



**Embedded  
Code Generation**  
*DEMO MODEL*

## LLC Resonant Converter

**Control of an LLC resonant converter with embedded code generation  
for TI C2000 MCUs**

Last updated in C2000 TSP 1.11.1

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

- ▶ Request a PLECS and PLECS Coder trial license
- ▶ Get the latest TI C2000 and RT Box Target Support Package
- ▶ Check the PLECS, RT Box and TI C2000 TSP documentation

# 1 Overview

The half-bridge LLC resonant converter is an attractive topology for DC/DC converters in applications such as battery chargers and power supplies, as it allows for zero voltage switching (ZVS) over a wide range of operating conditions resulting in comparatively low switching losses, high efficiency, and high power density.

This demo model highlights a variable switching frequency control approach for the LLC converter using a Texas Instruments (TI) C2000 microcontroller (MCU). The MCU runs embedded code generated from a PLECS model. The power stage for the model is based on the TI LLC Resonant Half-Bridge Converter 300 W Evaluation Module [1].

Hardware-in-the-Loop (HIL) testing is used to evaluate the controller performance. The plant model is optimized to run on the PLECS RT Box. Use of the Half-Bridge LLC Converter Nanostep module from the PLECS library is key to obtain accurate real-time simulations with the high switching frequencies in this application.

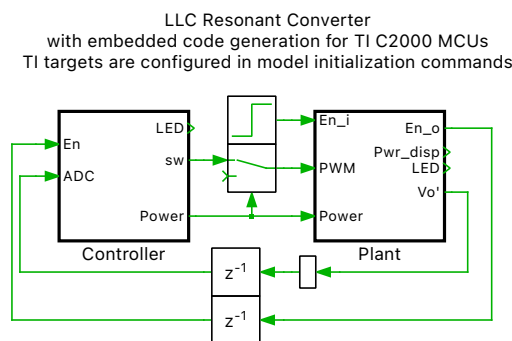
**Note** This model contains model initialization commands that are accessible from:

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

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

# 2 Model

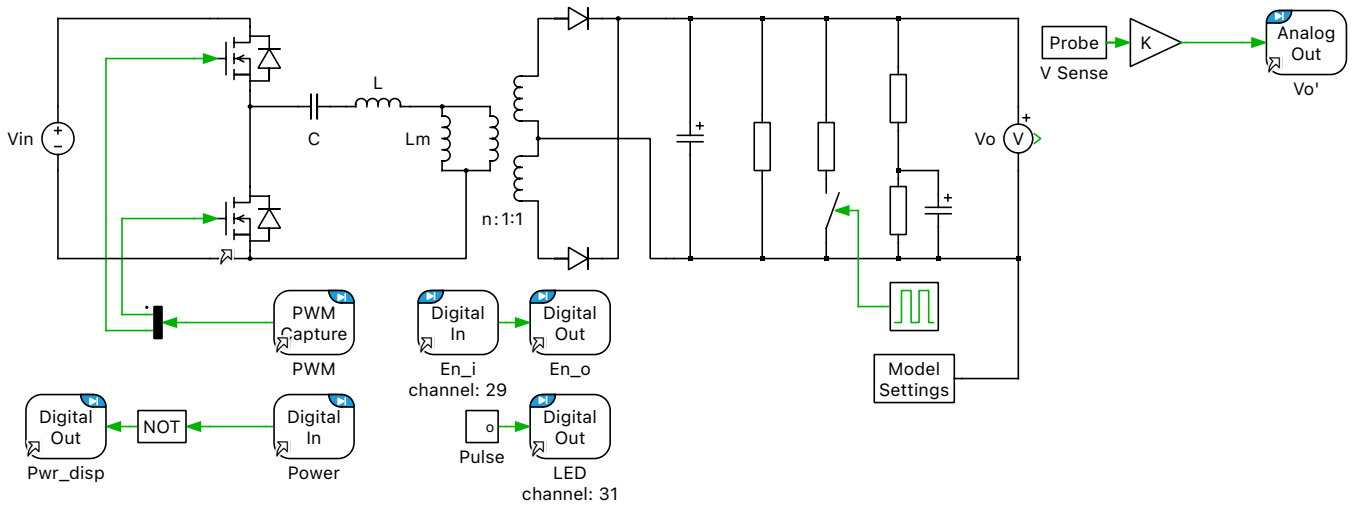
The top-level schematic contains two separate subsystems representing the controller and plant models, as shown in Fig. 1. Both subsystems are enabled for code generation from the **Edit + Subsystem + Execution settings...** menu. This step is necessary to generate the model code for a subsystem via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



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

## 2.1 Power Circuit

The power stage for the model is shown in Fig. 2. The design is based on the TI LLC Resonant Half-Bridge Converter 300 W Evaluation Module which has a 390 V input and a regulated 12 V output.



