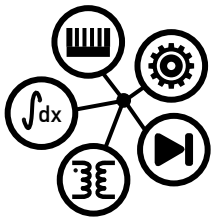

Embedded
Code Generation

DEMO MODEL

SVPWM Control of a Grid-Connected Three-Level NPC Inverter

Control of an NPC inverter with embedded code generation for TI C2000 MCUs

Last updated in C2000 TSP 1.6.1

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

This demo model shows the simulation of a grid-connected NPC inverter in closed current loop using SVPWM (Space-Vector PWM) and a neutral-point balancing technique. It provides an explanation of the typical workflow of the PLECS Embedded Coder, using Texas Instruments (TI) C2000 MCUs. Combined with a PLECS RT Box, the performance of the MCU can be verified directly.

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***

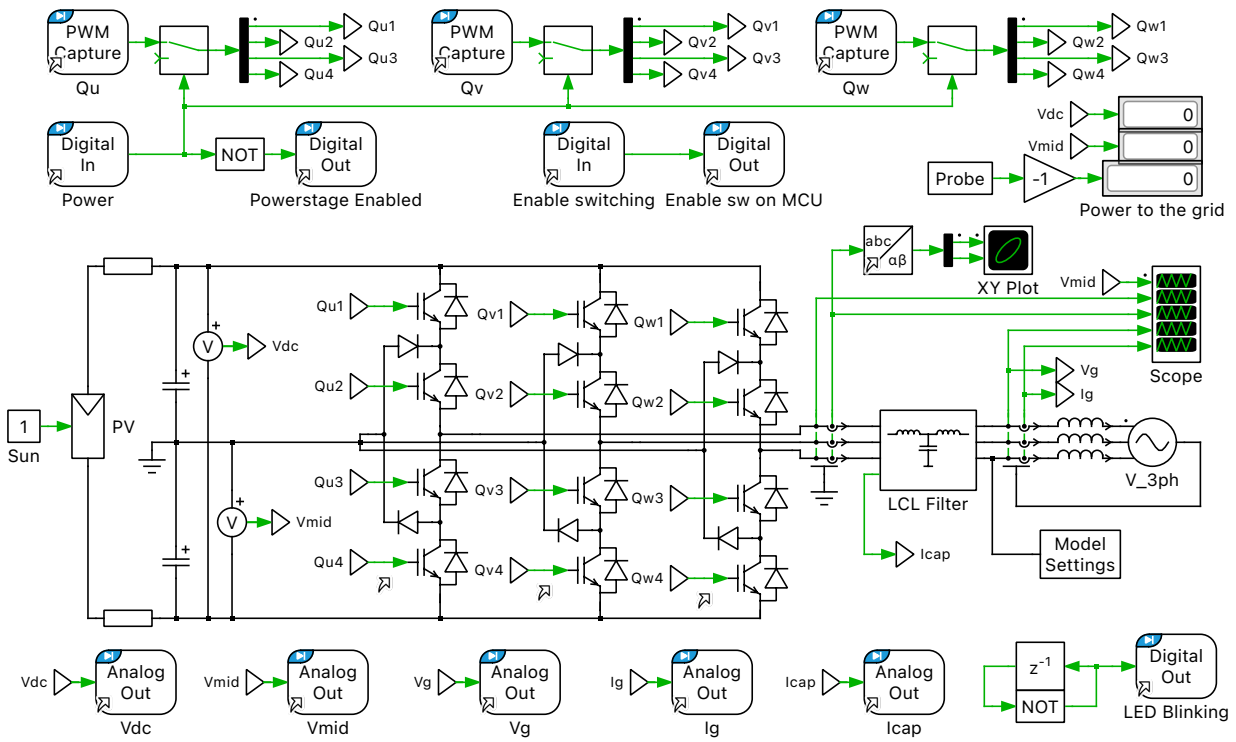

Figure 1: Power circuit of the grid-connected NPC inverter system

2 Model

The demo model is composed of two subsystems: the “Plant” subsystem includes the power circuit and the “Controller” subsystem includes the control loop and SVPWM modulation scheme.

The following sections, 2.1 and 2.2, discuss the power circuit and controls in depth. Optionally, skip to *Section 3* to view instructions on running the real-time simulation.

2.1 Power Circuit

The power circuit includes a three-phase NPC (Neutral-Point Clamped) inverter connected to the grid through an LCL-filter. The DC input supplies a full voltage of 800 V when the “Sun” is at nominal radiance level (i.e. with a value of 1). Two DC capacitors provide inputs to the upper and the lower halves

of the inverter separately. Therefore, a neutral-point balancing technique is included in the SVPWM algorithm. The three-level voltage source inverter (VSI) is represented using three IGBT 3-Level Half Bridge power modules.

The twelve PWM switching signals are brought into the subsystem using three PWM Capture blocks from the PLECS RT Box component library. The measurements of the DC voltages, AC currents, AC voltages, and filter capacitor currents are exported out of the subsystem via Analog Out ports. A Digital Out block labeled "Enable sw on MCU" forwards the external digital input signal to a digital output pin of the RT Box. This pin is connected to the GPIO of the MCU to enable/disable PWM outputs via a finite state machine implemented in control software. The mechanism of the MCU's PWM enable/disable function will be elaborated on in Section 2.2.

LCL-filter design

Due to the pulsating voltage at the three-level inverter output nodes, some kind of filter has to be inserted between the inverter and the grid to attenuate the switching harmonics. The filter is usually composed of inductors and capacitors, which allow the inverter to exchange active and reactive power with the grid by means of inverter control. The design guideline of the LCL-filter used here follows reference [1].

Filter capacitor C_f design To design C_f we have to consider reactive power variations due to the LCL-filter. The design base values can be defined as:

$$Z_\mathrm{b} = \frac{V_\mathrm{LL}^2}{P_\mathrm{n}}, C_\mathrm{b} = \frac{1}{\omega_\mathrm{g} Z_\mathrm{b}}, L_\mathrm{b} = \frac{Z_\mathrm{b}}{\omega_\mathrm{g}},$$

where V_LL^2 is grid line-to-line RMS voltage, P_n is the rated active power, and $\omega_\mathrm{g} = 2\pi f_\mathrm{g}$ is the grid angular frequency. Filter capacitance is related to the base value C_b as a percentage of it. Setting the maximum power variation seen by the grid as 5% (empirical value 1% - 5%), we can obtain:

$$C_\mathrm{f} = 0.05 C_\mathrm{b}$$

Filter inductors L_c and L_g design At low frequencies the LCL-filter behaves like an inductor with the total inductance of:

$$L_\mathrm{dc} = L_\mathrm{c} + L_\mathrm{g}$$

Note that the current through filter capacitors C_f is negligible compared to that of the filter inductors. There is a relation between L_c and L_g that minimizes the voltage drop at the fundamental frequency and maximizes filtering ability. If we express L_c and L_g as a percentage of L_dc with $\alpha \in$ [0, 1], then:

$$L_\mathrm{c} = \alpha L_\mathrm{dc}, L_\mathrm{g} = (1-\alpha) L_\mathrm{dc}$$

Since the LCL resonant circuit consists of L_c, C_f, and L_g all in parallel, the LCL resonant frequency is:

$$\omega_\mathrm{r} = \sqrt{\frac{L_\mathrm{c} + L_\mathrm{g}}{L_\mathrm{c} C_\mathrm{f} L_\mathrm{g}}}$$

