Embedded Code Generation

DEMO MODEL

Current Source Inverter

Current source inverter with resistive load and open-loop controller using embedded code generation from TI C2000 MCUs

Last updated in C2000 TSP 1.8.1

1 Overview

A current source inverter (CSI) is typically used for high power drives because it can withstand high currents without a short circuit condition and exhibits low $\mathrm{d}v/\mathrm{d}t$ voltage over the stator windings of a permanent magnet synchronous machine (PMSM) [1] [2]. The control signal sequence of CSIs using current-type Space Vector Pulse-Width Modulation (SVPWM) is comprehensive and challenging to debug directly on real hardware targets especially when they operate at high power levels. HIL simulations can emulate the CSI operation in real time and present a virtual target for the control unit. This way the CSI control and special PWM peripheral configuration can be tested in a safe environment.

The challenges of creating a real-time capable simulation model of the CSI are the following:

- **Number of switching components:** If there are n independent switches in a converter, the number of state-space descriptions grows exponentially with 2^n [3]. As every real-time simulation platform offers limited memory to store these computed state-space matrices, this imposes a hard limit on how big topology can be simulated within a certain fixed step size.

- **Simulation accuracy:** If the forced-commutated power converters are simulated and the controlling PWM signals are sampled once per simulation step, the simulation may become inaccurate as the pulses do not naturally coincide with the simulation steps. This limitation can be overcome by using a concept called sub-cycle averaging [4] [5] which takes advantage of hybrid converter models instead of individual ideal switches. However a CSI can not be implemented with any of the existing power modules in the PLECS library, because it exhibits current source behavior on the DC side and voltage source behavior on the phase side. This is the opposite to the conventional voltage-source half bridge (see Fig. 1 in [5]).

- **Execution Time:** As previously mentioned in "Number of switching components", the bigger the number of independent switches in a model, the more time the real-time calculation consumes, therefore bigger the fixed step size. However the bigger the step size, the lower the spectral resolution on the switching waveforms. To improve the performance of the CSI model running on real-time platform, a three-phase CSI is modeled as one power module instead of modeling each phase leg separately.

This demo model introduces a power module for the CSI which is deployed in real-time on an PLECS RT Box. The controller and modulator is implemented on a Texas Instruments C2000 MCU, with the embedded code directly generated from this PLECS demo model using the PLECS TI C2000 Target Support Package.

Note This model contains model initialization commands that are accessible from:
PLECS Standalone: The menu **Simulation + Simulation Parameters... + Initializations**
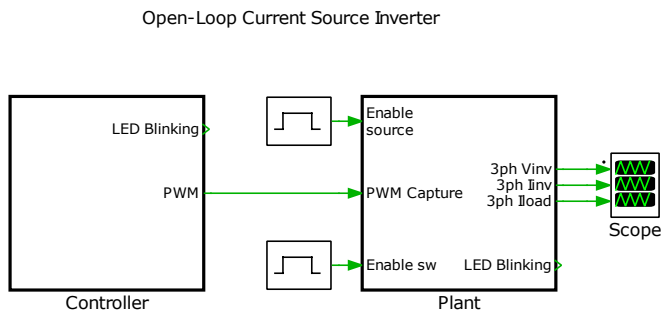*PLECS Blockset:* Right click in the **Simulink model window + Model Properties + Callbacks + InitFcn***

Open-Loop Current Source Inverter

Figure 1: Top level schematic of the current source inverter demo model

---

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 code for a subsystem via the PLECS Coder. The two enable/disable blocks on the root level implement the signals to enable and disable the DC current source and switching action of the converter. The same action can be achieved in the real-time setup of the model using DIP switches on the launchpad interface board, see section 3.