**Figure 2: Plant model optimized for the RT Box**

## Theory of operation

In the LLC converter the half-bridge is typically modulated to generate a square-wave voltage signal at the input of the resonant tank. The resonant tank consists of a series resonant inductance  $L_r$ , a magnetizing inductance  $L_m$ , and a resonant capacitor  $C_r$ . The upper and lower switching elements are modulated in a complementary fashion (with an appropriate dead-time) and typically at 50% duty. The square-wave voltage input results in a quasi-sinusoidal current flowing within the resonant network. An output transformer, rectifier circuit, and filter establish a DC output voltage from the sinusoidal tank currents.

Changing the frequency of the square-wave voltage input from the half-bridge changes the effective impedance of the resonant tank, and therefore the resulting current output. Assuming a gain  $M$  from the resonant tank, the relationship between the input and output voltage is:

$$V_o = M \frac{1}{n} \frac{V_{in}}{2}$$

The series resonant frequency of the LLC converter,  $f_0$ , is the point where the  $L_r$  and  $C_r$  combination have zero impedance and the gain  $M$  is unity (in a lossless circuit). Typically the converter will operate near  $f_0$ .

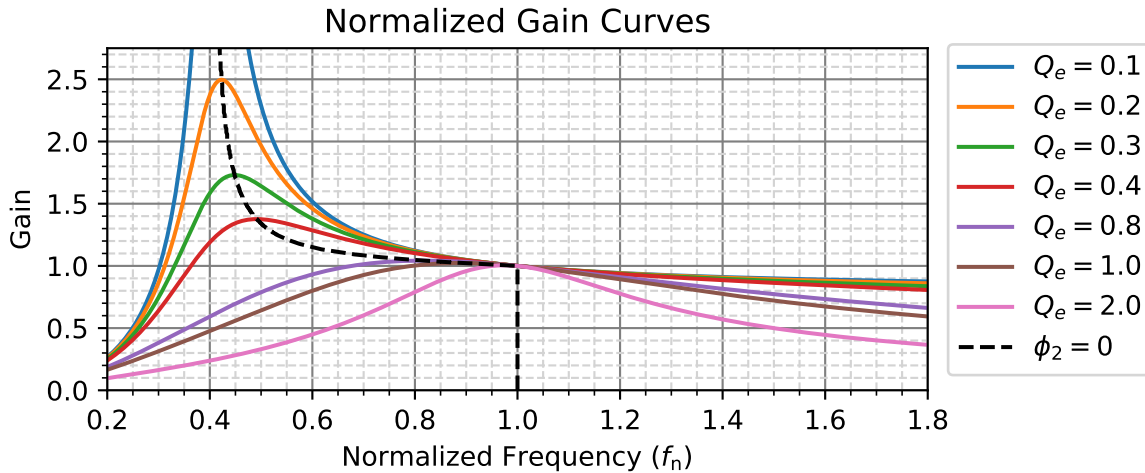
$$f_0 = \frac{1}{2\pi\sqrt{L_r C_r}}$$

However, the gain depends on not only the switching frequency and tank design, but also the converter load resistance  $R_o$ . The DC output resistance  $R_o$  can be approximated by an equivalent AC load resistance  $R_e$  [2].  $R_e$  is used to calculate the quality factor  $Q_e$ , which defines how the resonant gain changes as a function of frequency.

$$R_e = \frac{8n^2}{\pi^2} R_o \text{ and } Q_e = \frac{1}{R_e} \sqrt{\frac{L_r}{C_r}}$$

Fig. 3 shows a series of curves highlighting how the switching frequency and quality factor impact the gain of a lossless converter. For the power stage in this model,  $Q_e = 0.41$  at rated load and the resonant frequency is  $f_0 = 138.5$  kHz. [1]. The frequency axis is normalized such that  $f_n = f_{sw}/f_0$ .

