



PLECS

*DEMO MODEL*

## Buck Converter with Cascaded Controls

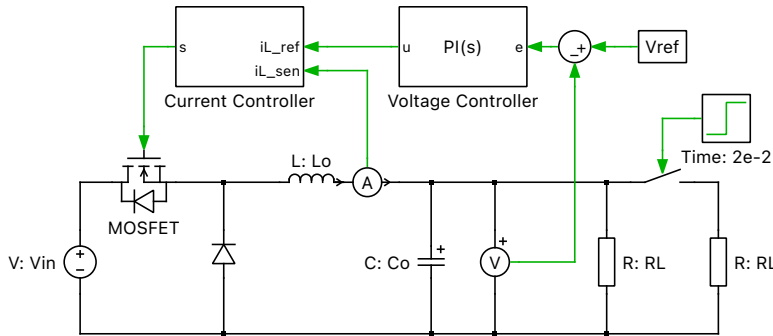
Last updated in PLECS 4.8.1

[www.plexim.com](http://www.plexim.com)

- ▶ Request a PLECS trial license
- ▶ Check the PLECS documentation

# 1 Overview

This demo uses a buck converter with resistive loads. The closed-loop controller is in cascaded form, which has an outer voltage controller and an inner current controller. The Voltage Controller is using the library Continuous PID Controller component. The Current Controller subsystem has two configurations. The first configuration uses a Constant On-Time Current Controller implemented with a State Machine component. The second configuration uses a Peak Current Controller library block. Fig. 1 shows the top-level schematic of the model.



**Figure 1: Buck converter with cascaded controls**

**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

### 2.1 Power circuit

The buck converter power circuit used in this demo is the same as described in the demo model “Buck Converter with Voltage Controls” in the PLECS demo models library. However, the modulation method used in this buck converter is not the conventional Pulse-Width Modulation (PWM) method. It uses two different current control methods to directly generate the on/off signal of the MOSFET. Therefore, the output signal of the Current Controller subsystem is directly the MOSFET gate signal  $s$ , instead of a modulation index value  $m$  which is the case in the “Buck Converter with Voltage Controls” demo.

### 2.2 Control

The controller is composed of an outer Voltage Controller in cascaded with an inner Current Controller. The users can choose in the mask dialog of the Current Controller subsystem between two configurations: Constant On-Time Control and Peak Current Control.

#### Voltage Controller

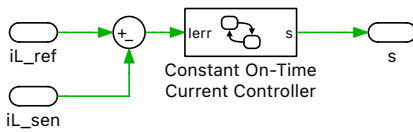
The outer-loop Voltage Controller is a proportional-integral (PI) controller from the library block Continuous PID Controller. The desired output voltage reference  $V_{\text{ref}} = 12 \text{ V}$  is directly compared with the

measured converter output capacitor voltage. The error passes through the PI controller calculation and outputs a current reference.

This current reference is then compared with the measured inductor current through the inner-loop Current Controller. The Current Controller generates directly the MOSFET gate signal  $s$ .

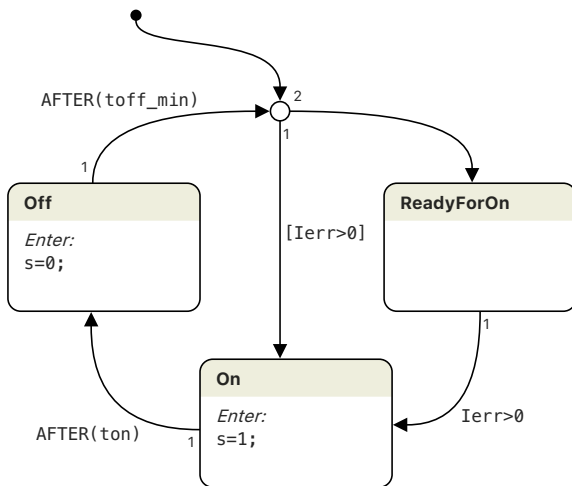
### Constant On-Time Current Controller

This current controller configuration is shown in Fig. 2. The current reference computed from the voltage controller is compared with the measured inductor current. The current compensator constant on-time control (COT) scheme is as follows: when the actual current  $iL\_sen$  becomes less than the reference current  $iL\_ref$ , the MOSFET is switched on for a fixed time duration. After the MOSFET is switched off, it must remain off for a minimum time duration before it can switch on again.



**Figure 2: Constant on-time controller schematic**

The COT scheme is implemented with a state machine that has the current error  $Ierr$  as its input and the MOSFET switching signal  $s$  as its output. Fig. 3 depicts the state machine diagram. The constant on-time and the minimum off-time are configured as internal constants  $ton$  and  $toff\_min$ . The state machine has two states (On and Off) that are used to switch the MOSFET on and off, and a waiting state ReadyForOn.



