RT Box

Tutorial

RT Box Timing and Step-Size Selection

Last updated in tutorials release 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

In this tutorial you will learn step-by-step how to build and optimize a basic power converter on the PLECS RT Box and select an appropriate simulation step size. The tutorial is designed for users making the transition from a PLECS model to a hardware-in-the-loop (HIL) simulation on the RT Box using PLECS Standalone.

While basic PLECS circuit models can run on the RT Box, there are optimizations that improve the simulation accuracy and reduce computational load of real-time simulations. One key real-time optimization is to use the PWM Capture component and hybrid power modules [1]. These two components unlock the high sample rate of the FPGA, such that the accuracy of the sensed duty cycle is no longer strictly determined by the model time step. This is critical in selecting an appropriate simulation step size.

The specific learning goals for the tutorial are:

- Establish the importance of sub-cycle averaging and hybrid power modules for real-time simulation.
- Understand how to optimize simple power converter models for real-time applications.
- Learn the process to select an appropriate model step size using the CodeGen mode.

Before you begin You should complete the previous RT Box tutorials before continuing with this exercise.

The tutorial is designed so that it can be completed with only an RT Box. There is one optional exercise that requires a loopback cable. The loopback cable is used to drive the RT Box analog and digital inputs from the RT Box outputs. A D-SUB 37-pin male-to-female cable can be used for this purpose.

2 From Offline to Real-Time: A Buck Converter Model

2.1 Sampled Switching Signals in a Buck Converter

If the switching signal, $s(t)$, is used to control the power semiconductor is sampled at a regular interval with a time step of T_{disc}, then the duty cycle of the sampled signal will have an error, ε. Figure 1 shows a continuous switching waveform with a duty ratio D and the corresponding sampled switching signal $\widetilde{D}$ at time steps t_i. If T_{sw} is the period of the switching signal and if we assume one on-transition and one off-transition per period, the upper limit for the absolute error of $\widetilde{D}$ is given by:

$$|\widetilde{D} - D| = |\varepsilon| < \frac{T_{\mathrm{disc}}}{T_{\mathrm{sw}}}$$

For an ideal buck converter operating in the continuous conduction mode (CCM) the equation below describes the relationship between the input and output voltages. In a fixed time step simulation, the error of the sampled duty ratio (ε) will directly translate to an error in the calculated state variables.

$$\frac{V_{\mathrm{out}}}{V_{\mathrm{in}}} = \widetilde{D} = D + \varepsilon$$

The error increases for larger simulation time steps (T_{disc}) and for shorter switching cycles (T_{sw}).

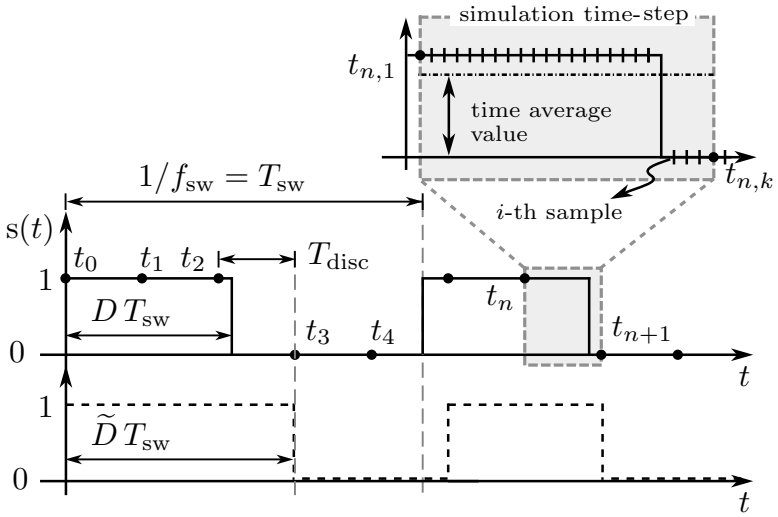

Figure 1: Comparison between original switching signal (top) and sampled signal (bottom). An expanded view displaying over-sampled data points is shown for one simulation time step.

2.2 A Buck Converter Using Digital Inputs and Outputs

In this exercise you will take a simple open-loop buck converter model and modify it for real-time execution. However, you will not perform any additional model optimizations at this point.

