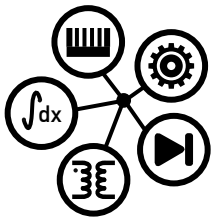

Embedded
Code Generation

DEMO MODEL

Single-Phase PV Inverter

A single-phase, single-stage, grid-connected PV inverter with embedded code generation for TI C2000 MCUs

Last updated in C2000 TSP 1.5.1

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

Single-phase PV inverters are commonly used in residential rooftop PV systems. In this example, a single-phase, single-stage, grid-connected PV inverter is modeled. The PV system includes an accurate PV string model that has a peak output power of 3 kW, as described in Section 2.1.

The control system comprises three control loops: a maximum power point (MPP) controller, a voltage controller and a current controller. The outer control loop is a MPP controller that ensures maximum power is extracted from the PV string for a given insolation level. This model uses multi-tasking feature, as described in Section 2.2.

Additionally, this demo model provides an explanation of the typical workflow of the PLECS Embedded Coder, using Texas Instruments (TI) C2000 microcontrollers (MCUs). Combined with a PLECS RT Box, the performance of the MCU can be verified directly. Section 3 provides instructions on running the real-time simulation.

The model is split into two distinct subsystems called “Plant” and “Controller”. 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 it.

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.

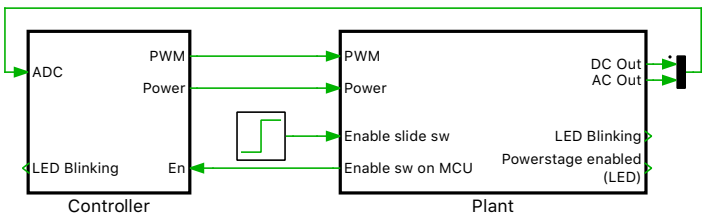

Figure 1: Top level schematic of the plant and the controller subsystems

2.1 Power Circuit

The power generation system, shown in Fig. 2, is comprised of a solar array that provides a steady-state output of approximately 380 V DC, an IGBT-based full bridge inverter, and an LCL output filter connected to a 230 V_{rms}, 50 Hz single-phase mains. The sun insolation value for the PV array is toggled between a nominal value of 1 kWhr/m^2 and a reduced insolation level of 0.7 kWhr/m^2 every two seconds.

The full bridge is implemented using the Full Bridges (Series Connected) power module component. The pulse-width modulated (PWM) switching signals are 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 input and the AC output voltage and current measurements are connected to Analog Out blocks from the PLECS RT Box library. The Analog Out pins of the RT Box are connected to an embedded controller. Therefore the voltage and current measurements are scaled and offset 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 $2.5\,\mu\mathrm{s}$.

