

PLECS

*DEMO MODEL*

## Buck Converter with Constant On-Time Control

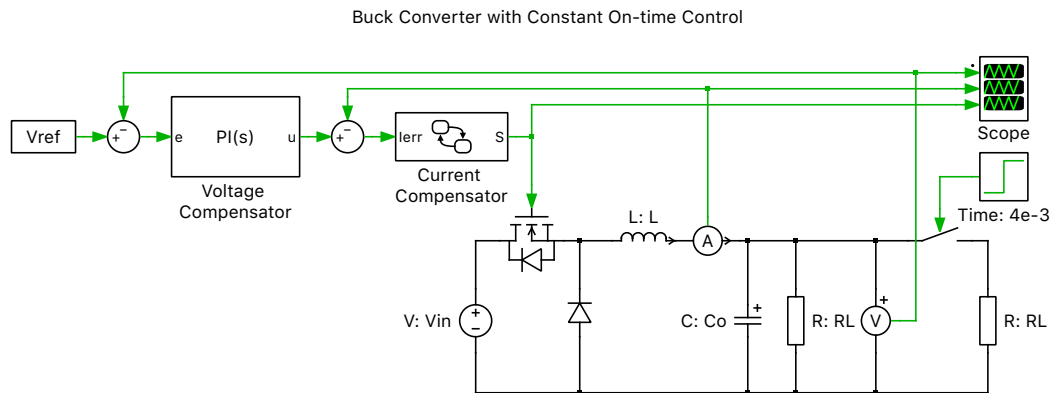
Last updated in PLECS 4.4.2

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

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

# 1 Overview

The demonstration shows a buck converter with constant on-time control implemented using the PLECS State Machine block.



**Figure 1: Buck converter with constant on-time control, state machine**

**Note** This model contains model initialization commands that are accessible from:

*PLECS Standalone:* The menu **Simulation + Simulation Parameters... + Initializations**

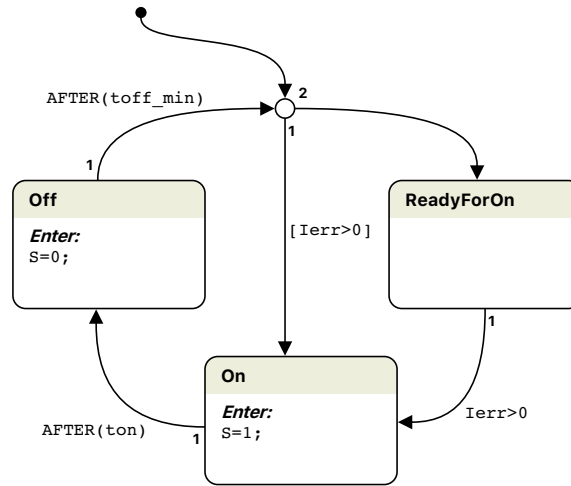
*PLECS Blockset:* Right click in the **Simulink model window + Model Properties + Callbacks + InitFcn\***

## 2 Model

The outer loop voltage compensator is implemented using a proportional-integral (PI) controller. This computes a current reference from the difference between the actual output voltage and the voltage reference. The inner loop current compensator uses a constant on-time control (COT) scheme as follows: when the actual current becomes less than the reference current, the FET is switched on for a fixed duration. After the FET is switched off, it must remain switched off for a minimum duration before it can switch on again.

The COT scheme is implemented with a state machine that has the current error  $I_{err}$  as its input and the FET switching signal  $S$  as its output. The constant on-time and the minimum off-time are configured as internal constants  $t_{on}$  and  $t_{off\_min}$ . The state machine has two states On and Off that are used to switch the FET on and off, and a waiting state ReadyForOn.

When state On is entered, its Enter action will set the output signal  $S$  to 1. Exactly  $t_{on}$  seconds after the start of state On, the state machine is executed again and the timer event  $AFTER(t_{on})$  becomes active. This causes the state machine to transition to state Off and the output signal  $S$  is set to 0. Exactly  $t_{off\_min}$  seconds after this event, the state machine is executed again, and the timer event  $AFTER(t_{off\_min})$  becomes active. The transition for this timer event is branched with a Junction. If the input signal  $I_{err}$  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  $I_{err}$  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  $I_{err}$  is greater than 0 at simulation start.



Configuration: Constant On-Time Controller

**Figure 2: Constant on-time controller using state machine**

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

### 3 Simulation

Run the simulation to view the output voltage, inductor current and gate signal waveforms. After the simulation has finished, open the Current Compensator (State Machine based Modulator) block. Open the state machine configuration dialog by clicking on the configuration icon. Then check the “Animation” box and restart the simulation. To view the animations, step through the state machine transitions using the Space bar.

## Revision History:

PLECS 4.3.1	First release
PLECS 4.4.2	Update PI controller component

## 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*

© 2002–2021 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.