Therefore the equivalent inductance L_eq that sets the resonant frequency can be defined as L_c and L_g in parallel:

$$L_\mathrm{eq} = \frac{L_\mathrm{c} L_\mathrm{g}}{L_\mathrm{c} + L_\mathrm{g}} = \alpha(1-\alpha) L_\mathrm{dc}$$

The minimum L_eq is achieved when $L_\mathrm{c} = L_\mathrm{g} = 0.5 L_\mathrm{dc}$. By choosing L_dc to be 10% (empirical value) of the base value L_b, L_c and L_g are easily determined:

$$L_\mathrm{c} = L_\mathrm{g} = 0.05 L_\mathrm{b}$$

Active damping technique using filter capacitor currents

To suppress the high peak gain at the LCL resonant frequency, a damping resistor for the input filter can be included as a passive damping method [2]. Passive damping is a simple approach but introduces additional ohmic losses to the system, while active damping solves the problem through a control modification with a virtual resistor, which is lossless.

Fig. 2 depicts the single-phase equivalent ideal LCL-filter and its corresponding block diagram [1]. Fig. 3 demonstrates the case of a damped filter, where the damping resistor is placed in series with the filter capacitor [1]. The block diagram in Fig. 3 shows that with a virtual gain component (emulating "RD" in Fig. 3) the same damping effect can be added to the control scheme of an ideal LCL-filter while removing the power losses of the physical damping resistor. For the active damping approach, filter capacitor currents have to be measured, which requires three (or only two for the case of balanced three-phase currents) additional analog in-/outputs. The filter in Fig. 2 can be described with the state equa-

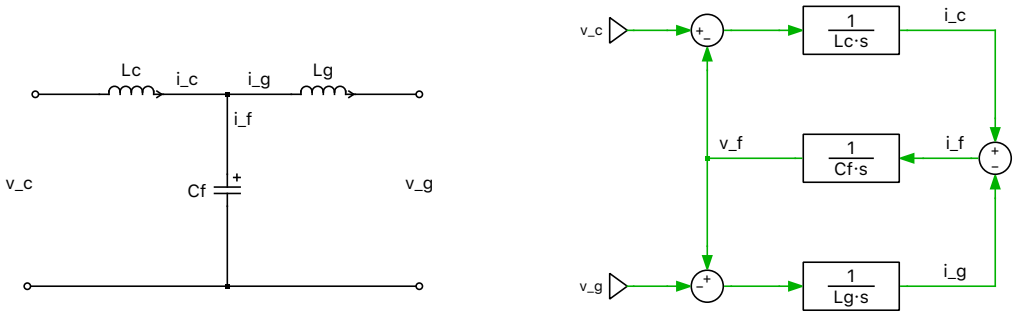

Figure 2: Single-phase equivalent ideal LCL-filter and corresponding block diagram

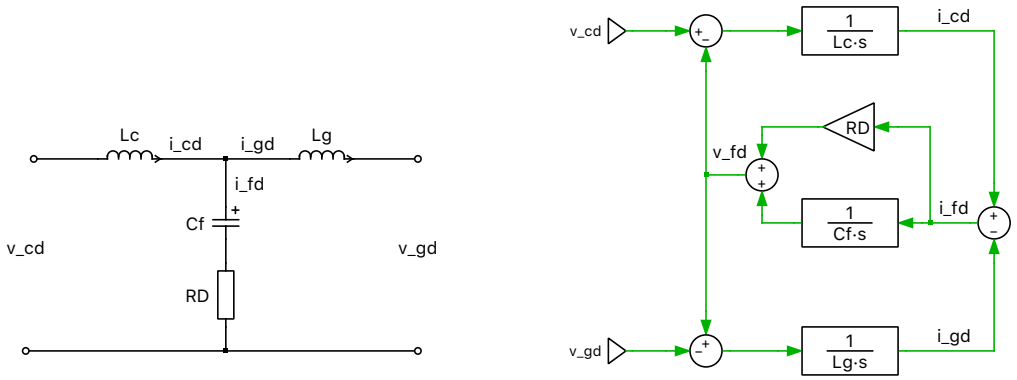

Figure 3: Single-phase equivalent damped LCL-filter and model, damping resistor in series with capacitor

tions in the s-domain:

$$v_{\mathrm{c}}(s) = sL_{\mathrm{c}}i_{\mathrm{c}}(s) + v_{\mathrm{f}}(s)$$

$$v_{\mathrm{g}}(s) = v_{\mathrm{f}}(s) - sL_{\mathrm{g}}i_{\mathrm{g}}(s)$$

$$v_{\mathrm{f}}(s) = \frac{i_{\mathrm{f}}(s)}{sC_{\mathrm{f}}}$$

$$i_{\mathrm{c}}(s) = i_{\mathrm{f}}(s) + i_{\mathrm{g}}(s)$$

The voltage and current transfer functions of the ideal LCL-filter are therefore:

$$H_{\mathrm{v_f}}(s) = \frac{v_{\mathrm{f}}(s)}{v_{\mathrm{c}}(s)} = \frac{1}{L_{\mathrm{c}}C_{\mathrm{f}}}\frac{1}{s^2 + \omega_{\mathrm{r}}^2}$$

$$H_{\mathrm{if}}(s) = \frac{i_{\mathrm{f}}(s)}{v_{\mathrm{c}}(s)} = \frac{i_{\mathrm{f}}(s)}{v_{\mathrm{f}}(s)} \frac{v_{\mathrm{f}}(s)}{v_{\mathrm{c}}(s)} = \frac{1}{L_{\mathrm{c}}} \frac{s}{s^2 + \omega_{\mathrm{r}}^2}$$

where ω_{r} is the resonant frequency defined before.
For the LCL-filter with damping resistor in Fig. 3 we have:

$$v_{\mathrm{c}}^d(s) = s \cdot L_{\mathrm{c}} \cdot i_{\mathrm{c}}^d(s) + v_{\mathrm{f}}^d(s)$$

$$v_{\mathrm{g}}^d(s) = v_{\mathrm{f}}^d(s) - s \cdot L_{\mathrm{g}} \cdot i_{\mathrm{g}}^d(s)$$

$$v_{\mathrm{f}}^d(s) = i_{\mathrm{f}}^d(s) \left(R_{\mathrm{D}} + \frac{1}{sC_{\mathrm{f}}} \right)$$

$$i_{\mathrm{c}}^d(s) = i_{\mathrm{f}}^d(s) + i_{\mathrm{g}}^d(s)$$

The damping resistor R_{D} is often selected as one third of the impedance of the filter capacitor C_{f} at resonant frequency [1]:

$$R_{\mathrm{D}} = \frac{1}{3\omega_{\mathrm{r}} C_{\mathrm{f}}}$$

Therefore we can derive the same voltage and current transfer functions of interest:

$$H_{\mathrm{v_f}}^d(s) = \frac{v_{\mathrm{f}}^d(s)}{v_{\mathrm{c}}^d(s)} = \frac{1}{L_{\mathrm{c}} C_{\mathrm{f}}} \frac{1 + sR_{\mathrm{D}}C_{\mathrm{f}}}{s^2 + \omega_{\mathrm{r}}^2 (1 + sR_{\mathrm{D}}C_{\mathrm{f}})}$$

$$H_{\mathrm{if}}^d(s) = \frac{i_{\mathrm{f}}^d(s)}{v_{\mathrm{c}}^d(s)} = \frac{i_{\mathrm{f}}^d(s)}{v_{\mathrm{f}}^d(s)} \frac{v_{\mathrm{f}}^d(s)}{v_{\mathrm{c}}^d(s)} = \frac{1}{L_{\mathrm{c}}} \frac{s}{s^2 + \omega_{\mathrm{r}}^2 (1 + sR_{\mathrm{D}}C_{\mathrm{f}})}$$

