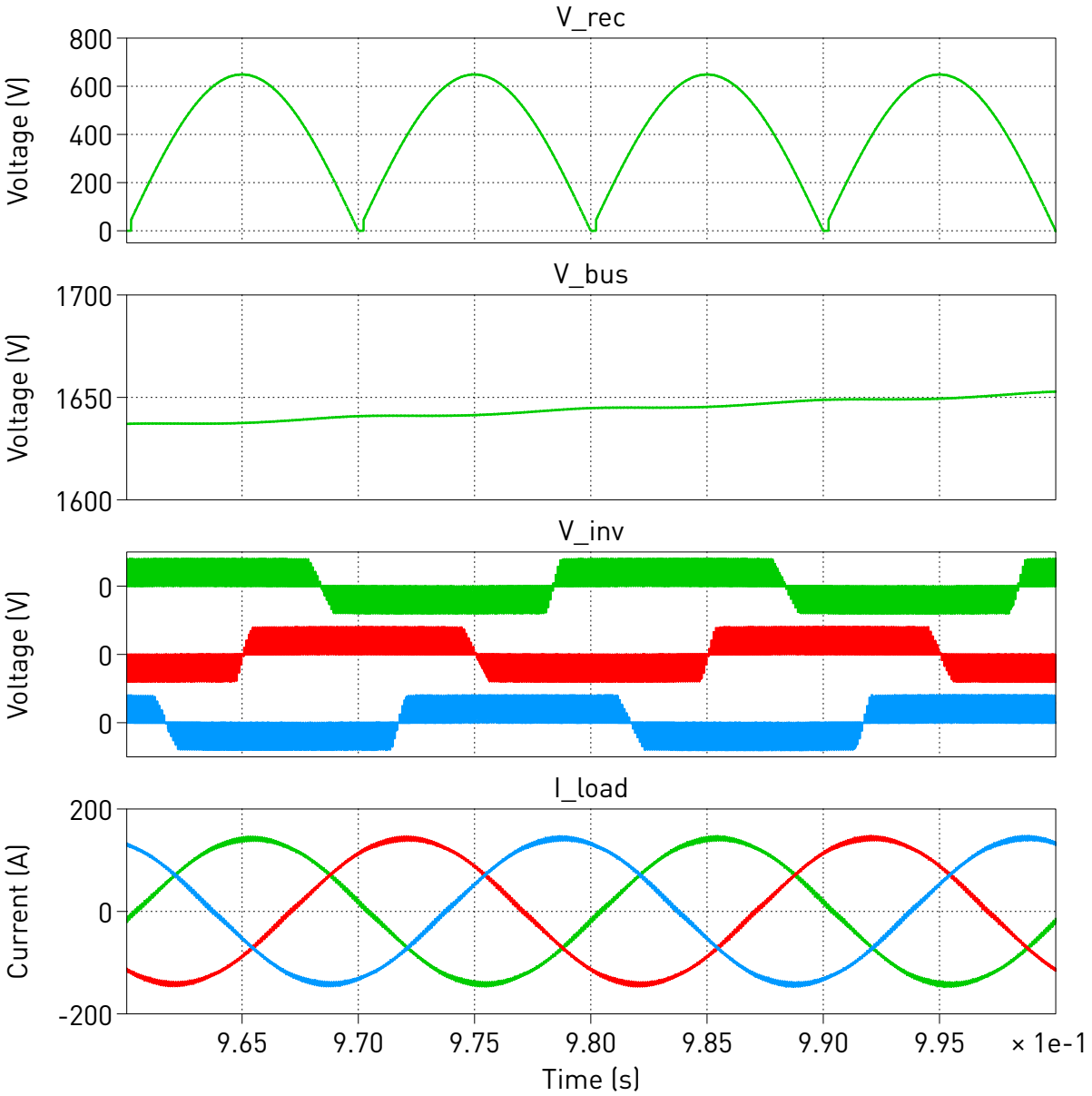

Figure 3: Simulation results of the rectifier and inverter system

Run the simulation again, click on the Diagnostics balloon at the right down corner of the schematic. What diagnostic messages do you see?