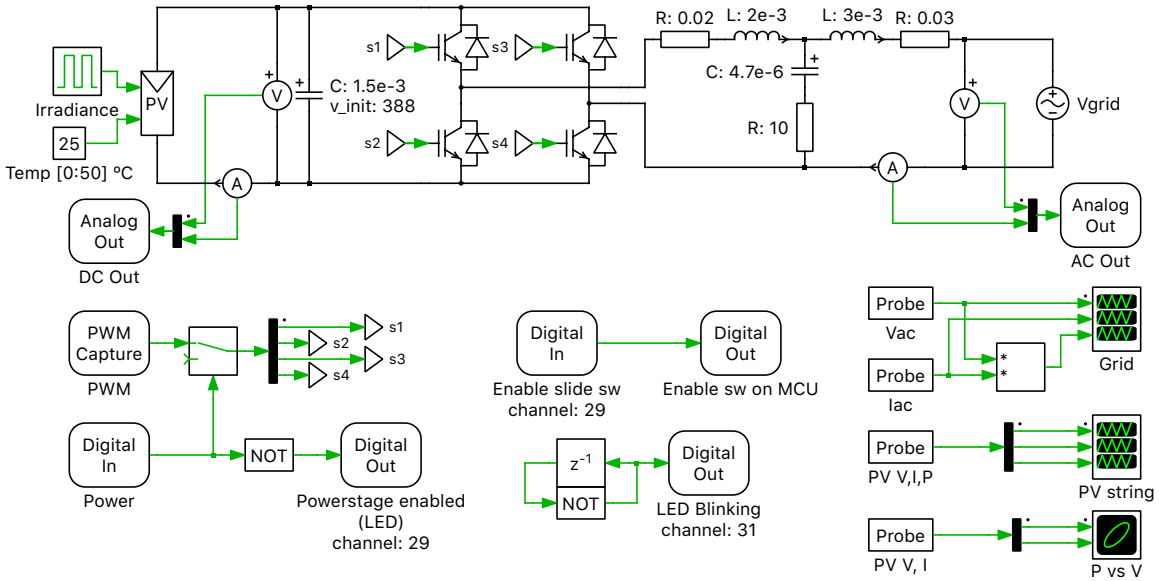

Figure 2: Power circuit of the single-phase PV inverter

PV String Model

The PV string component is based on a non-linear current source that accurately models the IV characteristic with variable inputs of insolation (sun intensity), output voltage, and temperature. Further, it can be connected in various series and parallel configurations and used as a DC source for both offline and grid-connected systems. The model is based on the Shockley diode equations for accuracy and can be used to study the interactions between a PV inverter and the supply modules. The typical output current characteristic of the PV model is shown in Fig. 3.

In this example, insolation and temperature-dependent data has been mapped for a BP365 65 W solar module, and the array comprises 22 modules connected in each string, with 2 strings connected in parallel. The current surface data, is saved in a .mat file and contains information for the characteristic IV curves for an insolation range of 0 to $1\,\mathrm{kWhr/m^2}$, voltage values between 0 and 25 V, and three temperatures of 0, 25, and $50\,^{\circ}\mathrm{C}$.

2.2 Controls

The controller subsystem is shown in Fig. 4. The control system comprises three control loops: a maximum power point (MPP) controller, a voltage controller and a current controller. The outer control loop is a MPP controller that ensures maximum power is extracted from the PV string for a given insolation level. To do this, it calculates the optimal PV terminal voltage using a MPP algorithm known as dP/dV (incremental conductance) control, implemented using a C-Script block. The voltage control loop, based on a type-2 controller, regulates the PV voltage to this optimal level by controlling the amount of current that is injected into the grid. The innermost control loop, the current controller, sets

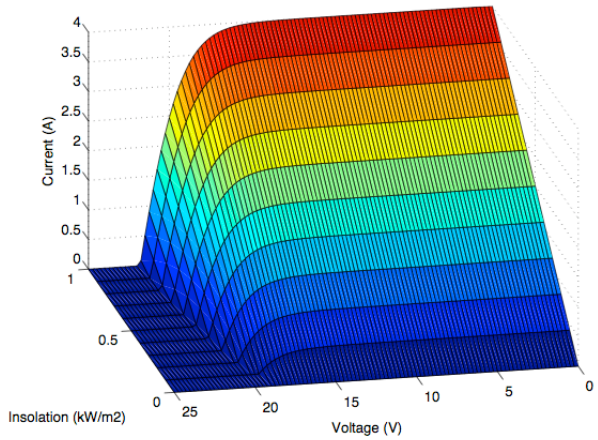

Figure 3: Typical output current characteristic of the BP365 PV module model at 25 °C.

the modulation index of the inverter such that the desired current is injected into the grid. The current controller is based on a PR controller with a resonant frequency of 50 Hz to ensure no tracking error is present. Finally, a unipolar modulation scheme is deployed with two 12.5 kHz PWM generators which are phase shifted 180 degrees relative to each other. This results in the cancellation of the harmonic component at the switching frequency in the converter output. A damping resistor is included in the LCL output filter, as seen in Fig. 2, to ensure stable operation of the current controller.