For the case without damping resistor, we define:

$$H_{\mathrm{v_f}}(s) = \frac{v_{\mathrm{f}}(s)}{v_{\mathrm{c}}(s)} = \frac{L_{\mathrm{g}}}{s^2 L_{\mathrm{c}} C_{\mathrm{f}} L_{\mathrm{g}} + L_{\mathrm{c}} + L_{\mathrm{g}}} = H_1(s)$$

$$\frac{i_{\mathrm{f}}(s)}{v_{\mathrm{f}}(s)} = sC_{\mathrm{f}} = H_2(s)$$

For the case with damping resistor, we define:

$$H_{\mathrm{v_f}}^d(s) = \frac{v_{\mathrm{f}}^d(s)}{v_{\mathrm{c}}^d(s)} = \frac{L_{\mathrm{g}}(1 + sR_{\mathrm{D}}C_{\mathrm{f}})}{s^2 L_{\mathrm{c}} C_{\mathrm{f}} L_{\mathrm{g}} + (L_{\mathrm{c}} + L_{\mathrm{g}})(1 + sR_{\mathrm{D}}C_{\mathrm{f}})} = H_3(s)$$

For the active damping technique, we would like to achieve $H_3(s)$ without a physical R_{D}, but rather with a virtual resistor with the value of R_{D}. Fig. 4 shows the block diagram of the active damping control loop with the gain of K_{AD}. We want the closed-loop transfer function in Fig. 4 to be similar to $H_3(s)$.

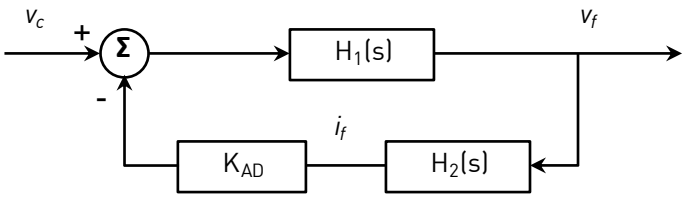

Figure 4: Active damping control loop block diagram with capacitor voltage/current as feedback

Thus in Fig. 4,

$$H_{\mathrm{CL}}(s) = \frac{H_1(s)}{1 + H_1(s)H_2(s)K_{\mathrm{AD}}} = \frac{\frac{L_{\mathrm{g}}}{s^2 L_{\mathrm{c}} C_{\mathrm{f}} L_{\mathrm{g}} + L_{\mathrm{c}} + L_{\mathrm{g}}}}{1 + \frac{L_{\mathrm{g}}}{s^2 L_{\mathrm{c}} C_{\mathrm{f}} L_{\mathrm{g}} + L_{\mathrm{c}} + L_{\mathrm{g}}} sC_{\mathrm{f}} K_{\mathrm{AD}}} = \frac{L_{\mathrm{g}}}{s^2 L_{\mathrm{c}} C_{\mathrm{f}} L_{\mathrm{g}} + L_{\mathrm{c}} + L_{\mathrm{g}} + sL_{\mathrm{g}} C_{\mathrm{f}} K_{\mathrm{AD}}}$$

And with K_{AD} selected as:

$$K_{\mathrm{AD}} = \frac{L_{\mathrm{c}} + L_{\mathrm{g}}}{L_{\mathrm{g}}} R_{\mathrm{D}}$$

we have

$$H_{\mathrm{CL}}(s) = \frac{L_{\mathrm{g}}}{s^2 L_{\mathrm{c}} C_{\mathrm{f}} L_{\mathrm{g}} + L_{\mathrm{c}} + L_{\mathrm{g}} + s L_{\mathrm{g}} C_{\mathrm{f}} \frac{L_{\mathrm{c}}+L_{\mathrm{g}}}{L_{\mathrm{g}}} R_{\mathrm{D}}} = \frac{L_{\mathrm{g}}}{s^2 L_{\mathrm{c}} C_{\mathrm{f}} L_{\mathrm{g}} + (L_{\mathrm{c}} + L_{\mathrm{g}})(1 + s R_{\mathrm{D}} C_{\mathrm{f}})}$$

This final $H_{\mathrm{CL}}(s)$ is close enough to $H_3(s)$ that we can neglect the term "$R_{\mathrm{D}}C_{\mathrm{f}}$" in the numerator of $H_3(s)$ since it is much smaller than 1. Note that the implementation should be considered for both axis regulators for i_{d} and i_{q}. This virtual damping gain K_{AD} is modeled in the controller subsystem (see Fig. 5) as a Gain block named "Virtual damping resistor".

2.2 Controls

In the "Controller" subsystem, two closed-loop d-q current controllers and three-level SVPWM with neutral-point balancing scheme are implemented. Fig. 5 depicts the controller model. It contains ADC and PWM blocks from the TI C2000 Target component library. The measurement of the DC-link voltages, AC currents, AC voltages, and filter capacitor currents are introduced to the model environment using the ADC blocks. In order to convert the detected analog voltage into values with physical units to be used by the control algorithm, a scaling factor and an offset are provided for each channel via the parameter window of the ADC block. The ADC unit and the Analog input channel parameters can be modified accordingly per available resources of different MCUs.

