Embedded
Code Generation

DEMO MODEL

Boost Converter with Peak Current Control

Peak current control of a boost converter with embedded code generation for TI C2000 MCUs

Last updated in C2000 TSP 1.4.1

1 Overview

This demo model features a boost converter circuit with peak current mode control. The peak current control is implemented using the Peak Current Control (PCC) component of the TI C2000 Target Support Library. The component integrates several MCU peripherals including a high-resolution timer, comparator, and a digital-to-analog converter to achieve the desired PCC functionality.

The model is split into two distinct subsystems called “Plant” and “Controller”. The plant contains a boost converter circuit, and the controller uses the TI C2000 PCC component to implement peak current mode control. Each subsystem is deployed to a separate real-time target. The control logic in the controller subsystem is built and then flashed to a TI C2000 MCU. The plant subsystem is deployed on the PLECS RT Box for Hardware-in-the-loop (HIL) testing of the generated embedded code.

The following sections provide a brief description of the model and instructions on how to simulate the model, and deploy it to the respective real-time targets.

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 top level schematic contains two separate subsystems representing the controller and plant models, as shown in Fig. 1. Both subsystems are enabled for code generation from the **Edit + Subsystem + Execution settings...** menu. This step is necessary to generate the model code for a subsystem via the PLECS Coder.

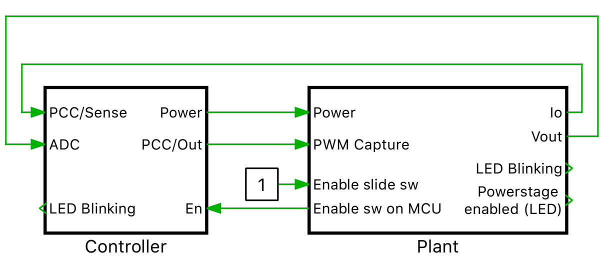

Figure 1: Top level schematic of the plant and the controller subsystems

2.1 Power Circuit

The power circuit, shown in Fig. 2, is supplied by a DC source voltage of $V_{\mathrm{dc}} = 5\,\mathrm{V}$. The boost converter circuit uses the power module component, with sub-cycle averaging. The pulse-width modulated (PWM) switching signals are then obtained from the PWM Capture block of the PLECS RT Box library. Further detail on the power module components and the sub-cycle averaging of PWM signals is described in [1].

The DC output voltage and inductor current measurements (the inductor current is sensed at the MOSFET source pin) are connected to Analog Out blocks from the PLECS RT Box library. These values are scaled to be within $0\,\mathrm{V}$ to $3.3\,\mathrm{V}$, to satisfy voltage limits of the MCU analog-to-digital converters (ADCs). The discretization step size of the plant subsystem is set to $2\,\mu\mathrm{s}$.

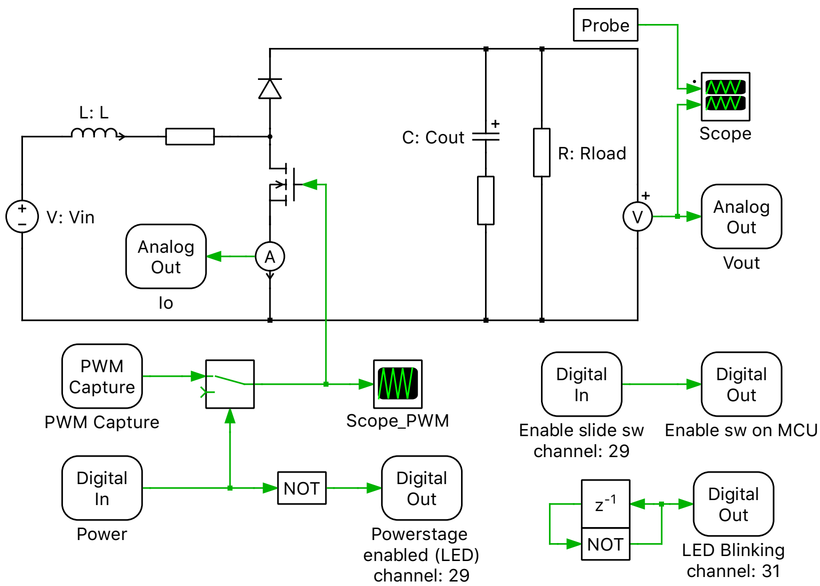

Figure 2: Power circuit of the boost converter

2.2 Controls

The controller subsystem is shown in Fig. 3. The “Peak Current Controller” component is responsible for current regulation. An outer voltage control loop supplies the peak current reference to the PCC block. This is implemented using a PI controller.

