Embedded
Code Generation

DEMO MODEL

Dual-core Control of an On-Board Charger

Dual-core control of a single phase on-board charger with totem pole PFC and dual active bridge.

Last updated in C2000 TSP 1.7.1

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

On board chargers (OBCs) are the power electronic interface between an electric vehicle's battery pack and an AC voltage supply. This demonstration highlights a 3.6 kW single-phase OBC with a cascaded AC/DC to DC/DC architecture. The AC/DC converter is a totem-pole power factor correction (PFC) that regulates the intermediate DC bus voltage and ensures low distortion on the AC current input. The DC/DC converter is a dual active bridge (DAB) that regulates the battery currents and voltages.

Dual-core microcontrollers (MCUs) are a logical choice for controlling two-stage power conversion topologies, offering a simplified control architecture and reduced parts count. The two processing cores operate independently, with one CPU core controlling the PFC stage and the second CPU core controlling the DAB. Data is exchanged between the two cores as needed, for example to ensure the DAB controller activates only once the PFC has stabilized the DC bus voltage.

This demo illustrates how to use a simulation model to generate code for a dual-core MCU including the configuration of multi-tasking and inter-process communication (IPC). Hardware-in-the-Loop (HIL) testing is used to evaluate the controller performance. The OBC power stage is configured to run on the PLECS RT Box 2 or 3, which offers multiple real-time simulation cores.

1.1 Requirements

To run this demo model in real-time, the following items are needed (available at `www.plexim.com`):

- One PLECS RT Box 2 or 3 with PLECS and PLECS Coder license
- One TI C2000 MCU LaunchPad or ControlCard with one RT Box LaunchPad Interface board or RT Box ControlCard Interface board
- A supported dual-core MCU target: TI 28379D LaunchPad, TI 28379D ControlCard, or a TI 28388D ControlCard.
- The RT Box Target Support Package
- The TI C2000 Target Support Package

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.

2.1 Power circuit

The OBC circuit shown in Fig 2 consists of a two-stage power conversion process. The totem-pole PFC converts from 230 Vrms to a 400 VDC intermediate bus with a 65 kHz switching frequency. The DAB converter switches at 100 kHz and regulates the battery charging.

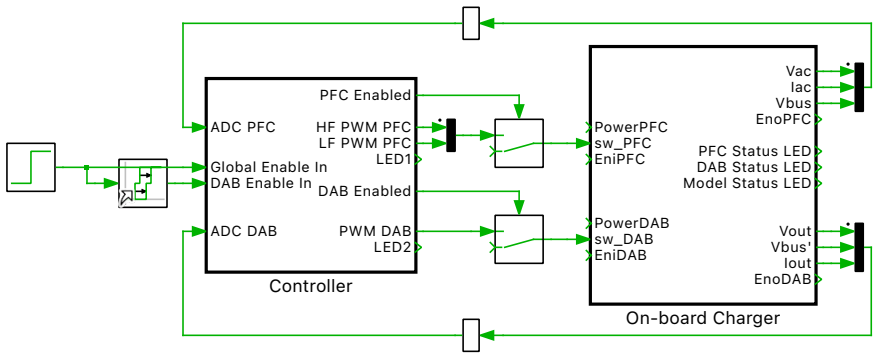

Figure 1: Top level schematic of the on-board and controller subsystems

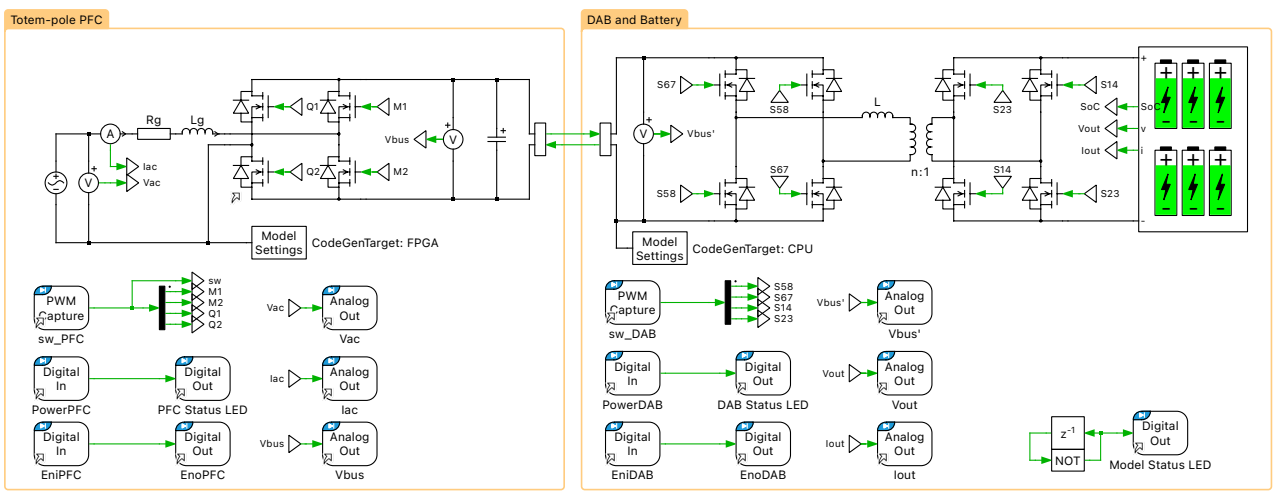

Figure 2: Schematic of the on-board charger system

Totem-pole PFC

The totem-pole PFC is an attractive circuit for high-power AC/DC conversion as it eliminates the diode conduction loss associated with boost PFC and bridgeless PFC topologies [1]. The topology consists of two half-bridges and an AC side inductance. One half-bridge switches at 65 kHz and the second half-bridge switches at the AC grid frequency, as shown in Fig. 3. Different semiconductor technologies can be selected for each half-bridge. The high-frequency switches are typically wide-bandgap devices due to their low reverse recovery and low conduction losses while the grid-frequency switches can be either Silicon MOSFETs or wide-bandgap devices.