There are three Diagnostics messages describing three independent state-space models. Click on each message, one should see a list of components which compose each state-space model. Meanwhile these components are highlighted in the schematic for easy inspection. These three state-space systems are circled by different colors, and depicted in Fig. 4. Next, Fig. 5 shows the look-under-mask view of one Boost converter leg in `Sub-cycle average` configuration. And Fig. 6 shows the look-under-mask view of the Half Bridge inverter leg in `Sub-cycle average` configuration. It is clear that the DC-side circuit of the Half Bridge belongs to the middle state-space system together with the DC capacitor, and the DC-side circuit of the Boost converter leg. Whereas the Half Bridge inverter phase-side circuit belongs to the right state-space system together with the three-phase loads.

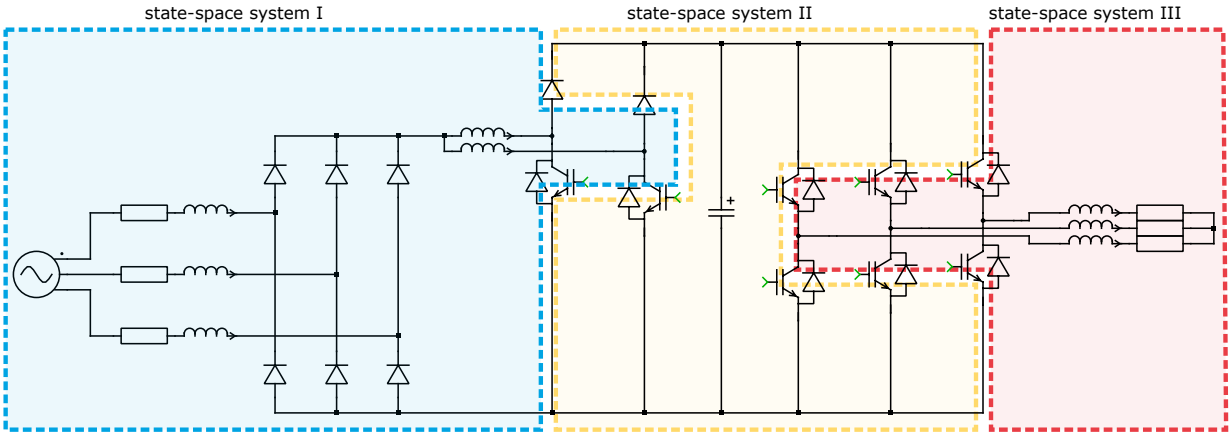

Figure 4: Three state-space systems generated by enabling state-space splitting function on the target system

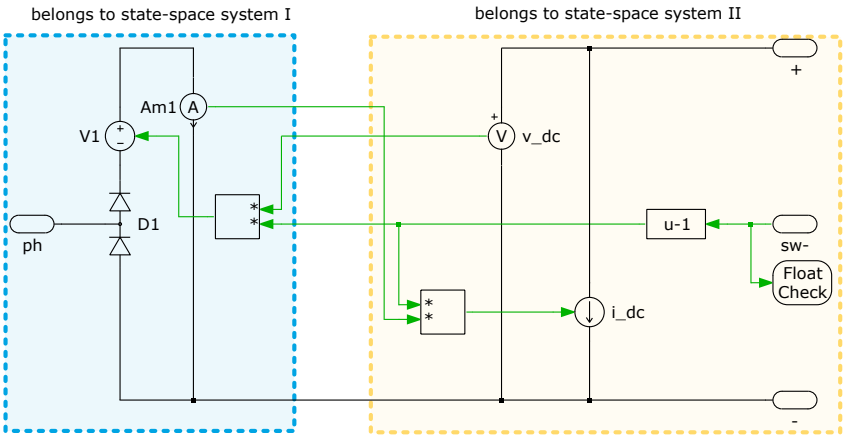

Figure 5: Look-under-mask view of the Boost leg in sub-cycle average configuration with state-space splitting function enabled