2.1 Plant

To focus on the performance of the CSI with current-type SVPWM modulation, a simplified application example of a CSI with RL-load in Fig. 2 is utilized. The electrical specification is listed in Tab. 1. The basic operation of the CSI consists of driving the switches in such a way so that the fundamentals of the phase output currents are changing in sinusoidal fashion. The most commonly used modulation strategy is the current-type SVPWM.
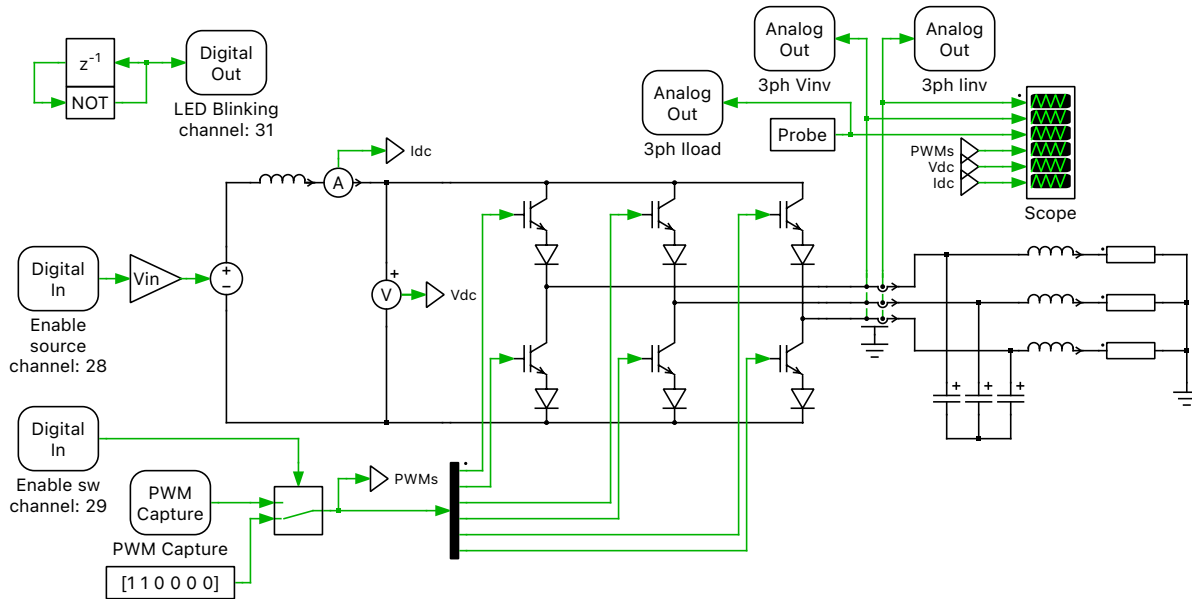

Figure 2: Power circuit of the current source inverter with symmetrical RL-load

Table 1: CSI system specification

V_{in}	I_{in}	L_{in}	C_{f}	L	R	P_{o}	f_{sw}
400 V	25 A	2 mH	4.8 μF	250 μH	16 Ω	10 kW	20 kHz

CSI Power Module

This section presents the analysis of the power module for a CSI (Fig. 3). The structure of the power module is shown in Fig. 4 with the grey area indicating the computation of the numerical model. The left side of Fig. 4 connects to an external dc-current source with current flowing into the DC+ terminal and out of DC- terminal. The DC side current shows an inductive or current source behavior so that it can be measured with an Ammeter (i_{dc} in Fig. 4). Since each phase leg has an IGBT in series with a
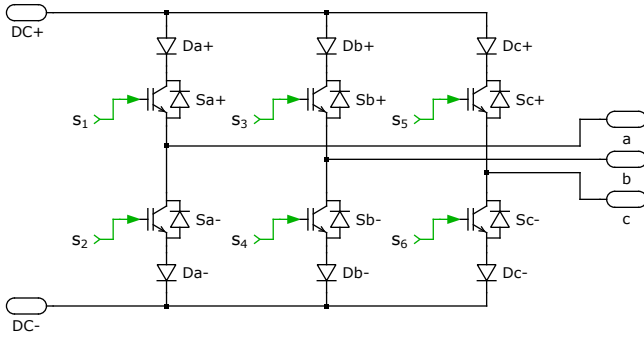
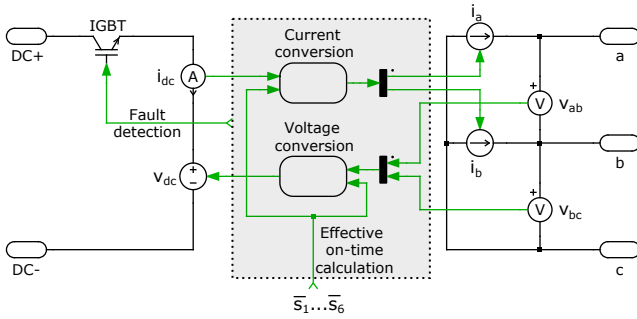**Figure 3: Switched model of a current source inverter**