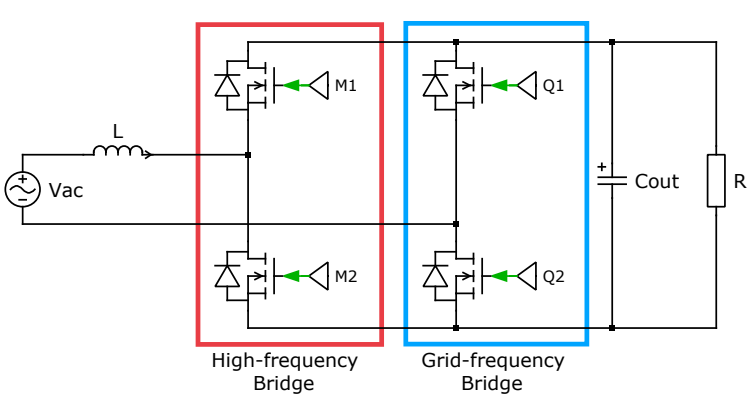

Figure 3: Totem-pole PFC circuit

The switch modulation strategy is a function of the AC line voltage, as shown in Fig. 4 and Fig. 5:

- $V_{\mathrm{ac}} > 0$: Q2 is closed during the positive line cycle. M2 is active boost switch and the current free-wheels through M1 delivering power to the load.
- $V_{\mathrm{ac}} < 0$: Q1 is closed during the positive line cycle. M1 is active boost switch and the current free-wheels through M2 delivering power to the load.

In both instances the gate of free-wheeling switch is modulated complementary to the active switch to reduce conduction loss.

At the transition between positive and negative AC voltage, the high-frequency bridge duty cycle must abruptly change to account for the change in active switch. The switch node voltage of the grid-frequency bridge must change between 0 V and the DC voltage. There is a risk of large current spikes should the abrupt duty-cycle change of the high-frequency switches occur asynchronous to the voltage zero crossing or while the grid-frequency bridge transitions [1]. Furthermore, it is difficult to control the inductor current while the AC line voltage is close to zero. One solution is to introduce an interval around the zero-crossing transition where all switches are open [2].

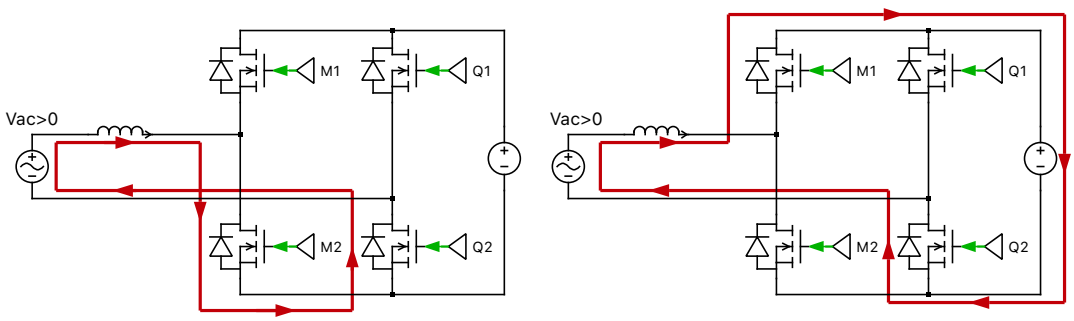

Figure 4: Totem-pole PFC modulation during positive line-cycle

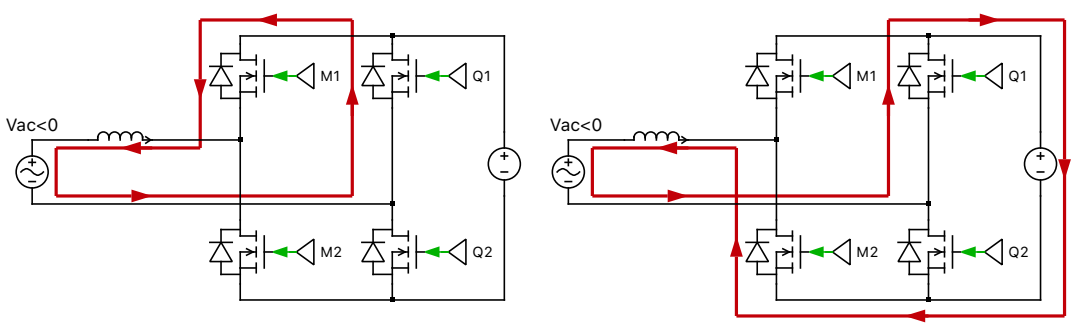

Figure 5: Totem-pole PFC modulation during negative line-cycle

Dual active bridge and battery model

The second stage of the OBC interfaces the DC output of the totem-pole PFC with a lithium-ion EV battery. The DAB is one suitable topology for the DC/DC stage to its bidirectional power flow, zero voltage switching, and inherent galvanic isolation [3].