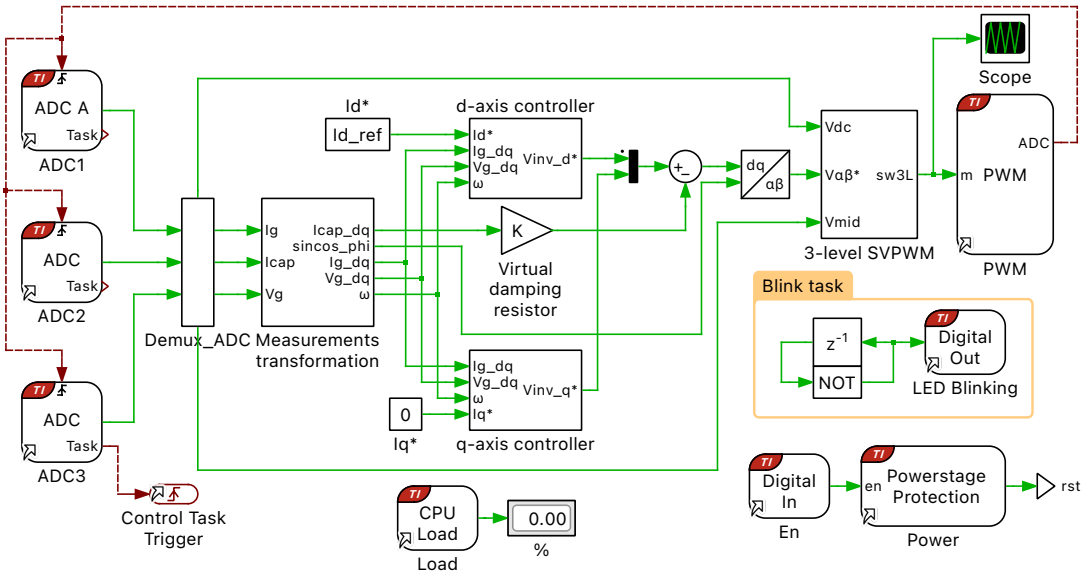

Figure 5: Controller model of the grid-connected NPC inverter system

The i_{d} and i_{q} current controllers are designed based on the following transfer function:

$$H_{\mathrm{OL}}(s) \approx \frac{1}{R_{\mathrm{g}} + R_{\mathrm{c}} + (L_{\mathrm{g}} + L_{\mathrm{c}})s}$$

This neglects the small current flowing into the filter capacitors and the stiff grid inductance $L_{\mathrm{grid}} \ll L_{\mathrm{g}}$.

SVPWM Scheme

There are three NPC legs (phases u, v, and w) shown in Fig. 1. Each leg contains four switches Qx1, Qx2, Qx3, and Qx4 (x = u, v, and w), and these four switches must be controlled in two complementary pairs. Qx1 and Qx3 make one complementary pair, while Qx2 and Qx4 make the other pair. By controlling these four switches the inverter output allows for three different voltage levels. Table 1 lists the three valid states for each leg, in which P means a switch leg is connected to the positive DC rail, N means it is connected to the negative DC rail, and O means it is connected to the neutral-point potential.

Table 1: Output status of each leg based on different switching combination of Qx1-Qx4

Switch No.	Qx1	Qx2	Qx3	Qx4	Phase voltage	Leg status
1	ON	ON	OFF	OFF	$\frac{V_{\rm dc}}{2}$	P
2	OFF	OFF	ON	ON	$-\frac{V_{\rm dc}}{2}$	N
3	OFF	ON	ON	OFF	0	O

Thus, in total there are 27 states of the three-level VSI which can be mapped to the space-vector diagram, depicted in Fig. 6. Assuming a reference vector $V_{\rm ref}$, according to the theory of SVPWM, we need to find the two nearest vectors $V_{\rm X}$, $V_{\rm Y}$ and one zero vector $V_{\rm Z}$ in order to synthesize $V_{\rm ref}$. Therefore, vector PNN ($V_{\rm X}$), PON ($V_{\rm Y}$) and NNN ($V_{\rm Z}$) in Fig. 6 can be selected accordingly to form $V_{\rm ref}$.

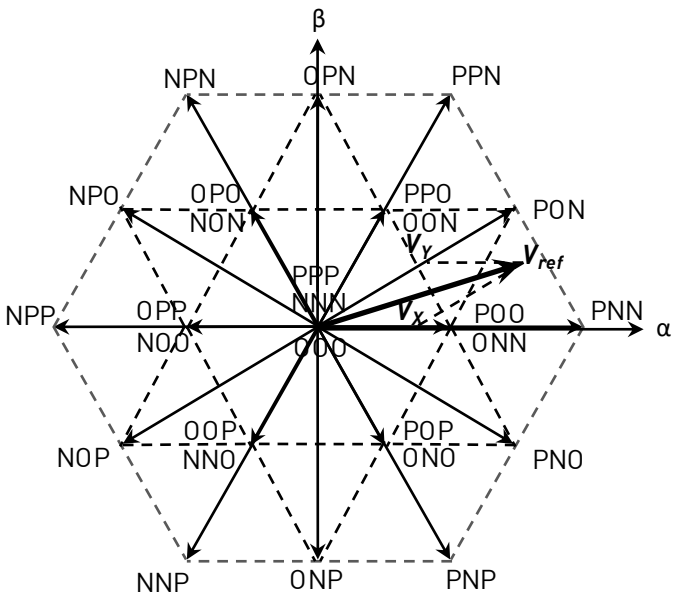

Figure 6: Three-phase three-level inverter SVPWM vector diagram

If the dwelling time of vector $V_{\rm X}$, $V_{\rm Y}$ and $V_{\rm Z}$ inside a switching period $T_{\rm sw}$ are $T_{\rm X}$, $T_{\rm Y}$ and $T_{\rm Z}$ respectively, the following functions must be satisfied:

$$V_{\rm X}T_{\rm X} + V_{\rm Y}T_{\rm Y} + V_{\rm Z}T_{\rm Z} = V_{\rm ref}T_{\rm sw} \tag{1}$$

$$T_{\rm X} + T_{\rm Y} + T_{\rm Z} = T_{\rm sw} \tag{2}$$

However, it is difficult to determine $V_{\rm X}$, $V_{\rm Y}$ and $V_{\rm Z}$ by the angle only, which is used in the 2-level SVPWM scheme, because the reference vector can be located in different sectors even if the angle is the same. To determine the sector, the amplitude of the reference vector is also needed, but this increases the complexity of the calculation.

Thus, [3] presented a simplified way to determine $V_{\rm X}$, $V_{\rm Y}$ and $V_{\rm Z}$, using the core of 2-level SVPWM. First, the whole vector diagram shown in Fig. 6 is divided into six main sectors. Each main sector has a shape of a sub-hexagon, and all six sub-hexagons distribute continuously with a 60° angle difference. Fig. 7 depicts sub-hexagon 1 and 2 as an example.

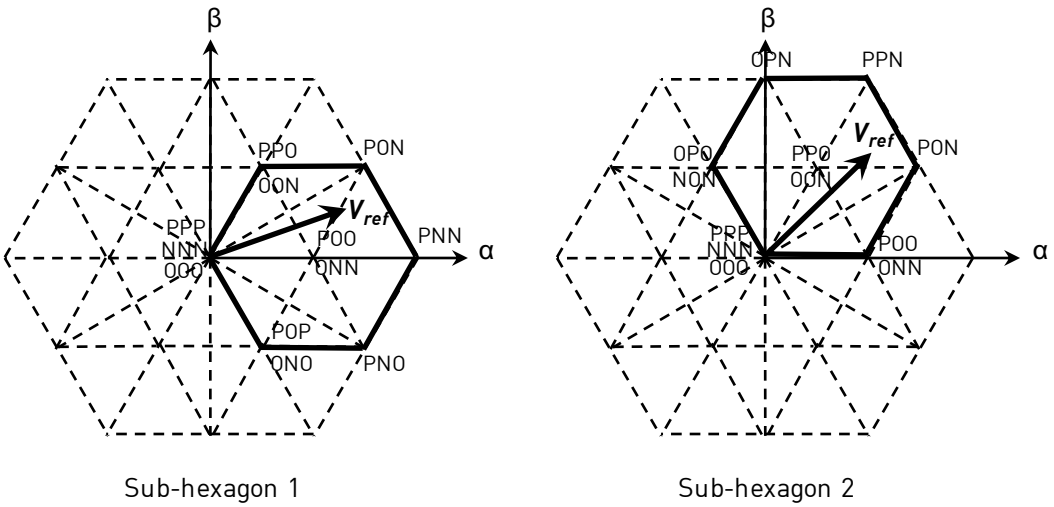

Figure 7: Sub-hexagon 1 and 2 out of the divided six sub-hexagons in the whole vector diagram