Prior to executing a real-time simulation you will use the CodeGen mode. The CodeGen mode utilizes the PLECS Coder to generate generic C code that executes on your host PC in PLECS. The generic C code represents the discretized model that can be benchmarked against a continuous state-space model. The CodeGen mode is useful to ensure the selected discretization step size is suitable for your model. In this exercise the CodeGen simulation will show there is an error in the effective duty cycle applied to the switch, leading to an error in the inductor current and output voltage. The duty cycle error is a result of the digital input updating only once per model step.

You will estimate the discretization step size required to reduce the error to an acceptable level.

Your Task:

1 Open the PLECS model `step_size_selection_start.plecs` and run the simulation. Notice that the MOSFET and Diode components are from the Power Semiconductors section of the PLECS Library. This model represents a conventional PLECS simulation that we will optimize for real-time execution. The "Plant" subsystem represents the circuit that you will eventually deploy to the RT Box.

2 The next step is to add RT Box specific components from the library browser to configure the I/Os connected to the RT Box. Drag and drop a Digital In block into the "Plant" subsystem and replace the Gate Signal In component. Add a Digital Output to replace the Gate Signal Out port. The Gate Signal Out will be used for monitoring the discretized PWM signal. Assign the inductor current and output voltage measurements to the RT Box analog outputs, replacing the corresponding Signal Outputs. Reconnect the signals at the top-level schematic.

Refer to Figure 2 to see the structure of the "Plant" subsystem and Figure 3 for the top-level schematic. Leave all component parameters at their default values except for those identified in the schematic.

3 Run the simulation and save all scope traces. Label the traces as "Offline".

4 Next, enable the subsystem for code generation. Right-click on the subsystem and navigate to **Subsystem + Execution settings...** and click the **Enable code generation** checkbox. The border of

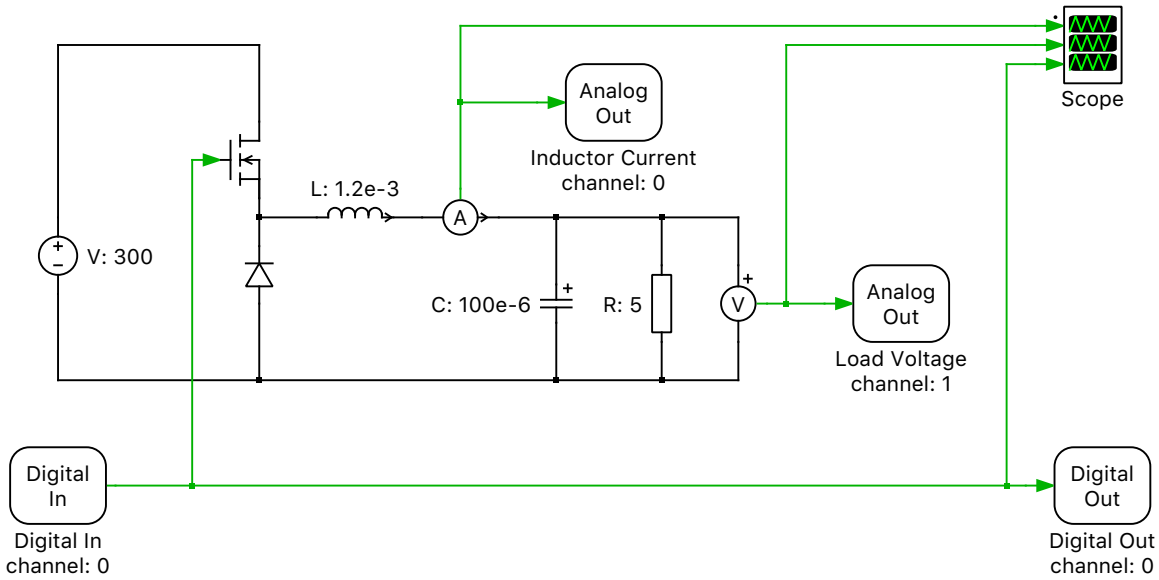

Figure 2: Buck converter model contained within the "Plant" subsystem

the "Plant" subsystem is now bold. From the **Coder Options** window navigate to the "Plant" subsystem in the left-hand menu. In the **General** tab enter discretization time constant for the system so there are 10 discrete samples per switching period. With a 10 kHz switching frequency the discretization time step should be $10e{-}6\,s$.

In the **Target** tab select the Generic target from the drop-down menu. Click the **Build** button. In the directory where you saved your model, a new folder will be created that contains the C code representing the discretized plant model.

5 Right-click on the Plant subsystem and navigate to the **Subsystem + Simulation mode** option and change the simulation mode from **Normal** to **CodeGen**. You should notice the border of the Plant subsystem is now a dashed bold line, as shown in Figure 3.