In Fig. 3 the  $\phi_2$  curve shows the point where the phase of the resonant circuit is zero, indicating a transition from a capacitive current in the resonant tank at lower frequencies to an inductive current at higher frequencies. Maintaining an inductive current is essential to achieve ZVS on the primary half-bridge. This also maintains a consistent relationship where increasing the converter switching frequency results in a decreased voltage gain.



**Figure 3: Normalized gain curves for the converter**

### The Half-Bridge Resonant LLC Converter Nanostep module

The Half-Bridge LLC Converter is implemented with the respective component from the Nanostep section of the PLECS library. The RT Box's Nanostep solver simulates with time steps in the single-digit nanosecond range. At each time step the gate signals are sensed, the converter operating mode is determined, and the tank currents are calculated. The small step size is critical to accurately model high-frequency resonant converter topologies, such as the LLC, due to the high-frequency current components in the resonant tank and the frequent current reversal and clamping events.

#### RT Box configuration

The RT Box's PWM Capture target block connects to the gate signals of the Half-Bridge LLC Converter Nanostep component. Gate signals are evaluated by the Nanostep solver with the same single-digit nanosecond step size.

The effective load resistance is controlled by a Pulse Generator that toggles the output resistance between full load and half load. The variable load provides a disturbance which is used to evaluate the controller performance. A voltage divider provides a scaled version of the output voltage which is then routed to an RT Box analog output channel.

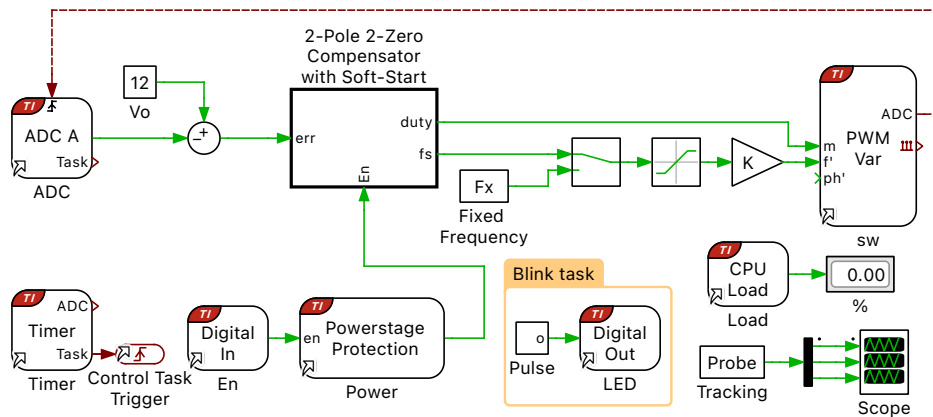
Additional digital signals are used so the user can enable or disable the controller and power stage through the use of an external status switch.

## 2.2 Controls

Fig. 4 shows a high-level overview of the control logic for the TI C2000 MCU.

The controller senses the input voltage via the analog-to-digital converter (ADC) and regulates the output voltage of the LLC converter by changing the switching frequency of the PWM output. Additionally, there is a manual signal switch so the converter can be operated at a user-specified fixed frequency, bypassing the regulator. The output of the compensator and the "Fixed Frequency" value are normalized to the series resonant frequency,  $f_0$ . The output is then limited to be in the expected operating range, such that at rated maximum load the converter will retain ZVS, and is scaled to generate the desired switching frequency.

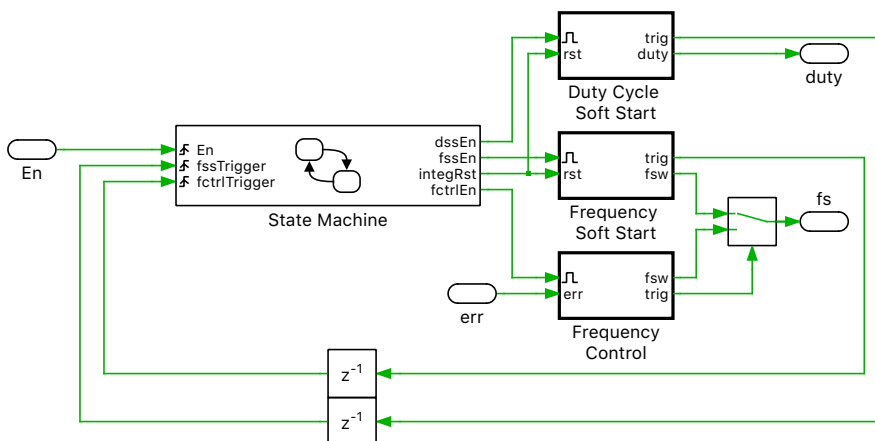
The description of the controls below is based on the operation with real power hardware in true real time.