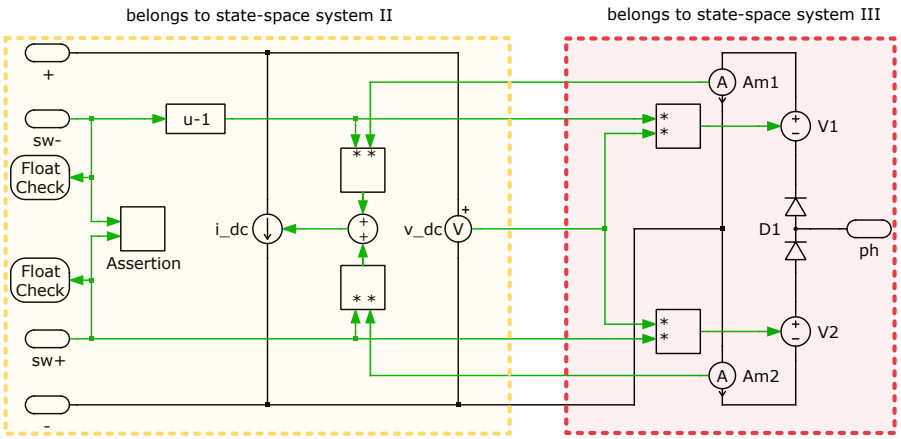

Figure 6: Look-under-mask view of the Half Bridge inverter leg in sub-cycle average configuration with state-space splitting function enabled

2 Enable code generation. Right-click on the subsystem, choose **Subsystem + Execution settings...**, then select the check box **Enable code generation** and set the **Discretization step size** as 4 μs.

3 Now we will build the model onto one RT Box. First connect the Digital In and Digital Out ports on the same Box with a D-SUB 37-pin male-to-female cable to allow the generated open-loop PWM signals to be sampled back in the Box. Then open the **Coder + Coder options...** window, click to choose on the left side under **System** window, the "Rectifier+Inverter" subsystem. Then choose your correct Box name under the **Target** tab, and click the **Build** button. Wait until the Box is running.

Once the Box is running, click **Connect** under **External Mode** tab, then click **Activate autotriggering** to observe the real-time waveforms in the Scope. What is the execution time for the Box?

Open the RT Box Web Interface by clicking the button, to find that the execution time is approximately 3.4 μs out of a 4 μs step size.

Note: A step size of 4 μs is actually adequate for this 10 kHz switching converter running on one Box. However, the next section shows how to further decrease the step size of two parallel Boxes by finding a proper node to split the system.

At this stage, your model should be the same as the reference file: `distributed_realtime_simulation_1.plecs`.

2.2 Model Splitting and Deployment on Two RT Boxes

For larger systems, a key optimization approach is to split the model into two or more parts. These parts are described by smaller independent state-space models that can be calculated and updated individually. This model splitting technique reduces the total number of switching combinations that need to be analyzed by the PLECS Coder and decreases the size of the state-space matrices. The following two points are important when splitting a system:

- The split needs to be implemented using a *state variable* to prevent the formation of an algebraic loop. In the electrical domain, this is done by measuring for example a capacitor voltage or inductor current.
- Moreover, the temporal behavoir of the chosen state variables should be slow, like in DC capacitors or AC line inductors. The model split adds a delay of one discretization time step (T_{disc}) to the interaction between the connecting model parts but the calculation of the slowly varying state variable is not affected by such a small delay. This approach also prevents a negative influence on the overall system stability.

Specifically in our model, the DC capacitor in the state-space system II links both the Boost leg and the Half Bridge inverter leg. Seen from state-space system II in Fig. 5 and state-space system II in Fig. 6, the DC-link capacitor has Controlled Current Sources in parallel on both left and right side.

How exactly do we split at the DC-link capacitor in our model?