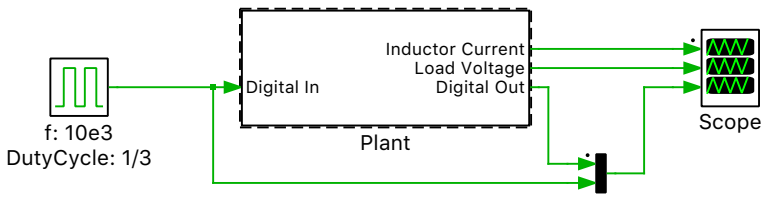

Figure 3: The top-level schematic with the subsystem simulation mode set to CodeGen

6 Run the model and save the scope trace as "CodeGen_Ts/10". Compare the scope results in your top-level schematic and notice any changes in the model output.

? Based on the output voltage, how has the effective duty ratio changed?

A The average output voltage for the CodeGen results is 120 V implying that the effective duty ratio is 0.4. This differs from assigned 0.333 duty cycle of the input signal.

7 Rerun the CodeGen simulation with the following discretization times and save the traces: $20e{-}6\,s$, $5e{-}6\,s$, and $1e{-}6\,s$. See Figure 4 for a comparison between the steady state voltages and currents between the offline simulation and the "CodeGen_Ts/10" results.

What is the required discretization time to achieve a 1% error in the average output voltage?

The discretization step size must be less than $1\mathrm{e}{-6}\,\mathrm{s}$ for this converter. This is based on the formula $|\widetilde{D} - D| = |\varepsilon| < \frac{T_{\mathrm{disc}}}{T_{\mathrm{sw}}}$ from Section 2.1 with $|\varepsilon| = 0.01$ and $T_{\mathrm{sw}} = 100\mathrm{e}{-6}\,\mathrm{s}$.

At this stage, your model should be the same as the reference model `step_size_selection_1.plecs`.

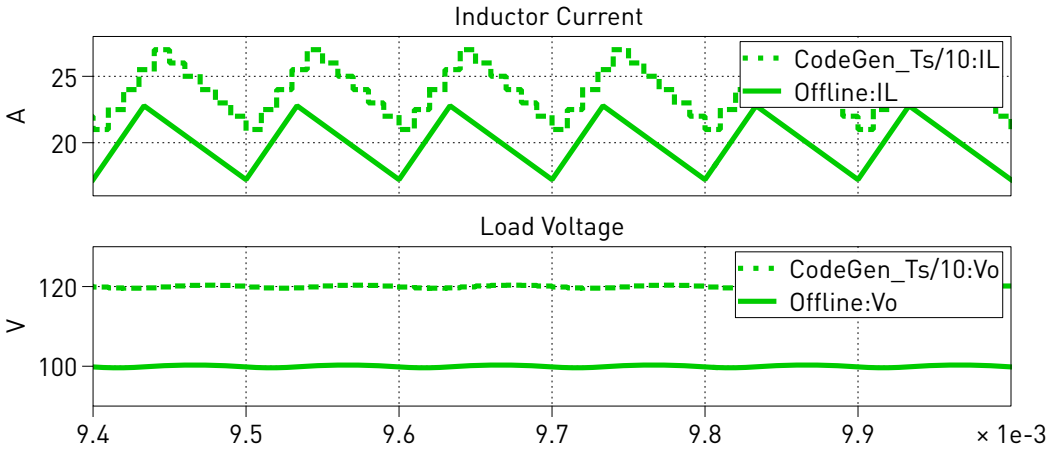

Figure 4: Current and voltage waveforms for the offline and CodeGen simulations using digital inputs

2.3 Hybrid Power Modules for Sub-Cycle Averaging

One approach to overcome the limited resolution of sampled switching signals is to determine the time-averaged value of the switching signal over each simulation time step. This requires over-sampling the the switching signal at at regular intervals much smaller than the simulation time step (T_{disc}). Figure 1 shows an over-sampled switching signal. At the end of each simulation time step a time-averaged value over the T_{disc} interval is determined.

In the context of the RT Box, the PWM Capture component is used to implement the above time-averaging approach. The PWM Capture component configures the RT Box to over-sample the specified digital input pins at the FPGA sample rate of 7.5 ns. All samples over a model time step are then averaged to determine the signal on time duration as a value between 0 and 1. The PWM Capture output contains additional duty cycle information compared to the Digital In component.