**Figure 4: Overview of the embedded controller logic**

## Control architecture

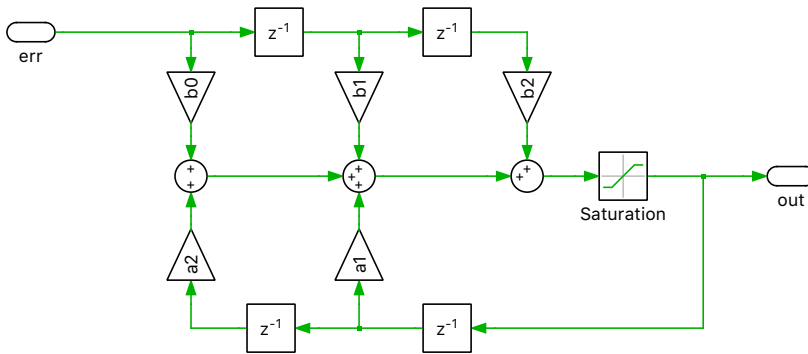
Fig. 5 shows the implementation of the compensator and soft-start mechanism. A State Machine is used to begin the soft-start sequence whenever the “En” input transitions from low to high. The State Machine inputs are triggers indicating the status of the soft-start actions. Outputs of the State Machine are used to control three different Enabled Subsystems, only one of which is active at a given time. The advantage of using the Enabled Subsystem is that when the enable input is low, the logic within the subsystem does not execute, reducing the computational load on the MCU.



### Figure 5: Controller soft-start implementation

The soft-start sequence begins from a switching frequency 1.5 times greater than the expected final operating point at the rated load. This ensures operation on the ZVS portion of the curve during soft start and a lower output voltage to prevent excessive gain under light-load conditions. With the frequency still fixed, the duty cycle is then linearly ramped from 0 % to 50 % duty over a fixed time period. A trigger signal is generated by the “Duty Cycle Soft Start” component to initiate the frequency soft-start mechanism. Next, the switching frequency is linearly ramped downwards to the expected final switching frequency. The “Frequency Soft Start” component sends a trigger upon completion that activates the “Frequency Control” subsystem, which contains a two-pole two-zero (2P2Z) compensator. If the sequence is aborted at any point by a logical low “En” input, the sequence starts again from the beginning.

The 2P2Z compensator is implemented as a Direct Form-I digital compensator with an output limitation, as shown in Fig. 6. The controller parameters are determined by first using Extended Describing Functions to obtain the plant transfer function, then pole-zero matching is used to design a controller in the continuous domain, and lastly the continuous controller is discretized [4]. The crossover frequency is 2 kHz in the final design.



**Figure 6: Two-pole two-zero (2P2Z) discrete implementation**

### Configuring TI C2000 Target library components

The schematic in Fig. 4 contains several components from the TI C2000 Target library. The PWM (Variable) component is used to generate a variable-frequency PWM signal. The input to the block, “f” is normalized to the **Carrier frequency** parameter in the component mask of 110 kHz. The PWM (Variable) block is configured to generate the start-of-conversion (SOC) signal for the ADC. From the **Events** tab of the PWM block parameter window, the **ADC Trigger** parameter is configured as Underflow. Then the **Trigger source** parameter of the ADC block is set to “Show trigger port”. The trigger signal is shown as a red dashed line connecting the PWM and ADC components.

The Timer component generates an interrupt at a fixed rate of half the nominal carrier frequency of the PWM (Variable) component, or 55 kHz. The Timer’s Task output triggers the control task, which is asynchronous to the PWM generation and ADC sampling. The control task uses the most recent ADC measurement when calculating the compensator response. Executing the control task at a fixed rate ensures the discrete 2P2Z compensator and other model calculations are appropriately discretized. Having the PWM trigger the ADC SOC ensures consistent point-on-wave sampling of the voltage measurement.