After the main sector (sub-hexagon) is determined, the original vectors must be mapped into the selected main sector. The mapping algorithm follows:

$$V' = V_{\text{original}} - V_{\text{map}} \tag{3}$$

For example the original vectors in the main sector 1 are PPP (OOO, NNN), POP (ONO), PNO, PNN, PON, PPO (OON), and POO (ONN). To get a hexagon similar to the 2-level SVPWM, take POO (ONN) as the mapping vector $V_{\text{map1}} = V_0$. After the mapping we can get the hexagon shown in Fig. 8, which is the same vector diagram as a 2-level SVPWM.

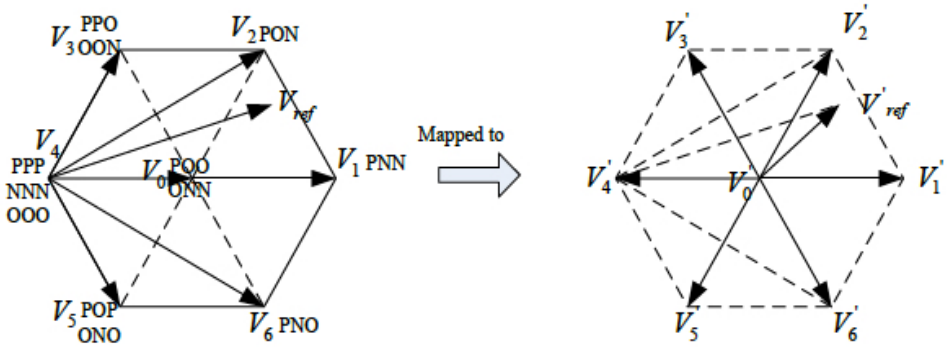

Figure 8: Mapping from the 3-level SVPWM to a 2-level SVPWM in sub-hexagon 1

From Fig. 8, it can be seen that V'_{ref} is still in the mapped sub-hexagon 1, and we can easily determine that the dwelling vectors are V'_1 and V'_2. The V'_0 can be taken as the zero vector in this 2-level SVPWM. So, we can get the following function:

$$V'_1 T_{\text{X}} + V'_2 T_{\text{Y}} + V'_0 T_{\text{Z}} = V'_{\text{ref}} T_{\text{sw}} \tag{4}$$

Note that Eq. (2) is still valid here. Combing Eq. (2) and Eq. (4), one can derive:

$$(V_1 - V_{\text{map1}})T_{\text{X}} + (V_2 - V_{\text{map1}})T_{\text{Y}} + (V_0 - V_{\text{map1}})T_{\text{Z}} = (V_{\text{ref}} - V_{\text{map1}})T_{\text{sw}} \tag{5}$$

which means:

$$V_1 T_{\text{X}} + V_2 T_{\text{Y}} + V_0 T_{\text{Z}} = V_{\text{ref}} T_{\text{sw}} \tag{6}$$

Thus, if the dwelling time of V_1', V_2', and V_0' can be calculated, the original vector dwelling time can be determined. From the mapping in Fig. 8, the vector selection and the dwelling time calculation of the 3-level SVPWM are converted to 2-level SVPWM totally.

Different main sectors have different mapping vectors. Table 2 summarizes the mapping vector for each main sector.

Table 2: Mapping vector for each main sector

Main sector No.	Mapping vector	Element of α	Element of β
1	POO or ONN	$\frac{V_{\rm dc}}{3}$	0
2	PPO or OON	$\frac{V_{\rm dc}}{6}$	$\frac{\sqrt{3}V_{\rm dc}}{6}$
3	OPO or NON	$-\frac{V_{\rm dc}}{6}$	$\frac{\sqrt{3}V_{\rm dc}}{6}$
4	OPP or NOO	$-\frac{V_{\rm dc}}{3}$	0
5	OOP or NNO	$-\frac{V_{\rm dc}}{6}$	$-\frac{\sqrt{3}V_{\rm dc}}{6}$
6	POP or ONO	$\frac{V_{\rm dc}}{6}$	$-\frac{\sqrt{3}V_{\rm dc}}{6}$

The main sector number can be defined by the angle of the $V_{\rm ref}$ in the α-β coordinate plane. For example, according to Fig. 7, the angle range of main sector 1 is $[-\frac{\pi}{3}, \frac{\pi}{3}]$, and the angle range of main sector 2 is $[0, \frac{2\pi}{3}]$. Thus, the overlapped area between main sector 1 and 2 can be split into the two adjacent areas equally, in order to have monopolized angle area for each sector. Fig. 9 depicts the simplified definition of the six main sectors.

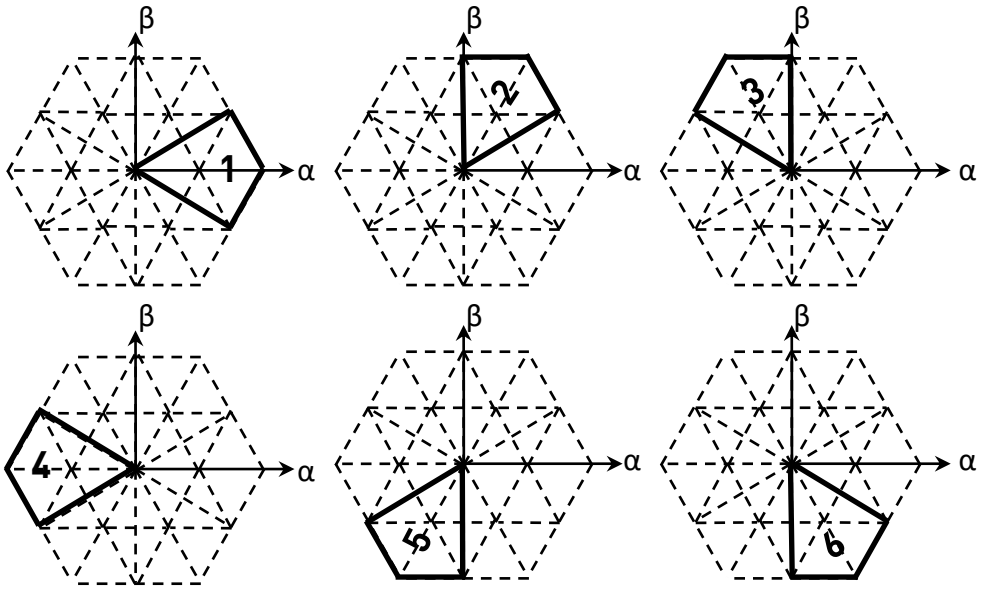

Figure 9: Simplified definition of the six main sectors

In the 2-level SVPWM, the first step is to find the sector number which can determine the dwelling vectors. The second step is to calculate the dwelling time for each of the selected vectors. According to the principle of 3-level SVPWM, when the main sector is determined and all the vectors are mapped to the main sector, the same process done in 2-level SVPWM can be implemented to determine the sub-sector and calculate the dwelling times for each dwelling vectors. Here in this demo model, a simple and effective way is deployed [3] to calculate the duty cycle for each switching pair out of the dwelling time of each dwelling vector.