A single phase shift (SPS) modulator drives the DAB switches. In SPS modulation the DAB primary and secondary switches have a constant duty cycle of 0.5 pu. The phase shift between the primary and secondary PWM carriers controls the output current. When the primary bridge leads the secondary bridge power is transferred to the battery. When the primary bridge lags the secondary grid power transfer is from the battery, enabling vehicle to grid applications.

The relationship between the secondary side DC output current and phase-shift angle φ is given by [4]:

$$I_{LV} = \frac{nV_{\mathrm{HV}}}{2\pi^2 f_{\mathrm{sw}} L_{\mathrm{tot}}} (\pi - \mid \varphi \mid) \varphi$$

The direct inversion of the above equation is then used to calculate the required phase-shift to apply the commanded reference current I^*_{LV}. This relationship can be written as:

$$\varphi^* = \frac{\pi}{2} \left(1 - \sqrt{1 - \frac{8 f_{\mathrm{sw}} L_{\mathrm{tot}} \mid I^*_{\mathrm{LV}} \mid}{nV_{\mathrm{HV}}}} \right) \mathrm{sign}(I^*_{\mathrm{LV}})$$

Differentiating the I_{LV} expression calculates the small signal behavior of the DAB output current, assuming the modulator delay is negligible near the frequencies of interest:

$$G_{\tilde{\mathrm{I}}_{\mathrm{LV}}/\tilde{\varphi}}(s) = \frac{nV_{\mathrm{HV}}}{2\pi^2 f_{\mathrm{sw}} L_{\mathrm{tot}}} (\pi - 2\varphi^*) = K_{\mathrm{DAB,i}}$$

Note the above equations are independent of the DAB output voltage V_{LV}.

The DAB output connects a simple battery model which consists of a DC capacitance, constant resistance, and variable internal voltage that is a function of the batteries state-of-charge (SOC). The simplified battery model represents identical charge and discharge curves under constant current conditions and is attractive due to its low computational complexity and straightforward parameterization. Several physical effects are not captured in the simplified model including polarization, SOC and temperature dependence of internal impedance, self discharge, and memory effect, among others [5].

Fig. 6 shows the equivalent circuit for the DAB output current and simple battery model. The output capacitance and battery resistance form a low-pass filter, although in practice other filtering techniques may be used. The small signal transfer function for the battery current as function of phase angle is based on Fig. 6:

$$\frac{\tilde{I_{\mathrm{bat}}}}{\tilde{\varphi}} = \frac{K_{\mathrm{DAB,i}}}{1 + sR_{\mathrm{s}}C_{\mathrm{bat}}}$$

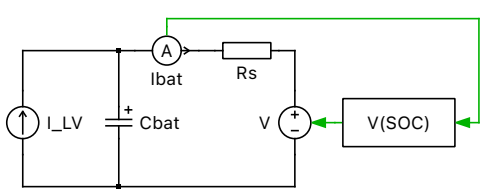

Figure 6: Equivalent circuit for battery control

RT Box configuration

The plant simulation model is configured to run on the RT Box 2 and RT Box 3 real-time platform. The model has several optimizations for high-fidelity real-time simulation. The simulation model utilizes the FPGA for electrical circuit simulation and all three available real-time cores of the RT Box 2 and 3 hardware, reducing the computational burden on any single core.

The real-time simulation model uses two different modeling paradigms for each power converter. The totem-pole PFC electrical model executes on the RT Box's FPGA with sub-cycle average switch models [6]. The Model Settings block associated with the PFC stage explicitly assigns the electrical simulation model to the FPGA resource, allowing for electrical model step-sizes of 400 nsec or less. The electrical model executing on the FPGA must be associated with the CPU core to manage inputs and outputs to the FPGA model. For example, the CPU performs the trigonometric calculations for the AC voltage source, which is then an input to the FPGA model.

The DAB converter model executes fully on the CPU with a sub-step event based model [7]. Sub-step event models are preferred for high-frequency converters where the direction of the phase current changes frequently, even several times within a model step. Given the initial system states and relative on-times of the gate switching signals, the sub-step event DAB model determines the sequence

of switching states within a model step and computes an average output current. The net result is an accurate simulation even with model step sizes that are relatively large compared to the switching period of the converter.

The electrical model is split at the DC link to electrically decouple the two-stage power converter, allowing assignment to separate processing cores. Refer to the “Modular Multilevel Converter” demo model of the RT Box Target Support Package for additional background on model splitting.

The Task component from the PLECS library assigns the blocks enclosed within the component frame to a specified task. For multi-core targets each task must be associated with a CPU core. Blocks not enclosed by a Task Frame are assigned to the default task. The **Scheduling** tab of the **Coder + Coder options...** dialog configures CPU assignments and sets the sample times for all tasks. For multi-core execution on the RT Box the **Tasking mode** must be set to `multi-tasking` to configure the processing core and sample time for each task.

In this RT Box model all CPU cores have the same execution time of 1.8µs. The CPU core assignment is:

- FPGA: The totem-pole PFC converter and DC bus capacitance
- Core 0: Default task used only to display synchronized scope data from all cores
- Core 1: Totem-pole PFC components not modeled on the FPGA
- Core 2: The DAB and battery model

2.2 Controller

The OBC control system is shown in Fig 7. Task Frames assign the PFC controls to CPU1 and the DAB controls on CPU2.