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 “En\_i” 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. The output of the Powerstage Protection block also enables the compensator and soft-start logic. 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. A Signal Selector component is used to model a gate driver. When the “Power” input is logical low the half-bridge gate signals are all zero. Once the “Power” input is logical high, then the switch connects the signals from the “PWM” input to the MOSFET gate signals. With the “Power” logical high, the red LED “DO-29” on the LaunchPad Interface board will also turn on, visually indicating the switching signals are connected to the MOSFET gates.

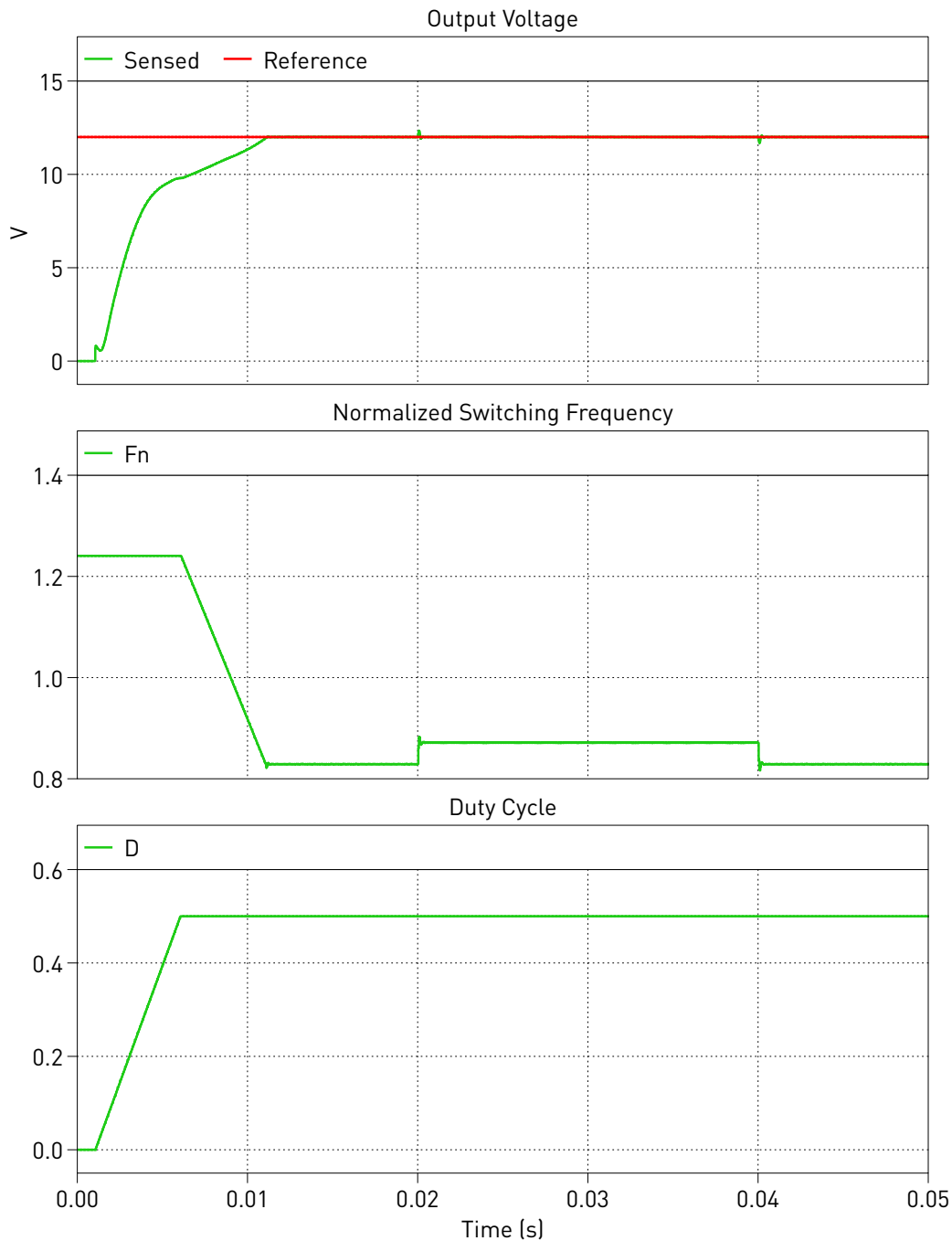
## 3 Simulation

The simulation model can be run offline on a desktop computer by choosing the **Simulation + Start** menu option.