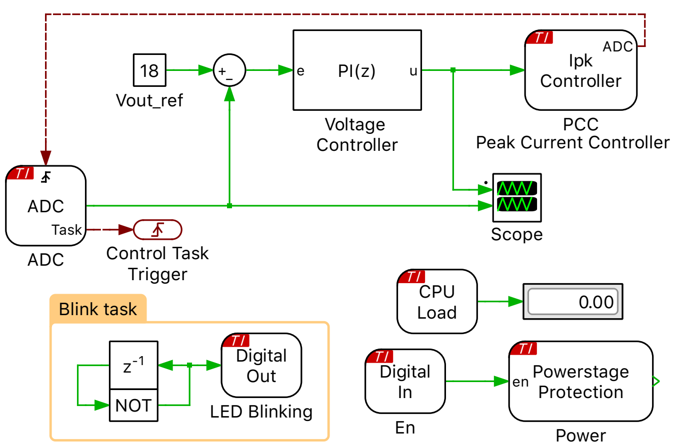

Figure 3: Controller of the H-Bridge circuit

Peak current controller

In a peak current-mode controller, at the beginning of each switching cycle the output is set (gate signal is turned ON) without a pre-determined duty cycle. Then, when the sensed inductor current exceeds the peak current reference value, the output is reset (gate signal is turned OFF). The duty cycle is therefore determined by the rise of the inductor current during the on-time.

One of the drawbacks of the peak current-mode controller is that it suffers from an inherent instability if the applied PWM duty cycle is greater than 50%. This is explained in Fig. 4. If a small disturbance is introduced into the system and if the applied duty cycle is less than 50%, the disturbance eventually decays to zero. However, if the applied duty cycle is greater than 50%, the inductor current will start to diverge and will no longer be stable. The resulting duty cycle values will vary from small to large, on an alternating cycle basis, called sub-harmonic oscillations. To limit these sub-harmonic oscillations, instead of providing a constant peak current reference, additional slope compensation is applied, as shown in Fig. 4. This ensures the stability of the inductor current [3].

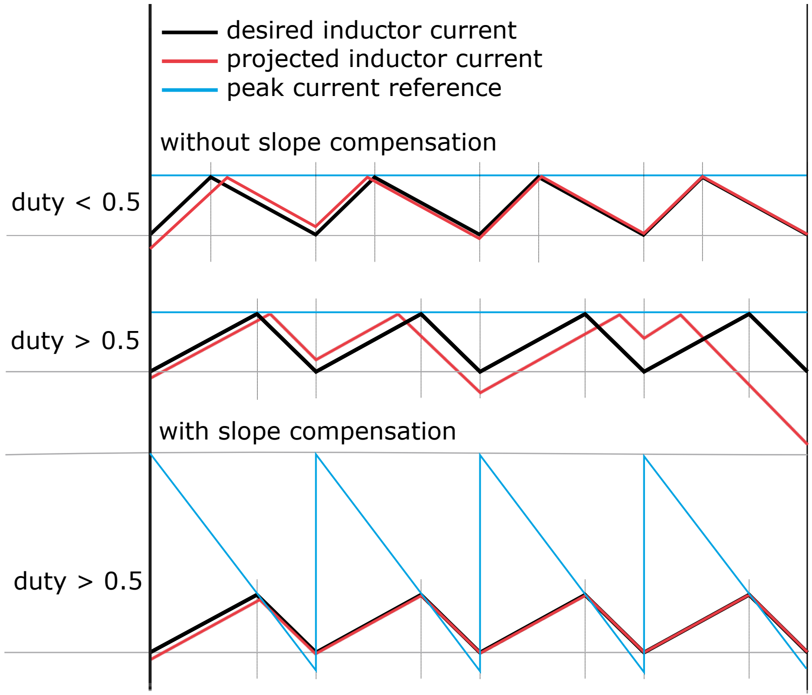

Figure 4: Slope compensation

Voltage Compensator

Plant transfer function To set the PI controller gain parameters a plant transfer function is needed. The transfer function for the outer voltage loop is $V(s)$. The voltage loop is designed to be slower than the current loop so that it does not distort the current reference.