In order utilize the additional duty cycle information contained in a PWM Capture signal, specialized hybrid power modules are required. The time-averaged value of the PWM input of the previous model step is applied as the effective duty cycle of the hybrid power module.

With an appropriately selected discretization time, hybrid power modules still represent the dominant characteristics of the switching ripple, can model the converter in continuous and discontinuous conduction modes, and can account for blanking times. Hybrid power modules also have the added benefit of reducing the number of ideal switches and diodes contained within the model, reducing the number of state-space matrices required to represent a power converter [1]. The advantages of this approach can be extended to several different types of multi-level converters [2]. The key limitations of the hybrid power modules is that there is no short circuit state and no information about the switching sequence within a time step.

2.4 A Buck Converter Using PWM Capture and PWM Out

As highlighted in the previous exercise, PWM measurements using the Digital In block requires sufficient oversampling to accurately detect the duty cycle input. The PWM Capture block improves the measurement accuracy and reduces the oversampling requirements. Beginning with the PLECS model from the previous exercise, you will modify the model to use "hybrid power modules" and the PWM Capture component. You will use the CodeGen mode to determine the impact of model discretization on the simulation results.

Your Task:

1 Starting with the model from the previous exercise, drag and drop an IGBT Chopper (High-Side Switch) power module into the "Plant" subsystem. Replace the switch and diode with the power module.

Look under the mask of the power module. The power module is implemented as a configurable subsystem with two configurations. The Switched configuration represents the chopper circuit as an IGBT and diode. The Sub-cycle average configuration represents the electrical equation of the chopper circuit with controlled sources and a diode that can be either blocking or conducting during each simulation time step [1]. Set the configuration parameter to Sub-cycle average.

2 Delete the Digital In component and replace it with a PWM Capture block from the RT Box Target Support library. Connect the output of the PWM Capture component to the switch gate signal. Confirm that the Averaging interval (offline only) parameter of the PWM Capture block matches the discretization time of $10e{-}6\,s$. The output signal of the PWM Capture block should also be connected to the Scope and the Digital Out component for plotting. Refer to Figure 5.

From the top-level schematic, connect the Pulse Generator output to the PWM Capture input. Then change the "Plant" subsystem simulation mode to Normal, run the model, and save the scope trace as "Offline". Zoom in on the PWM signals and notice the output of the PWM Capture block is now a value between 0 and 1.

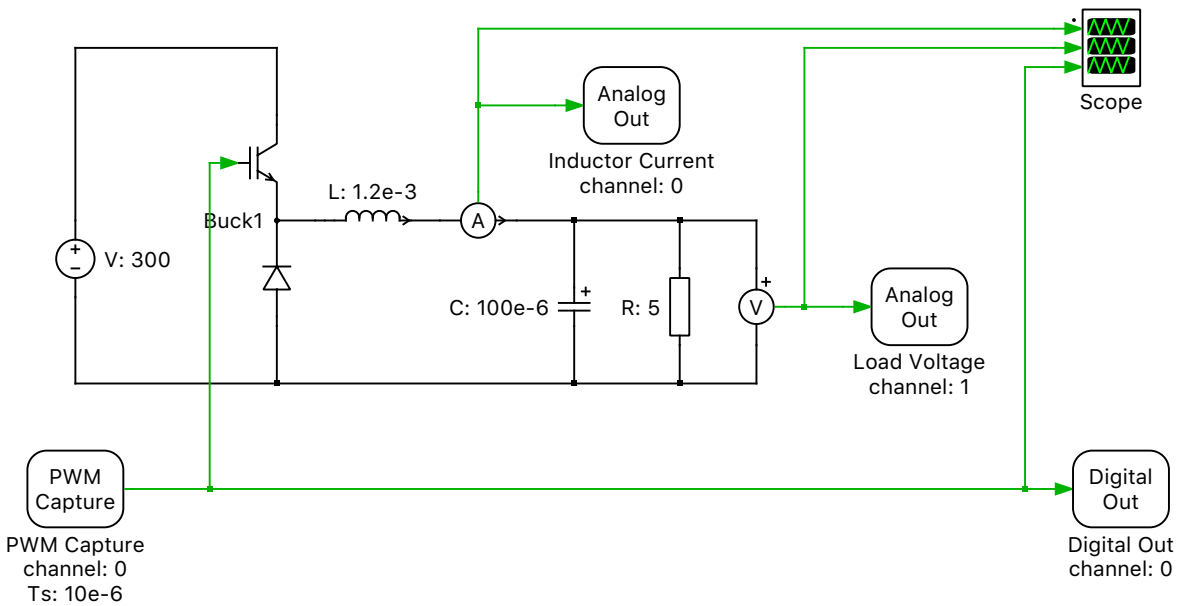

Figure 5: The "Plant" subsystem with PWM Capture and a hybrid power module

3 Change the "Plant" simulation mode from Normal to CodeGen and use the same $10e{-}6\,s$ discretization time constant. Generate code for the "Plant" and then run the discretized model.