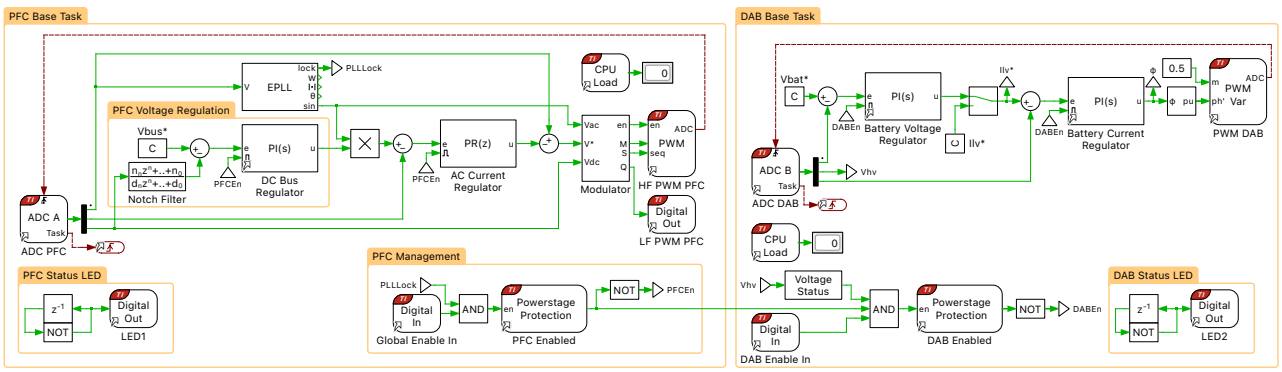

Figure 7: Schematic of the on-board charger system

Dual-core and multi-tasking configuration

The PLECS Coder and the TI C2000 Target Support Package supports dual-core and multi-tasking code generation. The dual-core TI C2000 MCUs contain two independent processing cores, referred to as CPU1 and CPU2. The processors execute independently so time-critical tasks execute in parallel. The two cores have access to the same set of underlying peripherals and a shared memory space for IPC.

Dual-core targets are configured using a single subsystem with code generation enabled. The single subsystem approach facilitates offline simulation of the entire control system. Core assignment for the TI C2000 MCU’s is similar to core assignment for the RT Box, described in Section 2.1. Task components from the PLECS library assign portions of the model to a task. Each task is assigned a CPU core, configured in the **Scheduling** tab of the **Coder + Coder options...** dialog. For dual-core execution the **Tasking mode** must be set to `multi-tasking` to configure the processing core and sample

time for each task. For further information, refer to the "Code Generation" section in the PLECS User Manual [8].

As the CPU cores are independent of each other, each CPU core executes its own rate monotonic scheduler. In a multi-CPU system each CPU requires an explicit Control Task Trigger. The Control Task Trigger defines the Base Task scheduling for each core. If one core encounters a run-time error such as a CPU overflow, the second core will continue to execute. The sample times for all tasks on a given CPU core must be an integer multiple of the core's Base Task sample time. However, the two CPU cores do not need to have the same Base Task sample time.

The Task Frame also assigns ownership of peripheral blocks such as the ADC, PWM, and Powerstage Protection to the associated core. The Powerstage Protection block only disables PWM blocks on the same core. For example, the "PWM DAB" modulator and the "DAB Enabled" Powerstage Protection block are both assigned to CPU2. Toggling the "DAB Enable In" switch will only disable the switching signals for the DAB converter. The "DAB Enabled" block does not impact the PWM resources associated with the PFC circuit on CPU1.

Note the output of the "PFC Enabled" Powerstage Protection block is part of the input of the "DAB Enabled" Powerstage Protection block. When a signal crosses the Task Frames for tasks on different CPUs, the coder target framework configures an IPC data channel. Each IPC data channel is a double buffer with two semaphores to guarantee that simultaneous read and write operations never access the same buffer. Refer to the PLECS TI C2000 Target Support User Manual [9] for additional information on task transitions between different CPUs.

The table below lists the tasks in the OBC controller, including CPU resource and sample time assignment. The execution rate of the PFC Base Task frequency is 32.5 kHz or half the 65 kHz PFC switching frequency. The DAB Base Task executes at 50 kHz, half of the DC/DC converter's 100 kHz switching frequency.

Task name	CPU Core	Sample time (s)
PFC Base Task	1	$2/F_{\mathrm{sw,pfc}}$
PFC Voltage Regulation	1	$32/F_{\mathrm{sw,pfc}}$
PFC Management	1	$320/F_{\mathrm{sw,pfc}}$
PFC LED Blink	1	0.5
DAB Base Task	2	$2/F_{\mathrm{sw,dab}}$
DAB LED Blink	2	0.5

PFC Controls

The PFC controls, shown on the left side of Fig 7, use a multi-stage control architecture. The PFC controller also includes a basic interlocking scheme for start-up, described further in Section 2.2.

The PFC controller's outer loop regulates the DC bus voltage with a proportional integral (PI) regulator. The bandwidth of the voltage control loop is set well below the grid frequency to reduce the control response to the DC bus voltage oscillations at twice the AC line frequency. A notch filter further reduces the 2nd harmonic oscillations in the measured DC bus voltage, reducing the input current harmonic distortion. The notch filter implementation is a Discrete Transfer Function block from the PLECS library. The discrete transfer function parameter calculations use the bilinear transform with pre-warping to preserve the notch frequency.