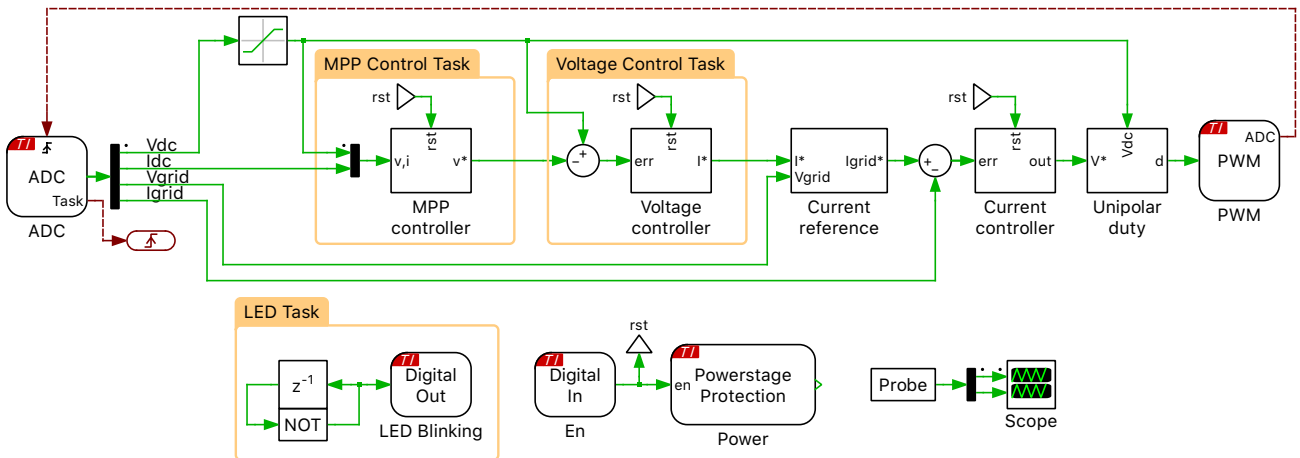

Figure 4: Controller of the single-phase PV inverter circuit

Multi-tasking code

The PLECS Coder and the TI C2000 Target Support Package allow the user to generate multi-tasking code for the TI C2000 family of MCUs. Multi-tasking code unlocks processing power for controls regulating multiple system outputs with dynamics on a range of time-scales. In this model there is a cascaded control scheme with a fast inner current control loop, followed by a PV string DC voltage control loop, and a slower outer MPP loop. Multi-tasking code is well suited for this type of control scheme.

Multi-tasking code generation is configured in the **Scheduling** tab of the **Coder + Coder options...** dialog. By changing the **Tasking mode** to `multi-tasking` and the **Task configuration** to `specify`, the sample time for each task can be configured. The base sample time is always equal to the **Discretization step size**. The **Sample time** setting for lower priority tasks must be an integer multiple of the base sample time. Up to 15 slower lower priority tasks that execute at different rates can be

specified, preserving processor time for the fastest, highest priority task in the application. For further information, refer to the "Code Generation" section in the PLECS User Manual [2].

Blocks in the PLECS schematic are assigned to lower priority tasks using the Task library component. In this model three lower priority tasks are defined in addition to the Base Task, as seen in Fig. 4. The table below lists the sample times of the lower priority tasks, with respect to the Base Task. The execution rate of the Base Task (fc), in this case is 25 kHz or two times the switching frequency (fs) of 12.5 kHz.

Task name	Sample time
Base Task	1/fc
Voltage Control Task	10/fc
MPP Control Task	55/fc
LED task	2500/fc

Configuring TI C2000 Target library components

The controller in Fig. 4 contains four main components from the TI C2000 Target library.

- **PWM** The gate signals are generated by the PWM component. The duty cycles are in the range of [0, 1] for generating gate signals. The **Carrier type**, **Carrier frequency** and **Blanking time** parameters can be configured from the **Main** tab of the PWM block parameters window. Please note that the input to the ePWM module generates a complementary PWM pair, i.e. ePWMxA and ePWMxB will be opposite polarity excluding the blanking time.
- **ADC** The measurements of the DC input and the AC output voltages and currents 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.
- **Control Task Trigger** In this model, the interrupt sequence of the embedded application is defined explicitly by connecting trigger signals between the PWM generator, the ADC, and the Control Task Trigger. Trigger signals are shown as red dashed lines as seen in Fig. 4. From the **Events** tab of the PWM block parameter window, the **ADC Trigger** parameter is configured as `Underflow and Overflow`. This means that the control task is triggered twice per PWM period. Figure 5 shows the corresponding PWM carrier, task trigger, and PWM outputs.