Figure 4: Current source inverter power module

diode, the current direction is uniquely constrained in each phase leg. Depending on which of the 12 sectors the instantaneous phase voltage space vectors reside in, the switching states of the 6 diodes can be determined. Together with the time-averaged switching signals ($\bar{s}_1$ to $\bar{s}_6$) the effective on-times of the six phase leg currents ($\bar{s}_1^*$ to $\bar{s}_6^*$) can be derived. Including the natural commutation of the diodes inside the CSI power module logic means that the code generation process of the overall system model can be simplified.

On the right side the inverter currents of phase A and B are modeled as Controlled Current Sources (i_a and i_b in Fig. 4). The values can be calculated from the following equations that depend on the DC side measured current and the effective on-time of each phase leg:

$$i_a = i_{dc} \cdot (\bar{s}_1^* - \bar{s}_2^*)$$

$$i_b = i_{dc} \cdot (\bar{s}_3^* - \bar{s}_4^*)$$

Note that the classical CSI topology does not have neutral point connection. Because the three star-connected phase currents sum up to 0 one only needs two current sources to completely determine all phase currents. Using a third current source otherwise results in an "inconsistent source condition".

The dc-side voltage is modeled as a Controlled Voltage Source (v_{dc}) based on the measured ac-side phase-to-phase voltages (v_{ab} and v_{bc}) and the effective on-time $\bar{s}_1^*$ to $\bar{s}_6^*$, following the equation:

$$v_{dc} = \frac{1}{3} \cdot ((\bar{s}_1^* - \bar{s}_2^* + \bar{s}_4^* - \bar{s}_3^*) \cdot v_{ab} + (\bar{s}_3^* - \bar{s}_4^* + \bar{s}_6^* - \bar{s}_5^*) \cdot v_{bc} - (\bar{s}_5^* - \bar{s}_6^* + \bar{s}_2^* - \bar{s}_1^*) \cdot (v_{ab} + v_{bc}))$$

The fault detection scheme is also considered in the CSI power module. The IGBT (see Fig. 4) in series with the external dc-current source turns off when there is no closed path to sustain the current. This calculation is implemented inside the grey area of Fig. 4 based on the time-averaged switching signals ($\bar{s}_1$ to $\bar{s}_6$). This means that:

$$\bar{s}_1 + \bar{s}_3 + \bar{s}_5 > 1$$

$$\bar{s}_2 + \bar{s}_4 + \bar{s}_6 > 1$$

must be satisfied at all times during a simulation, otherwise either the desktop PLECS simulation or the RT Box will stop and throw an error.

2.2 Controller

The "Controller" subsystem is shown in Fig. 5. Since the inverter is with passive RL-load, a random PLL angle for the inverter is generated. Next, inverter current references in the abc domain are generated, and fed to the CSI modulator.
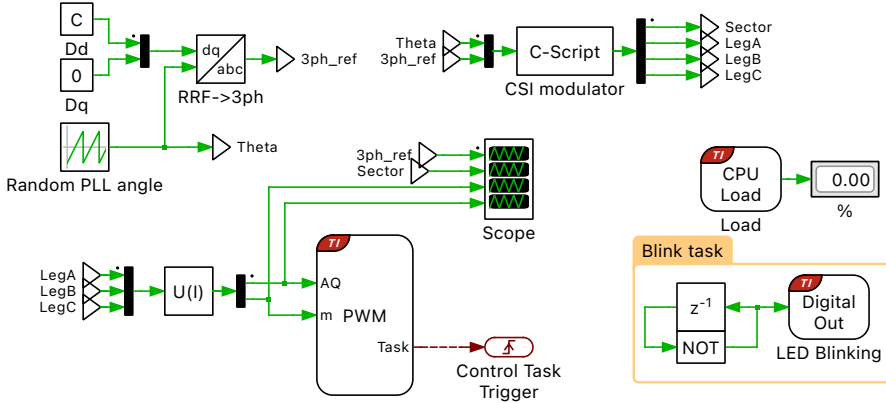