The inner loop regulates the AC current with a proportional resonant (PR) controller. The voltage controller's output sets the desired peak current. An Enhanced PLL with DC offset correction tracks the AC grid voltage [10]. The peak current reference is multiplied by the PLL output, resulting in an AC current reference in phase with the grid voltage. The PR controller implementation is configurable,

with two continuous implementations and a discrete implementation based on the Direct Form-I digital compensator.

The "Modulator" subsystem accounts for the polarity of the input voltage when generating the PWM signals, as described in Section 2.1. The grid voltage is divided into a positive region, negative region, and zero-crossing region. The duty cycle and switching sequence change as a function of the sensed region.

During the zero-crossing interval the PWM outputs are disabled by setting the "En" input to the PWM modulator as a logical low. A Digital Out block "LF PWM PFC" generates the grid-frequency gating signals. Compared to using a PWM block, this approach results in a lower processing load and bypasses the difficulties of enabling and disabling a TI C2000 PWM channel with dead-time at full duty cycle. The "PFCEn" signal sets the Digital Out signals to logical low when the Powerstage protection block is stage is disabled.

DAB Controls

The DAB controller implements a constant-current constant-voltage (CCCV) battery charging scheme. The DAB controller outer loop regulates the battery voltage and the inner loop the battery current. In the constant-current regime the voltage output is limited to the maximum charging current. Toggling the Manual Switch component allows one to set the battery current set point directly.

The DAB controller design uses the small signal expressions derived in Section 2.1 and pole-zero cancellation to achieve the desired control bandwidth. With the current controller bandwidth defined as $\omega_{\mathrm{bat,i}}$:

$$G_{\mathrm{i}\varphi}(s) = G_{\mathrm{ibat,pi}}(s) * G_{\tilde{\mathrm{I}_{\mathrm{LV}}}/\tilde{\varphi}}(s) = K_{\mathrm{ibat,p}} \frac{K_{\mathrm{DAB,i}}}{1 + sR_{\mathrm{s}}C_{\mathrm{bat}}} \frac{(s + K_{\mathrm{ibat,i}}/K_{\mathrm{ibat,p}})}{s}$$

$$K_{\mathrm{ibat,p}} = \frac{\omega_{\mathrm{bat,i}} R_{\mathrm{s}} C_{\mathrm{bat}}}{K_{\mathrm{DAB,i}}}$$

$$K_{\mathrm{ibat,i}} = \frac{K_{\mathrm{ibat,p}}}{R_{\mathrm{s}} C_{\mathrm{bat}}}$$

The design for the voltage controller uses a similar approach, with the closed-loop bandwidth of the current regulator defining the real-pole in the battery current to battery voltage transfer function.

Start-up Scheme

The OBC controller includes a start-up scheme to ensure the PFC control loops activate before the DAB controls.

The "PFC Enabled" Powerstage Protection block enables the PFC control loops and PWM modulation signals. The conditions to activate the PFC PWM outputs are:

- The PLL has correctly locked onto the grid frequency. The PLL is considered locked once the voltage threshold exceeds a minimum threshold and the PLL frequency is within a nominal range.
- The "Global Enable In" Digital In transitions to a logial high state.The "Global Enable In" is associated with the DI-29 DIP switch on the TI C2000 LaunchPad and ControlCard interface boards.

The DAB PWM outputs activate once the "DAB Enabled" Powerstage Protection block input transitions to a logical high. Three conditions must be met to activate the DAB PWM outputs:

- The PFC Powerstage is Active.
- The DC bus voltage rises above a 90 % the nominal bus voltage for one AC cycle.
- The "DAB Enable In" GPIO is logical high. The "DAB Enable In" signal depends on the interface board's DI-30 DIP switch position.

3 Simulation

3.1 PLECS Offline Simulation

Run the model as provided from **Simulation + Start**, to observe the results from the offline simulation. Fig. 8 shows the results from the PFC Scope in the "Controller" subsystem.

At 5 msec the global enable input is activated. The PLL lock condition is met at 25 msec and the PFC begins to regulate the DC bus voltage. With the DC bus voltage initialized at 400 V, the DAB control loops activate at 30 msec, delivering rated current to the batter. The increase in battery current results in a DC bus voltage reduction.

The PFC voltage controller increases the peak current reference $I_{\mathrm{ref,pk}}$. The PR current regulator closely tracks the reference set point, restoring the bus voltage to the 400 V reference. Note the 100 Hz oscillations in the V_{DC} measurement are attenuated in the $V_{\mathrm{DC,filt}}$ measurement at the voltage controller's input.