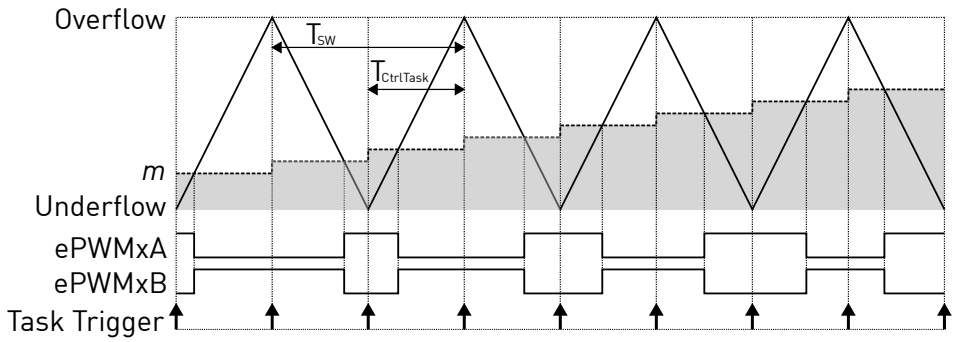

Figure 5: PWM carrier and task interrupts for PWM frequency set to half the control task frequency

- **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 an interlock, which is a safety mechanism, 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.

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 C2000 MCUs. The model is configured by default for a TI 28069 LaunchPad [4], but other targets are supported as explained later in this section. Furthermore, the model of the plant can be deployed on the PLECS RT Box for a hardware-in-the-loop (HIL) test of the generated code.

3.1 Configuring the TI C2000 Target

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 [3] 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.

Please note that the I/O configuration of all the peripheral blocks (ADC, PWM) are configured by mapping to the TI 28069 LaunchPad [4]. For a TI MCU other than the TI 28069 LaunchPad, the I/O configuration has to be adapted. In addition to the TI 28069 LaunchPad, this demo model also supports code generation for other TI C2000 MCUs, such as the TI 28377S [5], TI 28379D [6] and TI 280049C [7] LaunchPads; as well as the TI 28379D [8] and TI 28388D [9] controlCARDs. From the **Model initialization commands** window of **Simulation Parameters... + Initializations** tab from the **Simulation** menu, change the value of `type_evm`, to choose the desired target. You must also configure the corresponding **Target** in the **Coder Options** window accordingly.

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 HIL test. A typical hardware configuration is shown in Fig. 6, where the evaluation kit, a TI 28069 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.

3.3 Executing a Closed Loop HIL Test

With both the TI C2000 MCU and the PLECS RT Box programmed you can now use the switches on the RT Box LaunchPad interface board to enable the controller and PWM signals, as well as connect to the MCU and RT Box to observe real-time waveforms.

To enable the MCU, toggle the switch “DI-29” on the RT Box LaunchPad Interface board from low to high 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 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 autrotriggering** to observe the test results in the “Controller” subsystem Scope.

Similarly, connect to the RT Box external mode following the instructions in Section 3.2 . The simulation results in Fig. 7 compares the offline and real-time results of the grid during a step change in insolation at 2 seconds from 1 kWhr/m^{2}to 0.7 kWhr/m^2. The figure shows the sinusoidal mains voltage, injected sinusoidal grid current, and resulting power produced onto the grid. Fig. 8 shows the voltage, current, and power resulting from the PV array over a 4 second period. After the initial transient during the startup of the offline model, the offline and real-time results are in close agreement. An XY Plot provides the power vs. voltage characteristic and the MPP controller’s influence on the output power from the array.

The values of insolation and PV panel temperature can be changed on the fly, in real-time, since these components have been added to the "Exceptions" list found in the **Parameter Inlining** tab of the **Coder options...** window, prior to building the model.