Figure 5: Open-loop controller model of the CSI with RL-load

The CSI modulator is implemented in a C-Script block to calculate the PWM relative comparator value (in p.u.) for each switch together with its action qualifier value.

These signals are passed to the PWM block. In this PWM block's **Output** tab, the **mode** is Single output (channel A) with Active state as logic '1' for each of the 6 PWM modules. **Sequence** is chosen to be Variable AQ (expert mode).

The **AQ** inport of the PWM block takes the incoming varying value and writes to the 16-bit **AQCTLA** register of the EPWM module on the MCU, in order to directly control the PWM action (Set or Clear) to the compare events occuring at the comparator value. Below are two summary tables of the values for important bits inside the AQCTLA register for the PWM generation of the CSI.

Table 2: AQCTLA register bit values for positive PWM logic

Bit position	7-6	5-4	3-2	1-0
Functionality	CAD = '10' (Set)	CAU = '01' (Clear)	PRD = '00' (Do nothing)	ZRO = '00' (Do nothing)

Table 3: AQCTLA register bit values for negative PWM logic

Bit position	7-6	5-4	3-2	1-0
Functionality	CAD = '01' (Clear)	CAU = '10' (Set)	PRD = '00' (Do nothing)	ZRO = '00' (Do nothing)

Besides, the **Dead time** field of the PWM block is 0, because the live-time between switches needed in this type of CSI is ensured in the modulator C-Script by manipulating directly the comparator value.

CSI Modulation Scheme

The modulation scheme of CSI is very similar to that of a current source rectifier (also known as buck-type rectifier) using the current-type SVPWM method. The only difference is that the buck-type rectifier

can perform the freewheeling state through an external freewheeling diode, however the CSI needs to close both the upper and lower switch in one phase in order to freewheel the dc current.

Once the dwelling time for each switching state is decided, there are various ways to arrange the sequence of the switching states inside one switching period. The symmetrical modulation method for a buck-type rectifier is thoroughly analyzed in Fig. 4 "Method 1" of [6]. This modulation method exhibits smaller filter capacitor voltage ripples according to [6] and [7], thus it is chosen as the modulation sequence implemented for the CSI in this demo. Fig. 6 depicts the gating signals for 4 active switches in sector 1 (where phase-to-neutral voltage $v_a > 0 > v_b > v_c$). One can see that S_{a+} and S_{b+}, S_{b-} and S_{c-} are having the exact opposite PWM logic at the same comparator value. Note that the signals that are always off during sector 1 are not shown here. Fig. 7 shows the corresponding modulated inverter
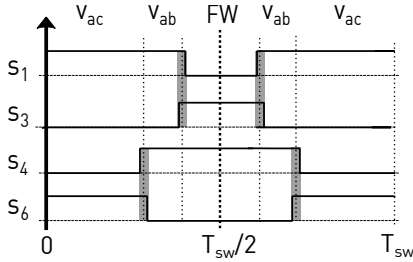

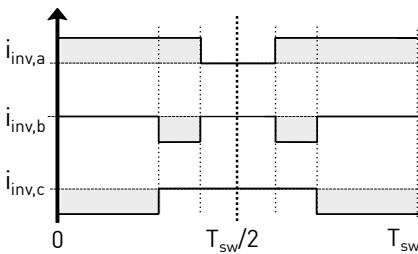**Figure 6: Switching signal sequence of the CSI under sector 1**

Figure 7: Modulated three-phase inverter output currents before LC filter

output currents at three-phase terminals at sector 1. These PWM-shaped inverter currents are later filtered through LC filter to get the switching-period averaged current values. Therefore the resistive load currents have the three-phase sinusoidal fashion.