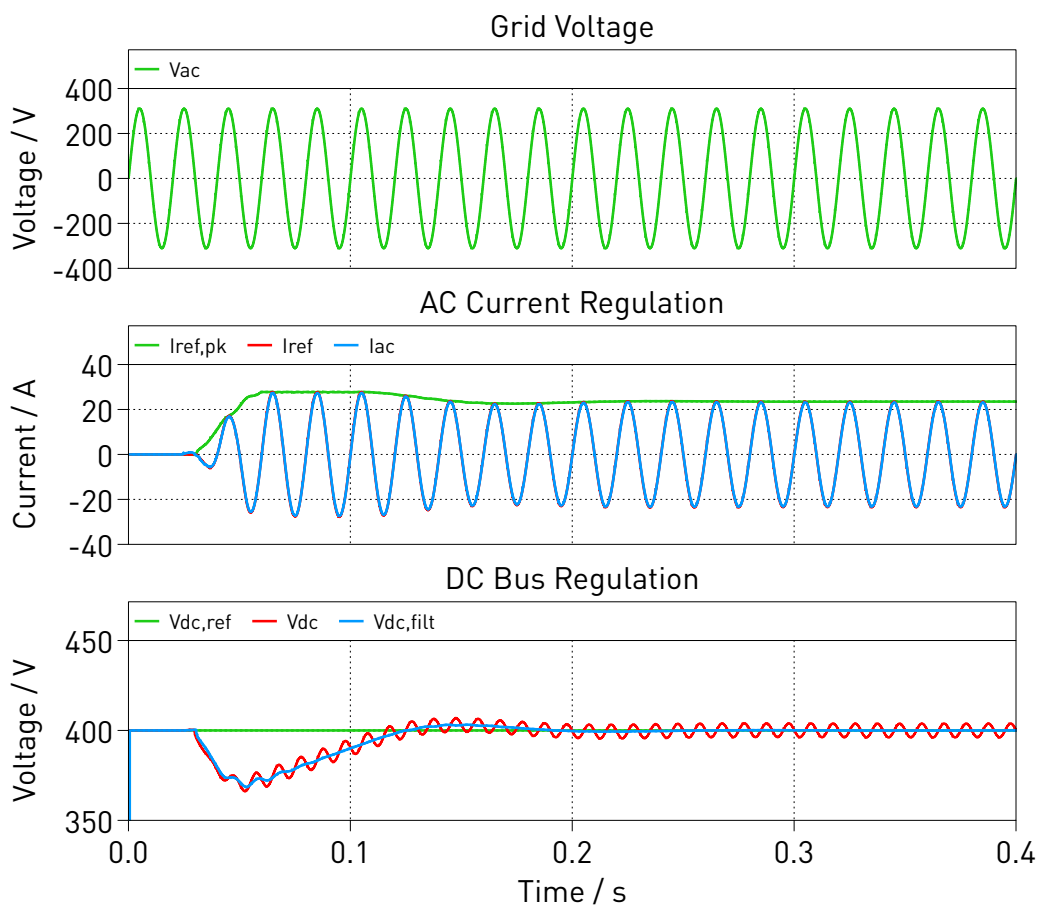

Figure 8: Offline simulation of OBC start-up

3.2 Configuring the TI C2000 Target

The "Controller" subsystem is configured to generate target specific code for dual-core TI MCUs. The model is configured by default for a TI 28379D LaunchPad, with support for the TI 28379D and TI 28388D ControlCard hardware.

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, select `LaunchPad`, `ControlCard`, or `Custom` as the **Board** type. Choosing `Custom` enables for the use of an external debug probe.
- Click **Build**.

With dual-core programming, MCU application for each core is compiled and flashed sequentially. Programming may take several minutes. If programmed correctly, the LEDs associated with “DO_DSP_LED1” and “DO_DSP_LED2” (listed in the model initialization commands) should blink.

Please note that the I/O configuration of all the peripheral blocks (ADC, PWM) are configured by default to the TI 28379D LaunchPad. For other targets the I/O configuration has to be adapted. To choose a TI 28388D ControlCard, change the **Target** in the **Coder Options + Target** menu. To choose a TI 28379D ControlCard, uncomment the relevant `board_type` line for the MCU in the **Model initialization commands** window, accessed from the **Simulation Parameters... + Initializations** tab in the **Simulation**.

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

3.3 Configuring the PLECS RT Box

Follow the instructions below to run a real-time model on the RT Box 2 or 3 hardware.

- From the **System** tab of the **Coder + Coder options...** window, select “On-board Charger”. 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.

3.4 Executing a Closed-Loop HIL Test

Toggle the switch “DI-29” on the RT Box Interface board from low to high to enable the PFC regulation. When PFC regulation is enabled, the LED corresponding to “DO-29” of the LaunchPad interface board should turn on. Toggling “DI-30” will enable or disable the DAB controller, with a corresponding change in state of the “DI-30” LED on the RT Box interface board. Observe the real-time waveforms in the Scope of the “On-board Charger” subsystem.

In order to observe the control signals of the MCU, 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.

Similarly, connect to the RT Box external mode following the instructions in Section 3.3.

Note The RT Box FPGA core updates faster than the RT Box external mode sample rate. Inspecting the signals with an oscilloscope will show the actual model fidelity for FPGA based simulations.

Fig. 9 shows the RT Box results during an OBC start-up event. The "PFC Waveforms" Scope in the "On-board Charger" was configured to trigger on the falling-edge of the DC bus voltage. The start-up event was triggered setting "DI-30" high on the RT Box Interface Board and then toggling "Global Enable In" signal associated with "DI-29". The AC current and DC bus voltage transient response aligns with offline simulation in Fig. 8. The behavior of the grid-frequency switches of Q1 and Q2 is also apparent in the figure.