Again, we will use main sector 1 as an example. According to Fig. 7 there is no N status for u phase. Besides, if OON, ONO, and OOO are selected for the vector mapping, there will be no P status for the v and w phases. For the u phase, we represent the P status with 1, and the O status with 0. For v and w phases, we represent the O status with 1, and the N status with 0. Fig. 10 depicts this replacement operation. After this downsizing operation, the dwelling time for three vectors can be determined. As shown in Fig. 10, T_X is the dwelling time of status 100, T_Y represents the one of status 110, and T_Z is the time for status 111 and 000.

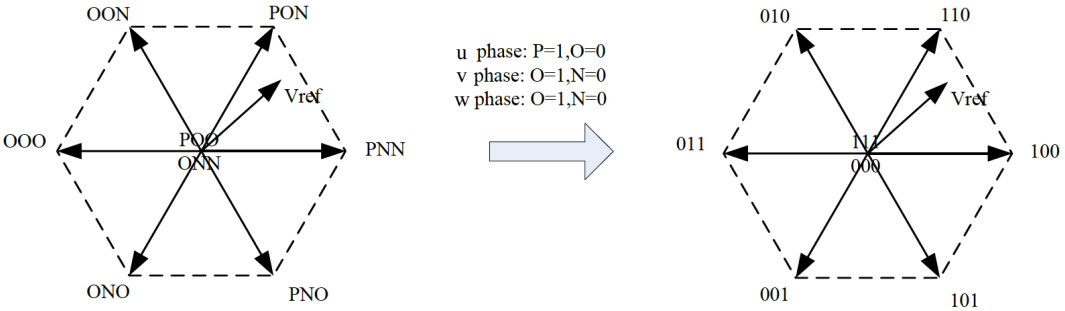

Figure 10: The status replacement rule downsizing 3-level vector diagram to 2-level one in main sector 1

Here we can calculate the three duty cycles for the upper switches out of the three complementary switch pairs (d1, d2, and d3), with the symmetrical PWM mode. The left side of Fig. 11 shows that effectively, the resulting 2-level vector sequence is 000 → 100 → 110 → 111→ 110 → 100 → 000. Next, if we apply the same status replacement rule as in Fig. 10, we get the right part of Fig. 11. Thus we can achieve the 3-level vector sequence: ONN → PNN → PON → POO → PON → PNN → ONN.

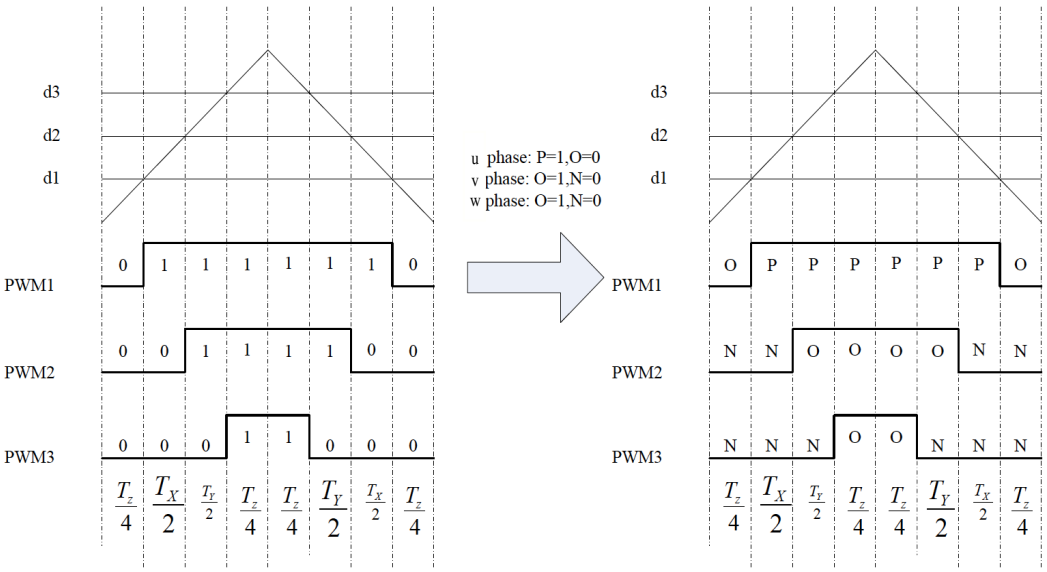

Figure 11: Symmetrical 2-level SVPWM applying the status replacement rule back to a 3-level SVPWM

Table 3 summarizes this status replacement rule for each main sector. The positive pair of power switches is Qx1 and Qx3 (x = u, v, w); the negative pair of the power switches is Qx2 and Qx4 (x = u, v, w). We also define the same status 0 and 1 for each pair as the 2-level SVPWM. So for the main sector 1, since in one switching cycle, u phase has no N status, the negative pair u phase is always 0. Similarly for the v and w phases, since they have no P status, the positive pair is always 0. That means in the main sector 1, d1 can be assigned to the positive pair of u phase (i.e. Qu1 has duty cycle of d1, and Qu3 has complementary logic to Qu1), d2 can be assigned to the negative pair of v phase (i.e. Qv2 has duty cycle of d2, and Qv4 has complementary logic to Qv2), and d3 can be assigned to the negative pair

of w phase (i.e. Qw2 has duty cycle of d3, and Qw4 has complementary logic to Qw2). The process can be extended to all the six main sectors and Table 4 shows this duty cycle assignment rule.

Table 3: Status replacement rule for each main sector

Main sector	U phase		V phase		W phase	
No.	1	0	1	0	1	0
1	P	O	O	N	O	N
2	P	O	P	O	O	N
3	O	N	P	O	O	N
4	O	N	P	O	P	O
5	O	N	O	N	P	O
6	P	O	O	N	P	O

Table 4: Duty cycle assignment rule for each main sector

Main sector No.	U phase		V phase		W phase	
	Positive pair (Qu1)	Negative pair (Qu2)	Positive pair (Qv1)	Negative pair (Qv2)	Positive pair (Qw1)	Negative pair (Qw2)
1	d1	1	0	d2	0	d3
2	d1	1	d2	1	0	d3
3	0	d1	d2	1	0	d3
4	0	d1	d2	1	d3	1
5	0	d1	0	d2	d3	1
6	d1	1	0	d2	d3	1

The simplified 3-level SVPWM algorithm discussed above is implemented using the C-Script block and the output provides six modulation index values for Qu1, Qu2, Qv1, Qv2, Qw1 and Qw2 individually, as summarized in Table 4. The complementary signal for each of them is realized by configuring the opposite Polarity in the corresponding PWM Out blocks.