$V(s)$ relates the change of the current through the inductor (L), I_L (the input variable), to the response of the capacitor voltage V_C (the output variable):

$$V(s) = \frac{V_\mathrm{C}}{I_\mathrm{L}} \approx \frac{1}{s\,C} = \frac{1}{s\,T_2}, \quad \text{where}\, T_2 := C$$

Equivalent delay The control system often introduces several small delays (e.g. from sensors, actuators, sampling, calculation delays, PWM delay). It is generally assumed that these delays are smaller than the time constant of the plant. If T_Σ is the equivalent delay of the control system, then the simplified transfer function of the delay is:

$$D_\Sigma(s) = \frac{1}{1 + sT_\Sigma}$$

The delays present in this model for this specific implementation are:

- a small time constant for control calculation, T_calc, is $\frac{1}{2}\,T_\mathrm{s}$
- a small time constant for PWM output generation, t_pwm, is $\frac{1}{2}\,T_\mathrm{s}$ where, T_s is the sample time of the controller.

The equivalent delay, T_Σ, is therefore:

$$T_\Sigma = T_\mathrm{calc} + t_\mathrm{pwm} = T_\mathrm{s}$$

Calculation of control parameters for the voltage loop The control parameters of the voltage PI controller (K_p and K_i), since the outer voltage loop of the plant represents a pure integrator, are calculated using the Symmetrical Optimum Criterion. The system's open-loop transfer function $V_\mathrm{OL}(s)$ is given by the product of the transfer functions of the controller, equivalent time delay and plant:

$$V_\mathrm{OL}(s) = \frac{1+sT_\mathrm{n}}{sT_\mathrm{i}} \cdot \frac{1}{1+sT_\Sigma} \cdot \frac{1}{s\,T_2}, \quad \text{where } K_\mathrm{p} = \frac{T_\mathrm{n}}{T_\mathrm{i}} \text{ and } K_\mathrm{i} = \frac{1}{T_\mathrm{i}}$$

After solving the corresponding closed-loop transfer function, the final coefficients are:

$$T_\mathrm{n} = 4 \cdot T_\Sigma \text{ and } T_\mathrm{i} = 8 \cdot \frac{{T_\Sigma}^2}{T_2}$$

For a more detailed explanation on calculating the controller parameters, refer to [4].

Configuring TI C2000 Target library components

The controller in Fig. 3 contains several components from the TI C2000 Target library.

- **Peak Current Controller:** The main TI C2000 target component used in the model is the peak current controller (PCC) block. This component implements peak current control with slope compensation.

  Internally, the PCC block makes use of multiple MCU peripherals, as shown in Fig. 5. The first component is a DAC that provides a peak current set-point including ramp, for controlling the inductor current. The second is a comparator (COMP); the current sensed via the MOSFET source pin in Fig. 2 is fed to the comparator, which is then compared to the peak current reference provided by the DAC. The output of the COMP block is fed to the third component, which is the PWM generator. The PWM generator generates the PWM waveforms at a frequency of 200 kHz.

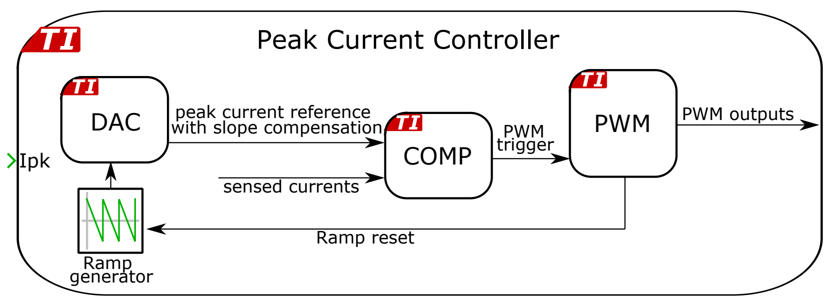

Figure 5: Peak current controller schematic

  To prevent the turn-on transient currents from triggering the peak current controller, leading edge blanking time is applied. When leading edge blanking time is applied, the first turn-on transient peak is ignored, and the duty cycle will continue to increase until the sensed inductor current exceeds the desired peak current reference value.