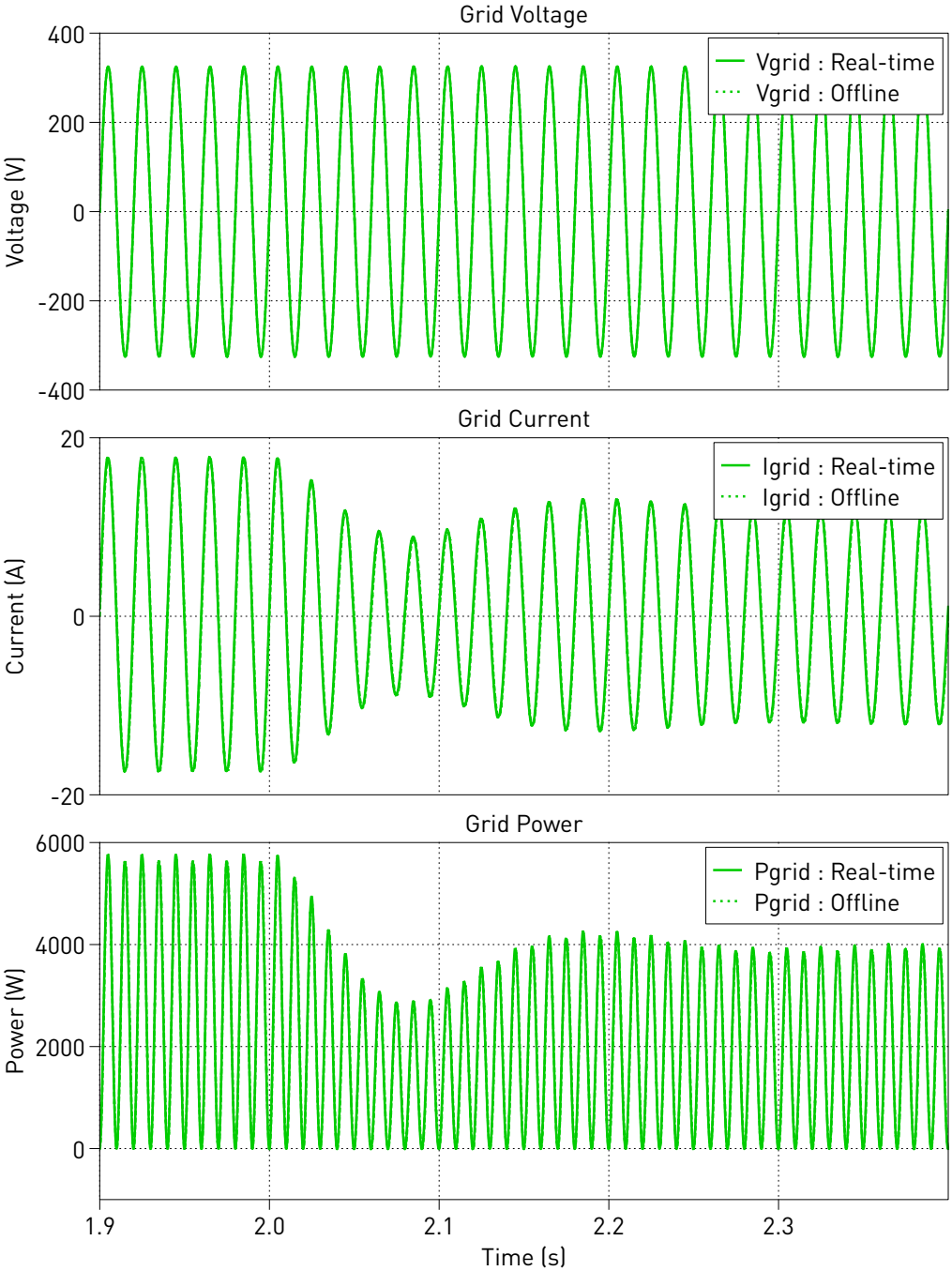

Figure 7: Grid measurements in real-time using the PLECS RT Box 1

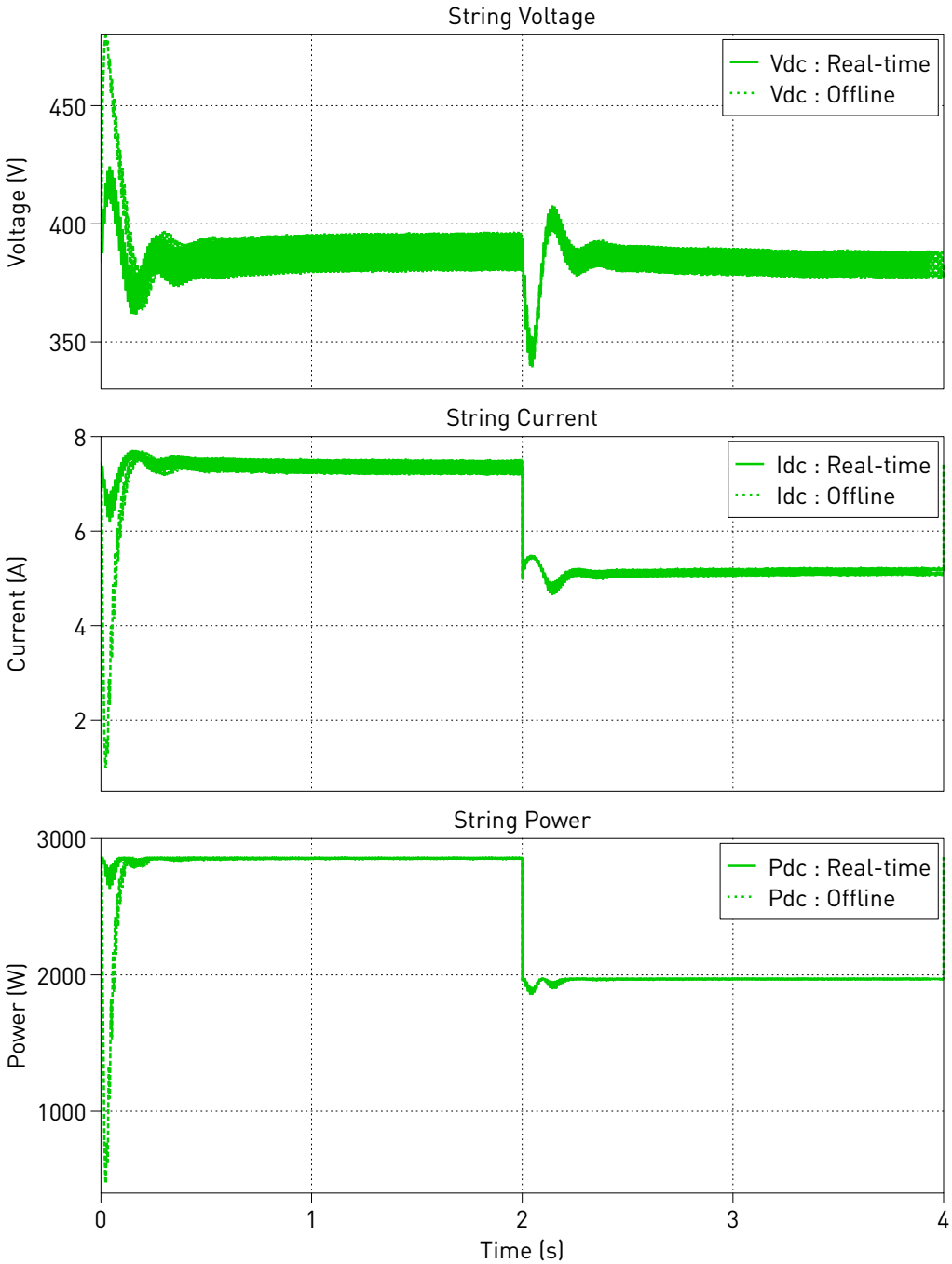

Figure 8: PV String inverter measurements in real-time using the PLECS RT Box 1

4 Conclusion

This model demonstrates a single-phase PV inverters are commonly used in residential rooftop PV systems, with a control system 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 also demonstrates the **Scheduling** feature, using which multi-tasking code can be generated. Multi-tasking code unlocks processing power for control systems that regulate multiple system outputs with dynamics on a range of time-scales. With the Task library component, 15 additional tasks at different rates can be executed, preserving processor time for the fastest, highest priority control task (Base task).

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] PLECS User Manual,
URL: https://www.plexim.com/sites/default/files/plecsmanual.pdf.

[3] PLECS TI C2000 Target Support User Manual,
URL: https://www.plexim.com/download/documentation.

[4] TI C2000 Piccolo MCU F28069M LaunchPad development kit,
URL: http://www.ti.com/tool/LAUNCHXL-F28069M.

[5] TI C2000 Delfino MCU F28377S LaunchPad development kit,
URL: https://www.ti.com/lit/pdf/sprui25.

[6] TI C2000 Delfino MCU F28379D LaunchPad development kit,
URL: http://www.ti.com/tool/LAUNCHXL-F28379D.

[7] TI C2000 Piccolo MCU F280049C LaunchPad development kit,
URL: http://www.ti.com/tool/LAUNCHXL-F280049C.

[8] TI C2000 F28379D controlCARD development kit,
URL: https://www.ti.com/tool/TMDSCNCD28379D.

[9] TI C2000 F28388D controlCARD evaluation module,
URL: https://www.ti.com/tool/TMDSCNCD28388D.

Revision History:

C2000 TSP 1.2	First release
C2000 TSP 1.4.5	Updated the web links
C2000 TSP 1.5.1	Added support for 28388D and 28379D controlCARD targets; Minimized the usage of double-precision math in the controller

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