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 default I/O configuration of all the peripheral blocks (ADC, PWM etc.) supports the TI 28379D [8] LaunchPads, and the TI 28388D [9] controlCARD.

Additionally, the demo model allows for code generation for the TI 28377S [10] LaunchPad, as well as the TI 28379D [11] controlCARD. To configure this, go to the **Model initialization commands** window from the **Simulation + Simulation Parameters... + Initializations** 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.

A typical HIL configuration is shown in Fig. 8, where the evaluation kit, a TI 28379D LaunchPad (the red board), is connected an RT Box via an RT Box LaunchPad Interface (the green board).

3.1 Build steps and enable operation

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** tab, select the desired **Build type**.
- Then, to Build and program the MCU directly from PLECS, choose Run from RAM as the **Build configuration** to program the MCU. Then select LaunchPad as the **Board** type, and click **Build**.
- After building, from the **External Mode** tab of the **Coder options...** window, **Connect** to the MCU and **Activate autotriggering** to observe the test results in real-time.
- After clicking **Activate autotriggering**, in this demo model the **Trigger channel** is pre-set to [CSI modulator:1], which indicates the inverter sector number.
- **Decimation** is set to 5 here, so that one can see the real-time waveforms over one line period by the configured **Number of samples**.

If programmed correctly, the red LED "D9" on the LaunchPad should blink.

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

PLECS RT Box

TI 28379D
Launchpad
Breakout

Figure 8: Hardware setup of the HIL verification

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

- From the **System** tab of the **Coder + Coder options...** window, select "Plant" and **Build** it onto the 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.
- After clicking **Activate autotriggering**, one can set **Trigger channel** to a specific signal, so that the running waveforms on the Scope can be more stable triggered by the **Sensitivity** condition, at **Trigger level** and **Trigger delay [steps]**. In this demo model, a certain trigger condition on [R4:Resistor current] is pre-set, to make sure that real-time simulation waveforms can overlay synchronously on top of the PLECS offline simulation.

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 capturing active PWMs from the MCU. Toggling the switch "DI-29" from "High" back to "Low" should enter safe off-state where only phase A upper and lower legs are on and all the other switches are off.
- Toggle the switch "DI-28" on the RT Box LaunchPad Interface board from "Low" to "High" to enable the DC voltage source in the "Plant" subsystem to the nominal input voltage. Toggling the switch "DI-28" from "High" back to "Low" should disable the DC voltage source.
- Toggle first "DI-29" and then "DI-28" to "High" will enable the CSI in normal operating mode.
- To stop the CSI operation, first toggle "DI-28" to "low", and then "DI-29" to "low".

Within the "Plant" subsystem running on the RT Box, the simulated voltages and currents are proportionally converted into analog signals, and delivered through Analog Out connectors on the front panel of the RT Box. These analog signals are captured by the RT Box LaunchPad Interface board and routed to the ADC input pins of the TI LaunchPad. Note that since this demo shows CSI in open-loop operation, no ADC blocks are engaged in the "Controller" subsystem. In a real closed-loop application, the MCU then processes these analog signals to generate PWM switching signals, which are delivered to the RT Box via the Digital In pins.

3.2 PLECS offline vs. real-time simulation results

To compare the PLECS simulation result with the real-time simulated waveforms, a PLECS offline simulation needs to run beforehand, and traces are held in the Scope inside the "Plant" subsystem. The simulation results for the inverter phase currents, line-to-neutral phase voltages and load currents are shown as dashed waveforms in Fig. 9. Symmetrical three-phase sinusoidal currents can be observed at the RL-load.

The zoom-in are on the left side focuses on several switching periods around the phase-voltage sector changing from 12 into 1. For one switching period at sector 1, one can see that the generated 6 ideal PWM signals (s_1 to s_6 from top to bottom) in dashed waveforms correspond to the sequence shown in Fig. 6. The zoom-in on the right shows the close-up waveforms of phase A for exactly one switching period ($T_{sw} = 1/20\,\text{kHz} = 50\,\mu\text{s}$) at sector 5 where $v_b > 0 > v_c > v_a$. In both zoom-in captures one can see ideal gating signals with ideal-switched currents and voltages.