- **ADC:** The measurements of the output voltage are introduced to the model environment from the ADC block of the TI C2000 Target component library. Scaling and offset factors are provided to each channel via the parameter window of the ADC block in order to convert the detected analog voltage into values with physical units to be used for the control algorithm. The **ADC unit** and the **Analog input channel** parameters can be modified according to the available resources of different MCUs. The control task is executed after the last conversion on ADC module. This is configured by connecting the Task output of the ADC to the Control Task Trigger block from the TI C2000 Target component library.
- **Powerstage Protection:** In order to enable or disable PWM signals during runtime, DIP switch "DI-29" on the RT Box LaunchPad Interface board is used. This input signal "DI-29" is connected to the Digital in block labeled "Enable slide sw" in the "Plant" subsystem, which is then routed as the

input of the Powerstage Protection block on the "Controller" subsystem through the RT Box LaunchPad Interface board. The Powerstage Protection block implements a finite state machine to enable or disable all PWM outputs on the target MCU. A logic low to high transition enables the PWM outputs, while a high to low transition disables them. For more details, please browse the **Help** section of this block.

When the power stage is enabled a digital output, configured in the **Powerstage enable GPIO number** of the Powerstage Protection block, is toggled. This signal is connected to the Digital In block labeled "Power" in the "Plant" subsystem. This allows the captured PWM signals to pass to the gates of the inverter bridge modeling a gate driver enable circuit. The red LED "DO-29" on the LaunchPad Interface board will turn on, visually indicating the switching signals are connected to the gate drivers.

2.3 Time-Scaling Concepts

Real-time simulation involves a trade-off between model complexity and model fidelity. As a model becomes more numerically complex it naturally takes a longer time to compute. Similarly, if a very short discretization step size is required to meet the model fidelity requirements, a limited amount of time is available for the processor to compute the model results in real time. At a certain point the execution time of the model may exceed the required discretization time step.

Time-scaling is one approach to overcome this limitation, where the execution of the RT Box plant model and the embedded controller are both slowed, in lock-step, as compared to real time. If the model is time-scaled to run at $1/10^{\mathrm{th}}$ of real time, then a 1 second event would occur over 10 seconds in a time-scaled model. Through careful manipulation of the RT Box parameters the "dynamics" of the model are retained, but the RT Box has additional time to perform the necessary calculations.

If we consider a time-scaling factor K_{scale} where $K_{\mathrm{scale}} < 1$ corresponds to slower than real time, then to time-scale the plant model for the RT Box, all inductances, capacitances, and time constants are divided by K_{scale}. Frequencies are multiplied by K_{scale}. One could consider this as changing all L/R and $1/(RC)$ time constants in the electrical circuit by a common factor. The model discretization step size is divided by K_{scale} as well.

Changes to the controller are also required. When using the TI C2000 TSP, the Timer "Frequency" setting and PWM "Carrier frequency" are multiplied by K_{scale} to increase the control task and PWM periods. The voltage compensator parameters are changed accordingly.

If the MCU code is hand-written, then comparable changes are required. While time-scaling does require altering the controller software, minimal changes to the code-base allow one to test the embedded controller in conjunction with the RT Box with a high-fidelity plant model. Then, when the controller is interfaced with real hardware the modifications required for time-scaling would be removed.

3 Simulation

In addition to running a simulation of this demo model in offline mode on a computer, the "Controller" subsystem can be directly converted into target specific code for the TI LaunchPads. The model is configured by default for a TI 28379D LaunchPad [6], but other LaunchPad targets are supported as explained later in this section.

Follow the instructions below to upload the "Controller" subsystem to a TI MCU.

- Connect the MCU to the host computer through a USB cable.
- From the **System** tab of the **Coder + Coder options...** window, select "Controller".
- Next, from the **Target** tab, select the appropriate target from the dropdown menu. Then under the **General** sub-tab, select the desired **Build type**.
- Then, to `Build and program` the MCU directly from PLECS, choose either `Run from Flash` or `Run from RAM` as the **Build configuration** to program the MCU either to flash memory or to RAM respectively, then select `LaunchPad` as the **Board** type, and click **Build**.