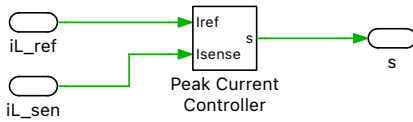
**Figure 3: Constant on-time controller using state machine**

When state On is entered, its Enter action will set the output signal  $s$  to 1. Exactly  $ton$  seconds after the start of state On, the state machine is executed again and the timer event  $AFTER(ton)$  becomes active. This causes the state machine to transition to state Off and the output signal  $s$  is set to 0. Exactly  $toff\_min$  seconds after this event, the state machine is executed again, and the timer event  $AFTER(toff\_min)$  becomes active. The transition for this timer event is branched with a Junction. If the input signal  $Ierr$  is already greater than 0 at this instant, the state machine transitions directly to state On again; else, the state machine transitions to the waiting state ReadyForOn and remains there until  $Ierr$  becomes greater than 0. The default transition targets the same Junction so that the state machine will start in the On state or the ReadyForOn state depending on whether  $Ierr$  is greater than 0 at simulation start.

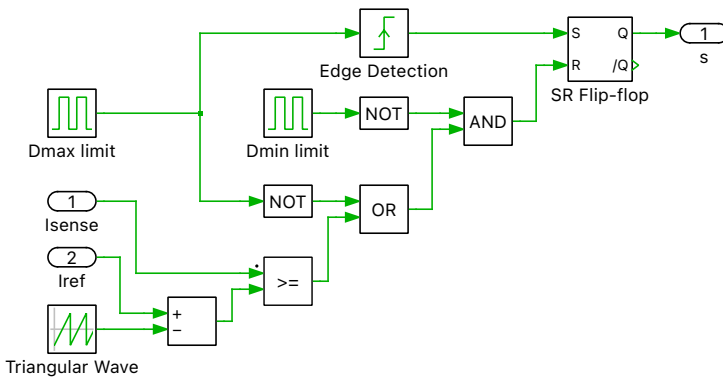
A COT implementation using the standard Control blocks is shown in the demo model “Series Capacitor Buck Converter” in the PLECS demo models library.

## Peak Current Controller

This current controller configuration is shown in Fig. 4. The switch modulation is generated by the PLECS Peak Current Controller (PCC) library block. Fig. 5 shows the implementation under the mask of this library block. At the beginning of each switching cycle, the output signal  $s$  is set. When the  $I_{sense}$  input exceeds the  $I_{ref}$  input, the output is reset. The next time when the switching cycle starts again, the the output signal  $s$  is set again. This controls the peak current flowing through the inductor.



**Figure 4: Peak current controller schematic**



**Figure 5: Implementation under the mask of the library Peak Current Controller block**

Inside this Peak Current Controller block, slope compensation is used to ensure stability when the duty cycle exceeds 50%. A minimum duty cycle and a maximum duty cycle are set to avoid saturation.

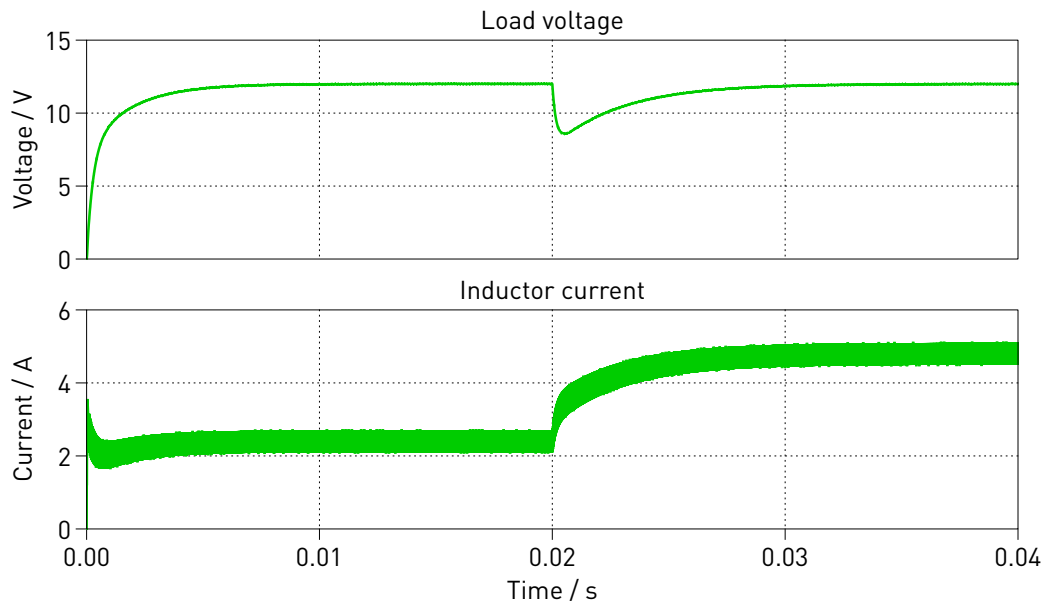
## 3 Simulation

First run the simulation for the default Current Controller configuration: Constant On-Time Control.