Next, let the real-time simulation run following the aforementioned steps. Once both real-time targets are running, **Connect** to the RT Box and **Activate autotriggering**, [R4:Resistor current] is chosen as **Trigger channel** with the pre-set trigger point. The real-time simulation waveforms are overlaid in Fig. 9 as solid waveforms, in comparison to the reference waveforms in dashed lines from PLECS offline simulation with a variable step-size solver. The real-time simulation result shows good agreement to the reference solution.

The "Plant" model is discretized with fixed step-size $T_{\text{disc.plant}} = 2.5\,\mu\text{s}$. The generated C-code is built onto the PLECS RT Box, and takes an average of $2\,\mu\text{s}$ execution time per real-time step. The "Controller" model runs on the MCU with time step equal to switching period, which means $T_{\text{disc.ctrl}} = 50\,\mu\text{s}$. While connecting to **External Mode**, the MCU's interrupt execution time can be shown dynamically by the CPU Load block inside the "Controller" model, which is around 30%.

Note that slight inverter output voltage distortion around sector crossing (see zoom-in on the left side of Fig. 9) is observed in HIL simulation, whereas not in the PLECS offline simulation where ideal PWM patterns at sector crossing are simulated. In fact, the advanced modulation scheme, proposed in Fig. 7(b) of [12], is already implemented in the Controller model to mitigate this problem. However the special handling of the PWM patterns at the very sector crossing instant, as proposed in Fig. 9 of [12], can not be implemented easily in the controller model with automatic embedded code generation.

4 Conclusion

This model demonstrates a current source inverter that supports embedded code generation for TI C2000 MCUs. It can be run in both offline PLECS simulation, as well as in real-time operation. This demo model is based on the previously published conference paper [13].
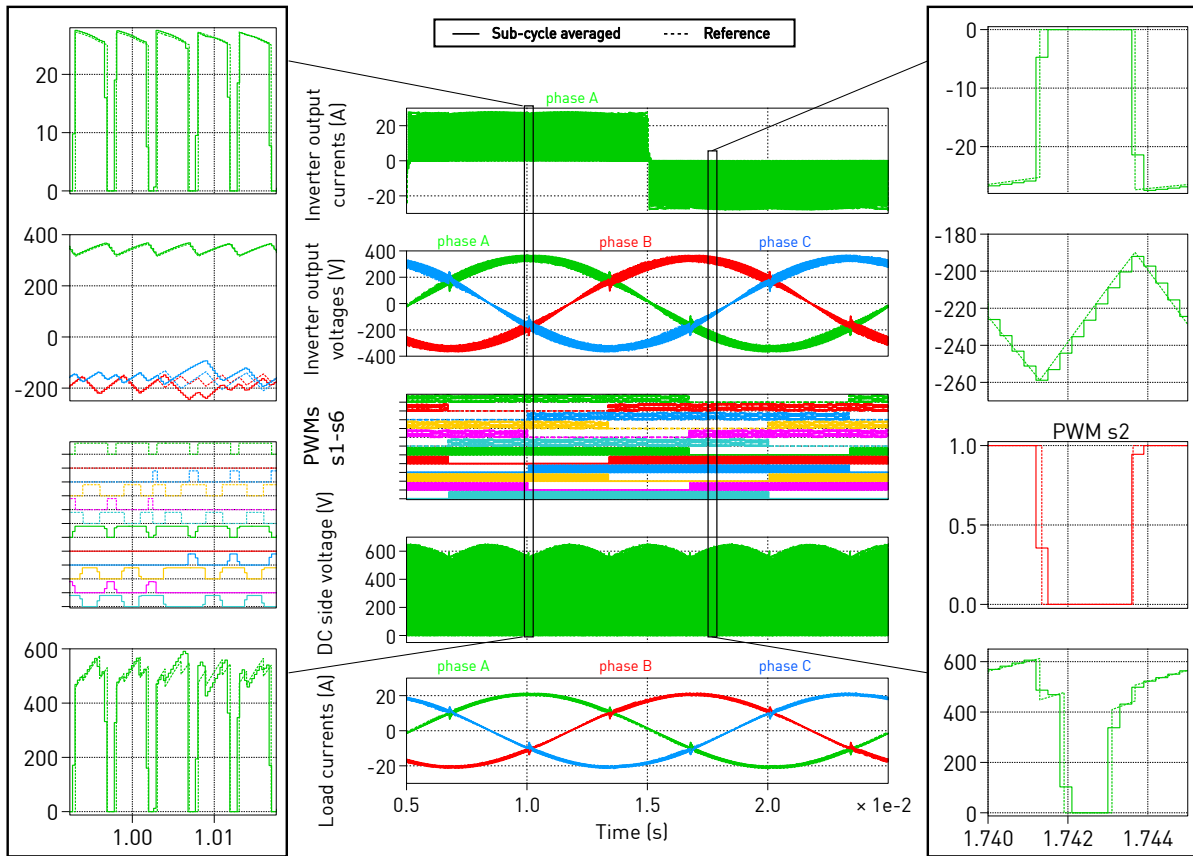
Figure 9: Experimental result comparison of PLECS simulation and RT Box real-time simulation