If programmed correctly, LED "D9" (or the LED corresponding to GPIO "DO_DSP_LED" listed in the model initialization commands) should blink.

For advanced users who are familiar with Code Composer Studio, there is an option to `Generate code into CCS project`. Locate the appropriate cg folder from the CCS project (refer to [7] for step-by-step instructions), enter its path into the **CCS project directory** field and click **Build**. The code of the "Controller" subsystem will be automatically generated. Then, proceed to build and debug the project as a normal CCS project.

Note If using the RT Box LaunchPad Interface board, make sure that the **RST** jumper is open throughout the simulation.

Prior to controlling a real power stage with the programmed MCU, it is highly recommended to first verify the behavior of the controller using a PLECS RT Box and perform a hardware-in-the-loop (HIL) test. A typical hardware configuration is shown in Fig. 6, where the evaluation kit, a TI C2000 LaunchPad (the red board), is connected to the RT Box via an RT Box LaunchPad Interface (the green board).

Figure 6: Hardware setup of the HIL verification

Follow the instructions below to run a real-time model on the RT Box.

- From the **System** tab of the **Coder + Coder options...** window, select "Plant". Click the **Target** tab and select a target device. Then click **Build** to deploy the model to the target RT Box.
- Once the model is uploaded, from the **External Mode** tab of the **Coder options...** window, **Connect** to the RT Box and **Activate autotriggering** to observe the test results in real-time.

If programmed correctly, the LED corresponding to "DO-31" of the RT Box LaunchPad Interface board should blink.

Toggle the switch "DI-29" on the RT Box LaunchPad Interface board from low to high to enable the MCU, as explained at the end of Section 2.2. When the power stage is enabled, the LED corresponding to "DO-29" of the LaunchPad interface board should turn on. Observe the real-time waveforms in the Scope of the "Plant" subsystem.

Note At this stage, verify that the LED corresponding to "DO-29" on the RT Box LaunchPad Interface board is turned on.

Toggling the switch “DI-29” on the RT Box LaunchPad Interface board from high to low should disable all the gating signals. “DO-29” of the LaunchPad Interface board should turn off. Toggling “DI-29” back to high will enable the PWM outputs once again.

In order to tune the parameters of the control program in the MCU and observe any intermediate values, follow the instructions below to connect to the external mode of the TI MCU.

- First, **Disconnect** the “Plant” subsystem from the **External Mode** of the PLECS RT Box, if connected.
- Then, from the **System** menu on the left hand side of the **Coder + Coder options...** window, select “Controller”.
- Next, from the **External Mode** tab, select the appropriate **Target device** and click **Connect**.
- Then, **Activate autotriggering** to observe the test results in the “Controller” subsystem Scope.

4 Conclusion

This model demonstrates a boost converter with peak current mode control that supports embedded code generation for TI C2000 MCUs. It can be run in both offline mode, as well as in real-time. The model has also been verified with a power circuit prototype. The model also demonstrates the **Parameter Inlining** feature using a current controller reference that can be changed in real-time.

References

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

[2] TI Application Report SLVA636: Practical Feedback Loop Analysis for Current-Mode Boost Converter,

URL: https://www.ti.com/lit/an/slva636/slva636.pdf.

[3] NPTEL lectures from Indian Institute of Science, Bangalore. Click to access online:

[4] Conception de systèmes automatiques, Hansruedi Bühler, Presses Polytechniques Romandes, Lausanne 1988, ISBN 2-88074-149-1

[5] TI C2000 Piccolo MCU F280049C LaunchPad Development Kit

URL: http://www.ti.com/tool/LAUNCHXL-F280049C.

[6] TI C2000 Delfino MCUs F28379D LaunchPad Development Kit,

URL: http://www.ti.com/tool/LAUNCHXL-F28379D.

[7] PLECS TI C2000 Target Support User Manual,

URL: https://www.plexim.com/download/documentation.

Revision History:

C2000 TSP 1.4 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

Embedded Code Generation Demo Model