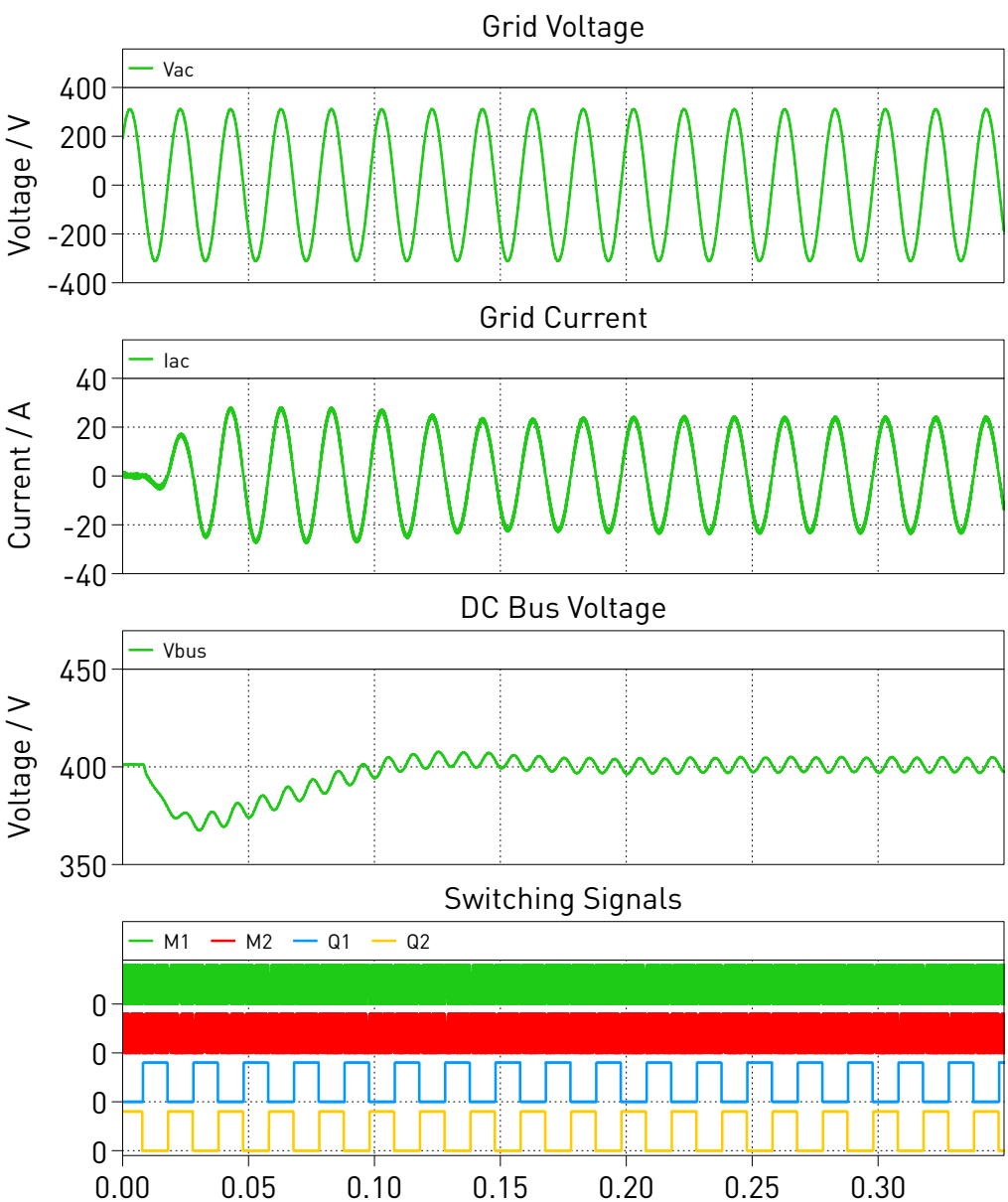

Figure 9: RT Box data-capture of OBC start-up

While connected to the external mode one can also adjust reference values, according to their config-

uration in the **Parameter Inlining** tab of the **Coder Options** menu. Fig. 10 shows the RT Box data capture when the battery voltage controller set point is reduced by 5 % with an SOC of 75 %. The battery voltage reaches the desired set point and power is delivered from the battery to the DC bus, causing an overshoot in the DC bus voltage until corrected by the PFC control system. The DAB LV current and output power are not strongly influenced by the DC bus transient.