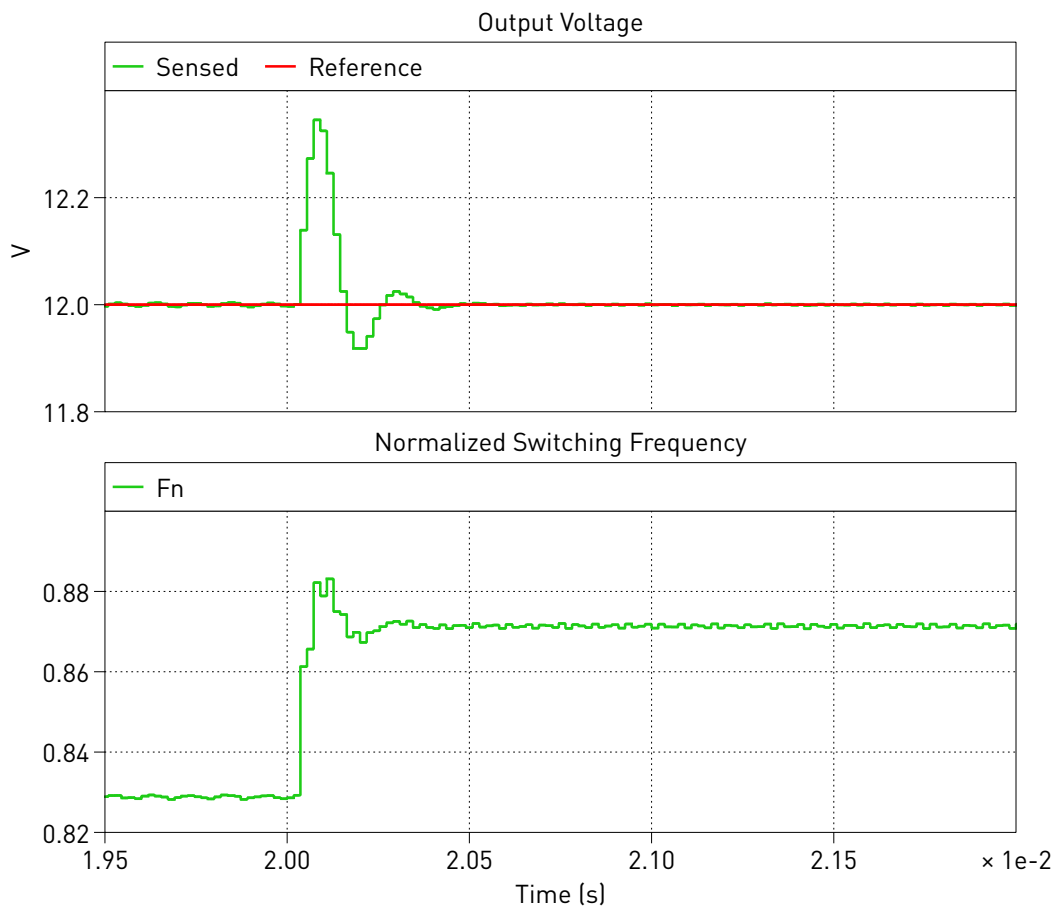
Fig. 7 shows the results from the “Controller” subsystem highlighting the startup sequence. At 1 ms the controller is enabled and the soft-start sequence begins. The output duty begins to increase from 0 % to 50% over 5 ms while the frequency is held constant. Then the frequency begins to ramp over the next

5 ms to the expected final operating point of  $\sim 0.83f_0$ . At 11 ms the 2P2Z compensator is activated and the output voltage is regulated to 12 V.

Step changes in the output load are visible at 20 ms and 40 ms resulting in brief voltage transients, as highlighted in Fig. 8



**Figure 7: Controller performance during startup**



**Figure 8: Controller performance during transient**

### 3.1 Configuring the TI C2000 Target

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 C2000 MCUs. The default I/O configuration of all the peripheral blocks (ADC, PWM etc.) supports the TI 280039C [5], TI 28379D [7], TI 28P550SJ9 [8], TI 28P650DK9 [9] LaunchPads, and the TI 28388D [12] controlCARD.