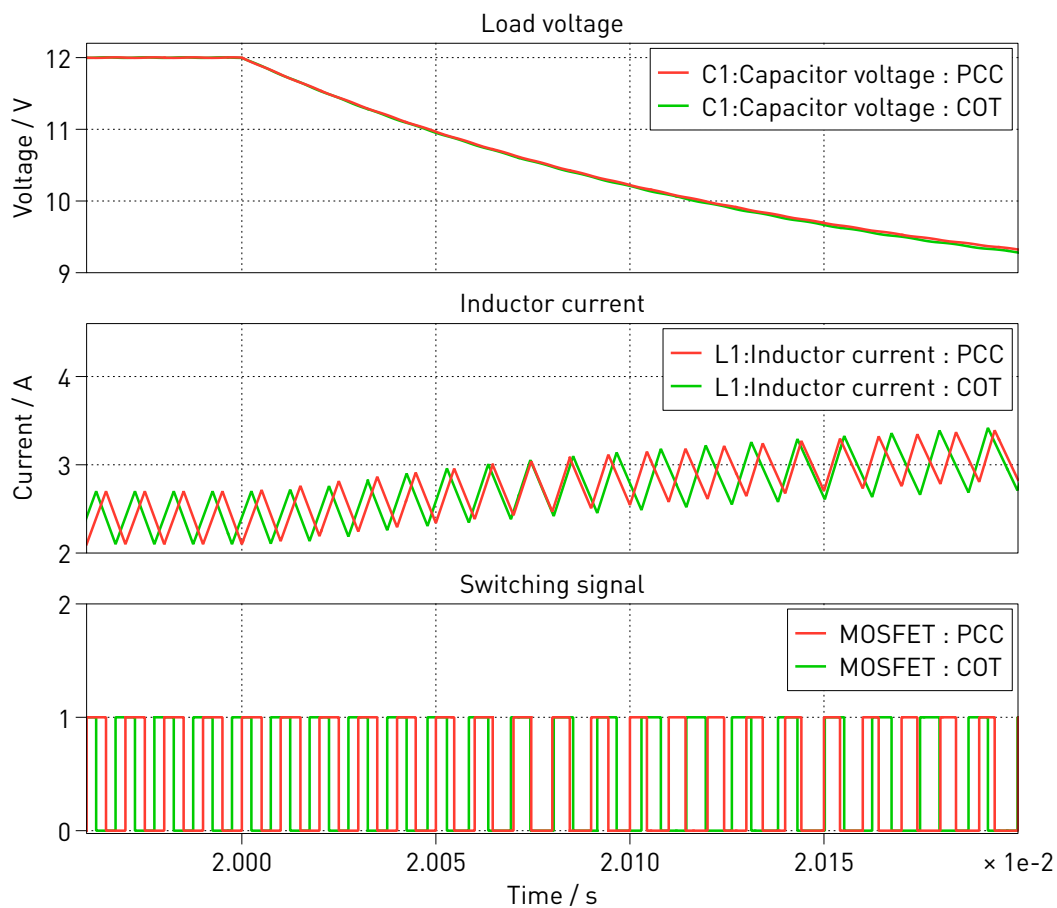
The simulation result is shown in Fig. 6. After starting the simulation, the output voltage starts rising from 0. After some initial dynamics it reaches the desired 12 V reference. At the time of 20 ms, the load resistance is halved. The current doubles and as the reaction of the cascaded controls, the perturbation in load voltage is regulated out in approximately 10 ms. This proves the stability of the closed-loop control.

After the simulation has finished, open the Constant On-Time Current Controller block (the State Machine). Open the State Machine Configuration dialog by clicking on the configuration icon. Then check the “Animation” box and start the simulation again. To view the animations on the zoomed-in Scope waveforms, step through the state machine transitions using the Space bar on the keyboard.

Next, change to the configuration: Peak Current Control and simulate. Fig. 7 shows the comparison of the overlaid two traces, zoomed in to the load step moment of 20 ms. The two configurations are both able to control the output voltage stably against the load step. However they show quite different waveforms, especially the inductor current and the MOSFET switching signal inside each switching period. This is because the COT is only with a fixed on-time but a variable switching frequency, and the PCC is with a fixed switching frequency.



**Figure 6: Simulation result of load voltage and inductor current for the Constant On-Time (COT) Control configuration**



**Figure 7: Zoomed-in result comparison of the Constant On-Time (COT) and the Peak Current Control (PCC) configuration around the load step**

## 4 Conclusion

This model demonstrates two different ways to model a current controller for a buck converter, in cascaded form with a voltage controller using PLECS components. The stability of the closed-loop control is validated by applying a load step.

## Revision History:

PLECS 4.3.1	First release
PLECS 4.4.2	Update PI controller component
PLECS 4.8.1	Make the Buck Converter with Constant On-Time Control and the Buck Converter with Peak Current Control demo obsolete, combine them into this demo

## 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
	<a href="http://www.plexim.com">http://www.plexim.com</a>	Web

## *PLECS Demo Model*

© 2002–2023 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.