Notice the inductor current and output voltage closely resemble the reference solution for the exact same discretization time in the previous exercise without the hybrid switch models and PWM Capture, as shown in Figure 6. Save your scope traces as “CodeGen_Ts/10”. Compare the voltage and current waveforms for the different discretization step sizes.

What is the required discretization time step to achieve less than a 1% error in the average output voltage with this approach?

The discretization time step is largely decoupled from the PWM sensing and average output error of this converter due to the sub-cycle averaging of the PWM input.

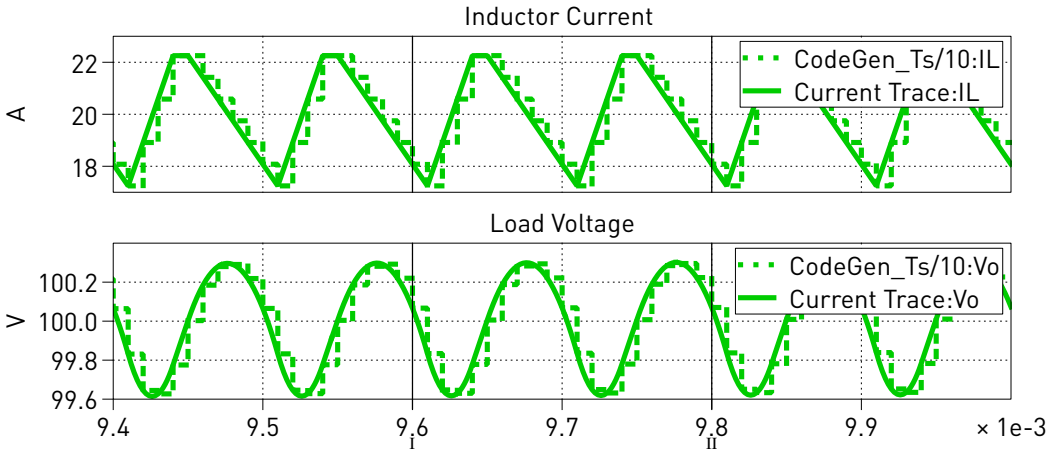

Figure 6: Current and voltage waveforms for the offline and discretized models using a PWM Capture block and a hybrid switch model

At this stage your model should be the same as the reference model `step_size_selection_2.plecs`.

Your Task: *(Optional Exercise!)* Connect a loopback cable from the RT Box digital outputs to the digital inputs. Remove the Digital Output block and replace it with a PWM Out component. Change the PWM Out block to use Digital output channel 0, a carrier frequency of 10 kHz, carrier limits of [0,1], and with zero Turn-on delay.
Connect a constant modulation index of $1/3$ to the PWM Out component. Add the modulation index to the Parameter Inlining exceptions list. Ensure the PWM Capture result is still connected to your Scope. Run your model on the RT Box and use the External Mode to view the “Plant” Scope. Compare the offline and real-time results. Change the modulation index and observe the change in inductor current and output voltage.

At this stage your model should be the same as the reference model `step_size_selection_2_optional.plecs`.

3 Conclusion

You now have the tools to determine if the model step size you selected is appropriate by using the CodeGen mode, even before deploying your model to an RT Box. The exercises showed the impact that PWM Capture and hybrid power modules have when selecting a step size. By structuring a model

around a subsystem that represents the real-time portion of the model, you can directly compare offline, real-time, and CodeGen simulation results from within a single PLECS model.

With an understanding of these topics, you should now be able to develop your own real-time simulation models for both HIL and RCP applications.

References

[1] Jost Allmeling, Niklaus Felderer “Sub-cycle average models with integrated diodes for real-time simulation of power converters”, 2017 IEEE Southern Power Electronics Conference (SPEC), pp. 1-6, 2017.

[2] Jost Allmeling, Niklaus Felderer, Min Luo, “High Fidelity Real-Time Simulation of Multi-Level Converters”, Power Electronics Conference (IPEC-Niigata 2018 –ECCE Asia) 2018 International, pp. 2199-2203, 2018.

Revision History:

Tutorials 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