Neutral-point balancing technique

The active neutral-point balancing technique implemented in this demo model is based on [4]. Neutral-point current i_{NP} is denoted as the current coming out of the DC capacitors neutral point shown in Fig. 1. According to the SVPWM vector diagram shown in Fig. 6, Table 5 summarizes all the vectors that influence the neutral-point balance. $i_{\mathrm{u,v,w}}$ denotes the current direction going out of the switching node of the three-level phase leg entering the grid side.

Small vectors come in pairs, e.g. POO and ONN, and they disturb the neutral point with exactly the same current value, but just the opposite sign. Furthermore, as shown in Fig. 10, POO and ONN are the new zero vectors in the downsized 2-level vector diagram in main sector 1. Therefore, the active control of neutral-point voltage lies in the manipulation of this new zero vector pair. This requires measuring the middle-point voltage (v_{mid} in Fig. 1) in addition to the full DC voltage (v_{dc} in Fig. 1). As a result, the total zero vector dwelling time T_{Z} in Fig. 11 will be split between its 2-level “111” and “000” vectors, each proportional to its half DC voltage. Since the medium vectors disturb the neutral point and there is no vector that can be used to balance with the medium vector inside each main sector, some neutral-point

variation can still be observed on the scale of each main sector. For more details and limitations of this method, please refer to [4].

Table 5: Neutral-point current i_{NP} for different space vectors

Positive Small Vectors	i_{NP}	Negative Small Vectors	i_{NP}	Medium Vectors	i_{NP}
ONN	i_{u}	POO	-i_{u}	PON	i_{v}
PPO	i_{w}	OON	-i_{w}	OPN	i_{u}
NON	i_{v}	OPO	-i_{v}	NPO	i_{w}
OPP	i_{u}	NOO	-i_{u}	NOP	i_{v}
NNO	i_{w}	OOP	-i_{w}	ONP	i_{u}
POP	i_{v}	ONO	-i_{v}	PNO	i_{w}

Configuring TI C2000 Target library components

The output of the SVPWM modulator is provided as an input to the PWM block, in the form of a duty cycle, in the range of [0, 1] for generating gate signals. The **Carrier type**, **Carrier frequency** and **Dead time** parameters can be configured in an intuitive way from the **Main** tab of the PWM block parameters window. Please note that each duty cycle signal is fed to both A and B outputs of an ePWM module, in opposite polarity with dead time. From the **Events** tab of the PWM block parameter window, the **ADC Trigger** parameter is configured as `Underflow`. This makes the first ePWM module configured in the **PWM generator** of the **Main** tab generate a "start of conversion" impulse for the ADCs, whenever the carrier value reaches its minimum. With this configuration, an additional output port called "ADC" appears on the mask of the PWM block.

Since TI microcontroller 28379D has four ADC modules, here ADC module A, B, and C are used to read in the measurements. On each ADC block, **Trigger source** is chosen as "Show trigger port", thus a trigger symbol appears on the mask of the ADC block. Each trigger port of the ADC block is connected to the ADC port of the PWM block, via the red dashed signal wire. This means that the start of conversion signal generated by the first ePWM module triggers the conversion of ADC A, B, and C at the same time.

The control task is executed after the last conversion on ADC C module. This is configured by connecting the output port called "Task" of ADC C to a "Control Task Trigger" block from the TI C2000 Target component library, by the red dashed signal wire.

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" from the "Plant" subsystem, is physically routed to 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 device for safety. 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 input to the "Powerstage Protection" block is a transition of low to high, enabling the PWMs, the output of the "Powerstage Protection" block is set to active low. This output is provided as an input to the Digital In block labeled "Power" of the "Plant" subsystem via the RT Box LaunchPad Interface board, allowing the captured PWM signals to pass to the gates of the inverter bridge. The red LED "DO-29" on the LaunchPad Interface board will turn on, visually indicating the same.

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 280039C [5], TI 28P650DK9 [7], and TI 28379D [6] LaunchPads, and the TI 28388D [9] controlCARD.

Additionally, the demo model allows for code generation for the TI 280039C [10] and TI 28379D [8] controlCARDs. 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. 12, 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).

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** subtab, 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” on the LaunchPad 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 [11] for step-by-step instructions), enter its path into the **CCS project directory** field and click **Build**. The code of the “Controller” subsystem will be automatically generated. Then, proceed to build and debug the project as a normal CCS project.

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

Prior to controlling a real hardware prototype of the 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. 12, 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.

- 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.

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

Within the “Plant” subsystem or the power circuit 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. The MCU then processes these analog signals to generate PWM switching signals, which are delivered to the RT Box via the Digital In pins.

Figure 12: Hardware setup of the HIL verification

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.

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” should enable the system switching again.

Note At this stage, verify that the LED corresponding to “DO-29” on the RT Box LaunchPad Interface board is turned 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** tab 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.
- Now, **Connect** to the RT Box and **Activate autotriggering** to observe the real-time results on the Scope of the “Plant” subsystem.

In this demo model, a step change in the d-axis current reference can be configured by changing the value of the Constant block “Id*” in the “Controller” subsystem to 1.2 times its nominal value. This value can be varied on the fly, in real-time, since it has been added to the **Exceptions** list in the **Parameter Inlining** tab of the **Coder options...** window. The waveforms with this step change can be observed in Fig. 13.

The influence of the virtual resistor on active damping can also be observed. Change the d-axis current reference, “Id*”, back to its original value, if it has been varied. Then, change the Gain block named “Virtual damping resistor” in the “Controller” subsystem to 0.7 times its original value. A moderate amount of ringing can be observed in both the inverter and grid currents. Fig. 14 depicts this phenomenon. This ringing effect is due to the inadequate virtual damping applied on the LCL-filter. Changing the “Virtual damping resistor” back to its original value will eliminate the ringing.