References

[1] M. Salo and H. Tuusa, "Vector-controlled PWM current-source-inverter-fed induction motor drive with a new stator current control method", *IEEE Transactions on Industrial Electronics*, vol. 52, no. 2, pp. 523-531, April 2005.

[2] S. A. Richter, B. Bader and R. W. De Doncker, "Control of a high power PWM current source rectifier", *2010 International Power Electronics Conference - ECCE ASIA*, June 2010, pp. 1287-1292.

[3] J. Allmeling and W. Hammer, "PLECS – piece-wise linear electrical circuit simulation for Simulink", *Proceedings of the IEEE 1999 International Conference on Power Electronics and Drive Systems. PEDS'99 (Cat. No.99TH8475)*, vol. 1, July 1999, pp. 355-360.

[4] 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)*, Dec 2017, pp. 1-6.

[5] J. Allmeling, N. Felderer and M. Luo, "High Fidelity Real-Time Simulation of Multi-Level Converters", *2018 International Power Electronics Conference (IPEC-Niigata 2018 -ECCE Asia)*, May 2018, pp. 2199-2203.

[6] M. Baumann, T. Nussbaumer and J. W. Kolar, "Comparative evaluation of modulation methods of a three-phase buck + boost PWM rectifier. Part I: Theoretical analysis", *IET Power Electronics*, vol. 1, no. 2, pp. 255-267, June 2008.

[7] T. Nussbaumer, M. Baumann and J. W. Kolar, "Comparative evaluation of modulation methods of a three-phase buck + boost PWM rectifier. Part II: Experimental verification", *IET Power Electronics*, vol. 1, no. 2, pp. 268-274, June 2008.

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

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

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

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

[12] T. Nussbaumer and J. W. Kolar, "Improving Mains Current Quality for Three-Phase Three-Switch Buck-Type PWM Rectifiers", *IEEE Transactions on Power Electronics*, vol. 21, no. 4, pp. 967-973, July 2006.

[13] S. Zhao, N. Felderer and J. Allmeling, "Real-Time Simulation of Three-Phase Current Source Inverter using Sub-Cycle Averaging Method," 2020 IEEE 21st Workshop on Control and Modeling for Power Electronics (COMPEL), Aalborg, Denmark, Nov. 2020, pp. 1-6.

Revision History:

C2000 TSP 1.2.3	First release
C2000 TSP 1.3.1	Update the broken-from-lib PWM (CSI) block
C2000 TSP 1.5.1	Added support for TI 28377S LaunchPad, TI 28388D & 28379D controlCARD targets, and minimized the usage of double-precision math in the controller
C2000 TSP 1.6.1	Added auto-pin selection
C2000 TSP 1.8.1	Substitute the broken-from-lib PWM (CSI) block with the updated library-linked PWM block

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