The system is split at the DC capacitor, which is a slowly varying state variable. The DC link capacitor can either be put into the left (boost stage) or right (inverter stage) part of the model (here it is part of the inverter stage). Next, since the capacitor voltage is a state variable, it can be measured and transmitted to the left-side circuit where it models the load of Controlled Voltage Source from the two boost legs. These two currents are measured and transmitted to the right circuit. One can not measure the voltage in the boost stage and transmit it to a Controlled Voltage Source on the right side in parallel with the capacitor. This will cause a state-source dependency since between the capacitor and the voltage source.

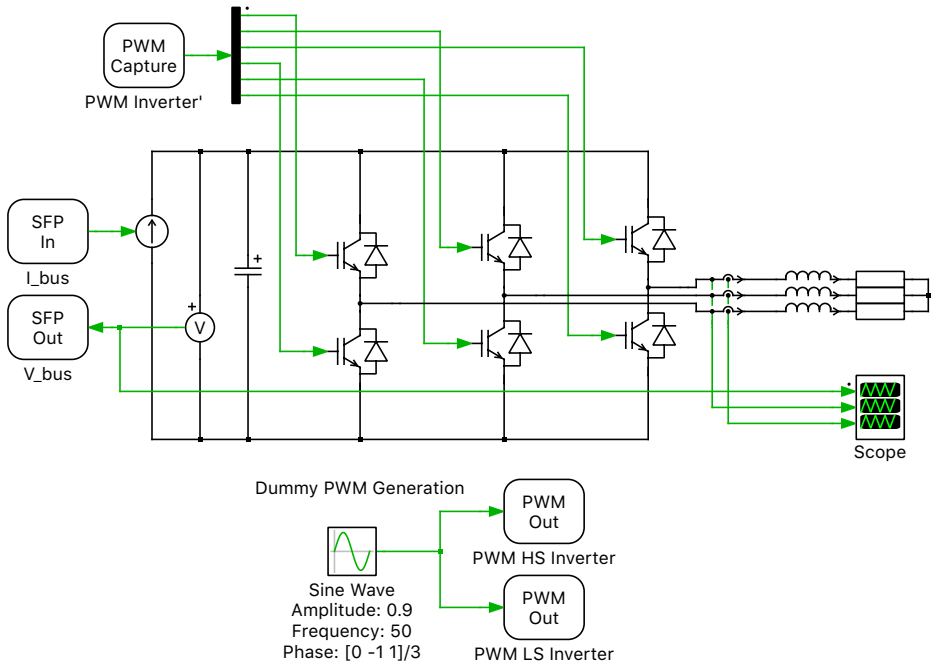

Figure 7: Inverter subsystem after splitting using SFP In/Out ports

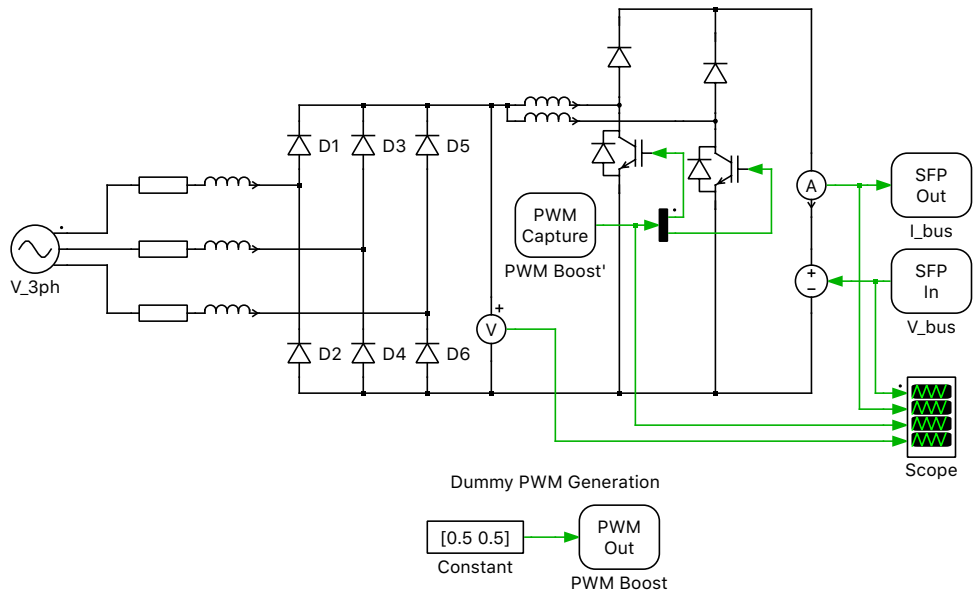

Figure 8: Rectifier subsystem after splitting using SFP In/Out ports

Your Task:

1. Drag a new Atomic Subsystem on the top-level schematic, name the existing one as “Inverter”, and the new one as “Rectifier”.

2. Inside the “Inverter” subsystem, keep the inverter-side circuit and its “dummy PWM Generation”, shown in Fig. 7. The DC-link capacitor is included in the inverter side. Since the capacitor voltage is a state variable, it is measured with a Voltmeter and sent out through an SFP Out block to the rectifier side. The current coming from the rectifier stage is introduced with SFP In block and

modeled using a Current Source (Controlled) block in parallel.

3. Cut the rectifier-side circuit with its "dummy PWM Generation" logic and paste it into the "Rectifier" subsystem, as shown in Fig. 8. The DC-link capacitor voltage is received through an SFP In block and is used to drive a Voltage Source (Controlled) block, which acts as the load of the rectifier stage. The load current is measured and sent out through an SFP Out block back to the inverter side.

4. At the top-level schematic, connect each subsystem's dummy PWM Out signal individually back to its corresponding PWM Capture port. Also, connect the SFP In/Out signals between the two subsystems to exchange the DC bus voltage and current measurements. Note that to freely move the subsystem terminals at the top-level schematic, you can drag along the subsystem block's border with the left mouse button while first holding down the Shift key. When you release the mouse button, the terminal is moved.