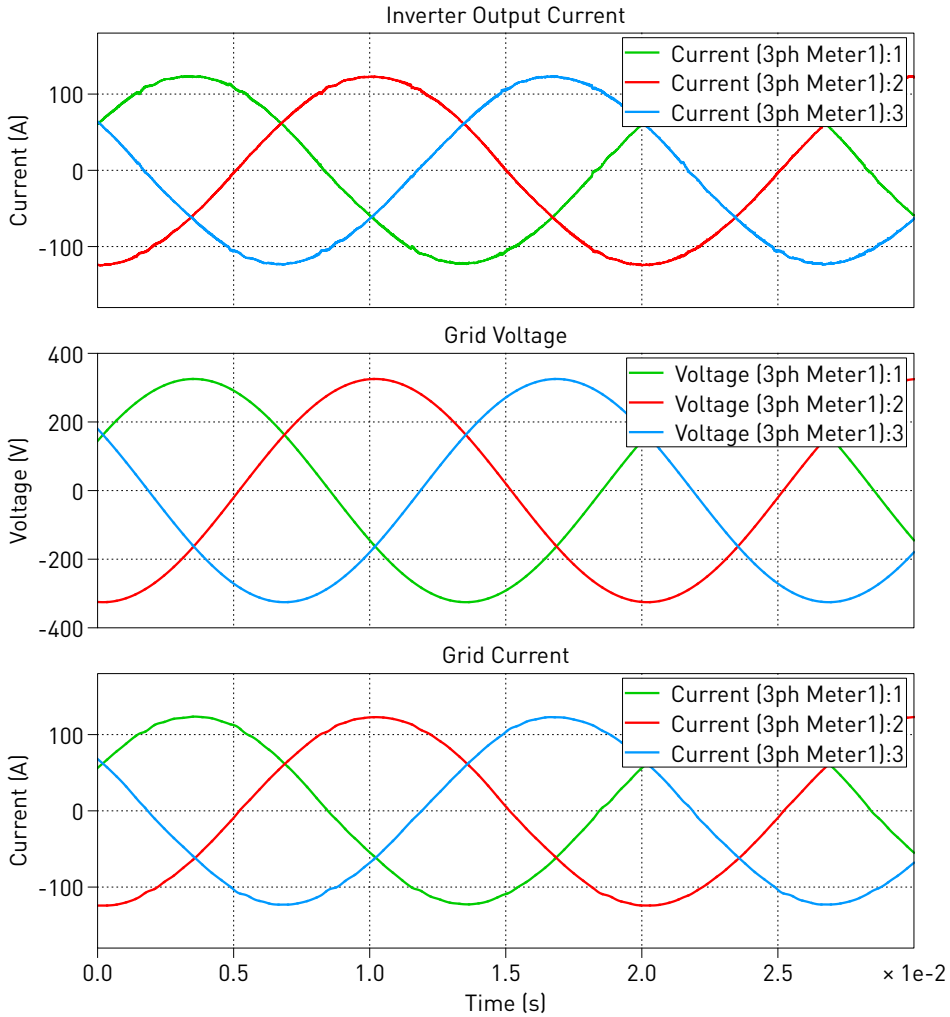

Figure 13: Inverter waveforms after changing to 1.2 times the original value in the d-axis current reference of the controller in real-time

4 Conclusion

This model demonstrates a grid-connected NPC inverter system that supports embedded code generation for TI C2000 MCUs.

References

[1] R. Teodorescu, M. Liserre and P. Rodriguez, “Grid converters for photovoltaic and wind power systems”, pp. 289-311, IEEE, Wiley, 2011.

[2] R. W. Erickson, “Optimal single resistors damping of input filters”, *APEC ’99, Fourteenth Annual Applied Power Electronics Conference and Exposition*, vol.2, pp. 1073-1079, March 1999.

[3] V. Xue, “Center-Aligned SVPWM Realization for 3-Phase 3-Level Inverter (Application Report)”, Texas Instruments, October 2012.

[4] N. Celanovic and D. Boroyevich, “A comprehensive study of neutral-point voltage balancing problem in three-level neutral-point-clamped voltage source PWM inverters”, *IEEE Transactions on Power Electronics*, vol. 15, no. 2, pp. 242-249, March 2000.

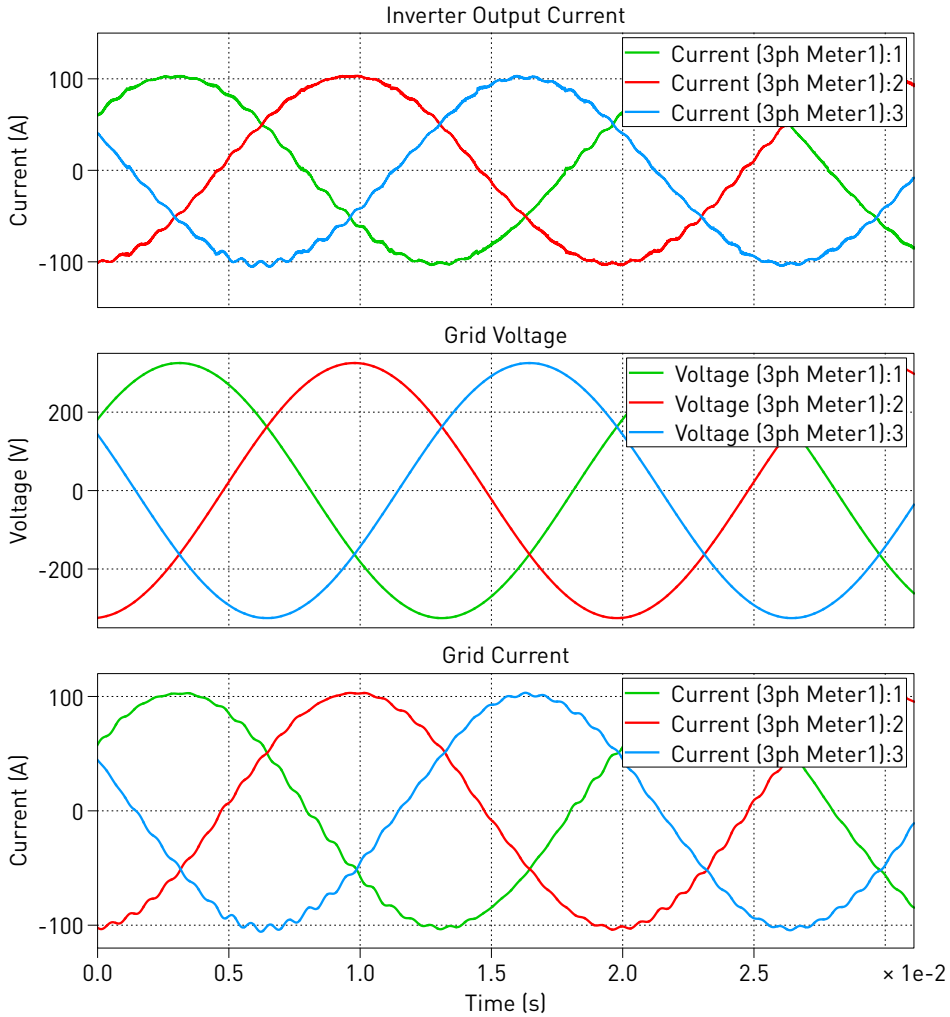

Figure 14: Inverter waveforms with poorly damped virtual resistor in real-time

[5] TI C2000 F280039C LaunchPad development kit,
URL: http://www.ti.com/tool/LAUNCHXL-F280039C.

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

[7] TI C2000 F28P650DK9 LaunchPad development kit,
URL: https://www.ti.com/tool/LAUNCHXL-F28P65X.

[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.

[10] TI C2000 F280039C controlCARD evaluation module,
URL: https://www.ti.com/tool/TMDSCNCD280039C.

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

Revision History:

C2000 TSP 1.0	First release
C2000 TSP 1.5.1	Added support for 28388D and 28379D controlCARD targets, and minimized the usage of double-precision math in the controller
C2000 TSP 1.6.1	Added support for 280039C LaunchPad and controlCARD targets, and auto-pin selection

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