Additionally, the demo model allows for code generation for the TI 280039C [10] and TI 28379D [11] controlCARDS. To configure this, go to the **Model initialization commands** window from the **Simulation + Simulation parameters... + Initialization** menu, and change the value of `board_type`, to select the desired board. You must also configure the corresponding **Target** and **Board** type in the **Coder Options** window accordingly.

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 [13] 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.

## 3.2 Configuring the PLECS RT Box

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. 9, where the evaluation kit, a TI 28379D LaunchPad (the red board), is connected to the RT Box via an RT Box LaunchPad Interface (the green board).

Follow the instructions below to run a real-time model on the RT Box. Before building on the RT Box, ensure that you have already built the “Controller” subsystem on the appropriate TI MCU, as shown in Section 3.1.

- 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.



**Figure 9: Hardware setup of the HIL verification with the RT Box**

---

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

---

### 3.3 Starting the Controller and Connecting via the External Mode

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

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.

Once connected, it is possible to change the voltage reference for the closed-loop controller, or to click the manual signal switch and operate in a fixed-frequency mode. The “Fixed Frequency” value can be changed in real time to see how the plant responds to changes in the switching frequency and load changes.

One may also observe the startup sequence by configuring the External Mode to trigger on an appropriate signal, for example the rising edge of the Duty Cycle crossing 1 %. The **Trigger channel** is set in the **External Mode** tab.

## 4 Conclusion

This model demonstrates the closed-loop control of a half-bridge LLC resonant converter using embedded code generation for TI C2000 MCUs. It can be run in both offline mode and in real time. The controller features a soft-start mechanism and a discrete implementation of a 2P2Z compensator. The Nanostep solver was used to enhance the model fidelity for HIL testing.

## References

- [1] UCC25600EVM LLC Resonant Half-Bridge Converter 300-W Evaluation Module  
URL: <http://www.ti.com/tool/BOOSTXL-DRV8305EVM>
- [2] Texas Instruments Power Supply Design Seminar: Designing an LLC resonant half-bridge power converter, Sept 2019.  
URL: <https://www.ti.com/seclit/ml/slup263/slup263.pdf>.
- [3] 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.
- [4] D. Patel and R. Kankanala, “Digital Compensator Design for LLC Resonant Converter”, Microchip Application Notes AN1477, December 2018.  
URL: <http://ww1.microchip.com/downloads/en/AppNotes/AN14177,-Digital-Compensator-Design-for-LLC-Resonant-Converter-DS00001477B.pdf>

- [5] TI C2000 F280039C LaunchPad development kit,  
URL: <http://www.ti.com/tool/LAUNCHXL-F280039C>.
- [6] TI C2000 Delfino MCU F28377S LaunchPad development kit,  
URL: <https://www.ti.com/lit/pdf/sprui25>.
- [7] TI C2000 Delfino MCU F28379D LaunchPad development kit,  
URL: <http://www.ti.com/tool/LAUNCHXL-F28379D>.
- [8] TI C2000 F28P550SJ9 LaunchPad development kit,  
URL: <https://www.ti.com/tool/LAUNCHXL-F28P55X>.
- [9] TI C2000 F28P650DK9 LaunchPad development kit,  
URL: <https://www.ti.com/tool/LAUNCHXL-F28P65X>.
- [10] TI C2000 F280039C controlCARD evaluation module,  
URL: <https://www.ti.com/tool/TMDSCNCD280039C>.
- [11] TI C2000 F28379D controlCARD development kit,  
URL: <https://www.ti.com/tool/TMDSCNCD28379D>.
- [12] TI C2000 F28388D controlCARD evaluation module,  
URL: <https://www.ti.com/tool/TMDSCNCD28388D>.
- [13] PLECS TI C2000 Target Support User Manual,  
URL: <https://www.plexim.com/download/documentation>.

## Revision History:

C2000 TSP 1.2	First release
C2000 TSP 1.3.1	Turn on Assertions in the Half-Bridge LLC Resonant Converter power module
C2000 TSP 1.4.5	Updated the web links
C2000 TSP 1.5.1	Added support for 28388D and 28379D controlCARD targets
C2000 TSP 1.6.1	Added support for 280039C LaunchPad and controlCARD targets, and auto-pin selection
C2000 TSP 1.9.1	Added support for 28P550SJ9 LaunchPad target
C2000 TSP 1.11.1	Upgrade plant to Nanostep solver

## 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

### *Embedded Code Generation Demo Model*

© 2002–2024 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.