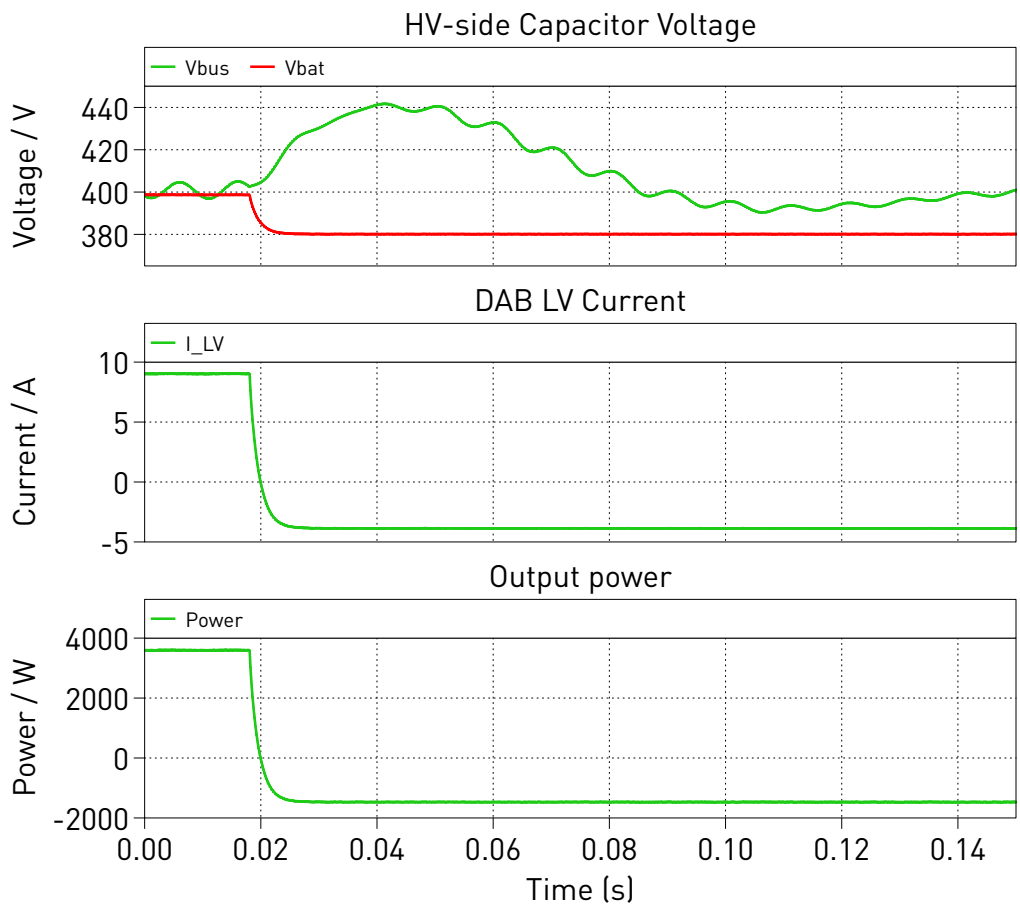

Figure 10: The figure caption of the simulation results

4 Conclusion

This model demonstrates the use of a dual-core MCU to control an EV on-board charger. PLECS generates the dual-core MCU control code directly from the simulation model. The charger's power stage of a 65 kHz totem-pole PFC and a 100 kHz DAB executes on the RT Box 2 and 3 for HIL testing of the OBC controller. The real-time model configuration reduces the computational load on any one processing core by using three RT Box CPU cores and the FPGA solver.

The OBC controller has the PFC controls execute on one CPU core and the DAB controls on a second core. Inter-process communication between the cores is configured through simple signal connections. The PFC and DAB controls use cascaded control loops along with a simple interlocking mechanism for start-up. The offline simulation results and real-time HIL tests align closely.

References

[1] X. Gong, G. Wang and M. Bhardwaj, "6.6kW Three-Phase Interleaved Totem Pole PFC Design with 98.9% Peak Efficiency for HEV/EV Onboard Charger," *2019 IEEE Applied Power Electronics*

Conference and Exposition (APEC), Anaheim, CA, USA, 2019, pp. 2029-2034.

[2] ST Microelectronics “Getting started with the STEVAL-DPSTPFC1 3.6 kW PFC totem pole with inrush current limiter reference design” Nov, 2022.

[3] Texas Instruments “Design Guide: TIDA-010054 Bidirectional, Dual Active Bridge Reference Design for Level 3 Electric Vehicle Charging Stations”, July,2022

[4] Y. Guan, Y. Xie, Y. Wang, Y. Liang and X. Wang, "An Active Damping Strategy for Input Impedance of Bidirectional Dual Active Bridge DC–DC Converter: Modeling, Shaping, Design, and Experiment," in IEEE Transactions on Industrial Electronics, vol. 68, no. 2, pp. 1263-1274, Feb. 2021, doi: 10.1109/TIE.2020.2969126.

[5] O. Tremblay, L.-A. Dessaint, and A.-I. Dekkiche, “A generic battery model for the dynamic simulation of hybrid electric vehicles,” Vehicle Power and Propulsion Conference, 2007.

[6] J. Allmeling and N. Felderer, "Sub-cycle average models with integrated diodes for real-time simulation of power converters," 2017 IEEE Southern Power Electronics Conference (SPEC), Puerto Varas, Chile, 2017, pp. 1-6, doi: 10.1109/SPEC.2017.8333566.

[7] J. Allmeling, E. Rodriguez-Diaz and N. Felderer, "Accurate Real-Time Simulation of Converters with Frequent Current Commutation Using Sub-Step Events," 2021 IEEE Vehicle Power and Propulsion Conference (VPPC), Gijon, Spain, 2021, pp. 1-6, doi: 10.1109/VPPC53923.2021.9699142.

[8] PLECS User Manual,

URL: https://www.plexim.com/sites/default/files/plecsmanual.pdf.

[9] PLECS TI C2000 Target Support User Manual,

URL: https://www.plexim.com/sites/default/files/c2000manual.pdf.

[10] M. Karimi-Ghartemani, S. A. Khajehoddin, P. K. Jain, A. Bakhshai and M. Mojiri, "Addressing DC Component in PLL and Notch Filter Algorithms," in IEEE Transactions on Power Electronics, vol. 27, no. 1, pp. 78-86, Jan. 2012, doi: 10.1109/TPEL.2011.2158238.

Revision History:

C2000 TSP 1.7.1 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