5. Run the PLECS simulation. Observe the Scope waveforms inside each subsystem. These should show similar results as the Scope in the unified system model in Fig. 3.

6. Enable code generation. Right-click on each subsystem, choose **Subsystem + Execution settings...**, then select the check box **Enable code generation** and set the **Discretization step size** as 2.5 μs.

At this stage, your model should be the same as the reference file: `distributed_realtime_simulation_2.plecs`, shown in Fig. 9.

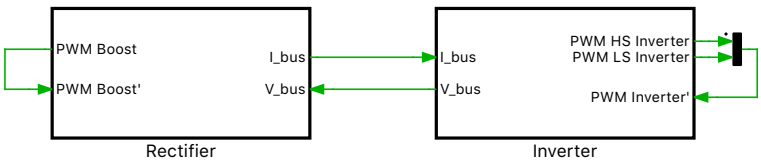

Figure 9: Top-level schematic of the system after splitting

Your Task:

1. Prepare the second RT Box. Make sure to connect the Digital In and Digital Out ports on each Box with a D-SUB 37-pin male-to-female cable to allow the generated open-loop PWM signals to be sampled back into the same Box. Also, use an SFP cable to connect both Boxes' rear side **Interconnect port A** together.

2. Now we will build the split model onto two RT Boxes. Open the **Coder + Coder Options...** window, click to choose on the left side under **System** window, the "Rectifier" subsystem. Then choose the correct Box for this subsystem, and click the **Build** button. Next do the same for the "Inverter" subsystem. Wait until both Boxes are up and running.

Once both Boxes are running, click **Connect** under the **External Mode** tab for each Box, then click **Activate autotriggering** to observe the real-time waveforms in the Scope. They should show similar results to the ones shown in Fig. 3. Now what is the execution time for each Box?

As seen in the RT Box Web Interface, the execution time is approximately 1.9 μs out of a 2.5 μs step size for the "Rectifier" side, and 2.1 μs out of a 2.5 μs step size for the "Inverter" side.

At this point you have successfully split a large system into two distributed systems, each running on a separate RT Box with a smaller time step than before, when the whole system was running on only one RT Box.

3 Synchronization of Split Systems

The previous section focused on splitting a system into different parts using controlled voltage and current sources. This section explains how the separated parts can be synchronized during start-up and execution on multiple RT Boxes.

A distributed real-time system running on multiple RT Boxes imposes certain difficulties:

- The I/Os of the different nodes need to be synchronized.
- The simulation startup needs to be synchronized to guarantee consistent model states.
- The latencies between different nodes in a distributed network need to be as small as possible.

In the following section, the synchronization features are explained based on the previous model `distributed_realtime_simulation_2.plecs`.

3.1 Synchronization of Multiple RT Boxes

Your Task: Based on the previous model, we will now enable the synchronization feature. We will designate the "Rectifier" Box as the synchronization master, and hence the "Inverter" Box as the slave.

1 Go to the **Coder + Coder options...** window, choose the "Inverter" subsystem, and go to the **Target** tab. Then go to **Interconnect** subtab to specify the synchronization function of the Box.

Note: By default, the **Master for startup/clock** combo box is chosen as `Self`. This means that each Box has its own startup once the build process completes, and also independent clocking for model calculation. Now since both Boxes use the same 2.5 μs step size, we can synchronize their timing.

2 For the slave "Inverter" Box, change the **Master for startup/clock** combo box to the `SFP A` option, then check the option **Use clock from master**. Proceed to build each model on each Box, enable external mode on both Boxes, and observe the waveforms again in the Scopes. They should show similar results to the ones shown in Fig. 3.

At this stage, your model should be the same as the reference file: `distributed_realtime_simulation_3.plecs`.

The **Use clock from master** option means that for time step synchronization the clock signal from a master Box is distributed to any assigned slave Boxes.

3.2 Startup Synchronization

Your Task: To guarantee a consistent model state, we want to achieve a synchronized deterministic startup moment.

1. Go to the **Coder + Coder options...** window, choose the "Inverter" subsystem, and go to the **Target** tab. You can stop a Box via the web interface. Open each Box's web interface, scroll down to the bottom, and click the **Stop** button. You will see that the blue "Running" LED on the Box's front panel turns off. Make sure you stop both Boxes before proceeding.
2. For the slave "Inverter" Box, in the **Interconnect** subtab, in addition to the previous selections, also check the option **Synchronize startup with SFP A**.
3. Proceed to build the model onto the "Inverter" Box. Once the build process has finished, the blue "Running" LED on the Box's front panel starts flashing. This suggests that it is waiting for the master Box to start up.
4. Now select the "Rectifier" subsystem, go to the **Interconnect** subtab, and also check the option **Synchronize startup with SFP A**. Proceed to build the model onto the "Rectifier" Box. Once the build process is finished, both Boxes' blue LEDs will light up normally. This indicates that both Boxes are now running, and their startup moment has been synchronized with the start of a new step size cycle.
5. Enable external mode on both Boxes, and again check the running waveforms inside the Scopes. They should show similar results to the ones shown in Fig. 3.

At this stage, your model should be the same as the reference file: `distributed_realtime_simulation_4.plecs`.

The **Synchronize startup with SFP A** option means that for startup synchronization the master first waits for all registered slaves to become available (and optionally synchronized with the master clock). Then the master sends a start signal to all slaves simultaneously and also starts its own simulation model.

Note: It is important to know that each SFP In/Out block introduces one discretization step size delay into the communication of data between Boxes. Therefore when two Boxes are running at the same simulation step, with the **Synchronize startup with SFP A** and **Use clock from master** options selected, a signal transmission latency of two discretization step is present. For offline simulations, these delays are modeled internally in the SFP In/Out blocks, hence no external delay block is necessary.

4 Conclusion

In this tutorial you have learned how to split a large system into different parts and execute them on different RT Boxes. Further, the concepts of model step synchronization and startup synchronization between multiple Boxes are explained.

These features are useful when splitting a virtual plant on multiple Boxes, or splitting a controller for I/O extension purposes.

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