Implementation of a Three-Phase Grid-Connected Inverter Controller

Processor-in-the-Loop Implementation on TI C2000 F28379D

Rohan Lalwani
Department of Electrical Engineering
IIT Roorkee

March 6, 2024

Abstract

This report details the implementation of a digital control system for a 10 kVA three-phase Grid-Connected Inverter (GCI). The system is designed to inject 3000 W of active power and 3000 VAR of reactive power into a 415 V, 50 Hz balanced grid. The control strategy employs a Synchronous Reference Frame Phase Locked Loop (SRF-PLL) for synchronization and a decoupled dq-frame current controller for power regulation. The entire control loop, including a discrete-time model of the L-filter and grid, is implemented on a TI C2000 F28379D microcontroller in a Processor-in-the-Loop (PIL) configuration. The results demonstrate accurate grid synchronization and precise tracking of power references.

Contents

1 Problem Statement

The objective of this project is to implement a control scheme for a three-phase inverter with the following specifications

- **DC Bus Voltage:** 800 V
- **Grid Voltage:** 415 V (Line-to-Line), 50 Hz
- **Switching Frequency:** 10 kHz
- **Power Injection:** 3000 W (Active), 3000 VAR (Reactive)
- **Filter Parameters:** Inductance drop of 10% of grid voltage; Resistance 0.1Ω.

The project is divided into three implementation sections:

1. Open-loop generation of three-phase balanced sinusoidal voltages using ePWM.
2. Implementation of SRF-PLL to obtain grid phase angle.
3. Full closed-loop control (Current and Power loops) with a discrete domain plant model.

2 Section 1: Open Loop 3-Ph Voltages Generation

2.1 Methodology

The first stage involves verifying the ePWM and DAC peripherals. Three sinusoidal reference signals shifted by 120° are generated. The instantaneous values are scaled and sent to the Digital-to-Analog Converter (DAC) for verification on an oscilloscope.

2.2 Implementation

The implementation uses a timer interrupt at 10 kHz. In every interrupt, the phase angle θ is incremented based on the reference frequency (50 Hz).

```
// Interrupt Service Routine for ePWM
extern interrupt void epwm2_isr(void){
    // Update angle
    theta = theta + (2 * PI * ref_freq * 100e-6); // 100us sampling
    if (theta > 2*PI) theta -= 2*PI;

    // Generate balanced three-phase voltages
    va = Vm * sin(theta);
    vb = Vm * sin(theta - 2*PI/3);
    vc = Vm * sin(theta - 4*PI/3);

    // Scale and output to DAC for visualization
    DacaRegs.DACVALS.bit.DACVALS = (0.5 + (va/Vm)/2.0) * 4095;
    DacbRegs.DACVALS.bit.DACVALS = (0.5 + (vb/Vm)/2.0) * 4095;

    // Acknowledge Interrupt
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP3;
```

```
    EPwm2Regs.ETCLR.bit.INT = 1;
}
```

Listing 1: SPWM Generation Logic

2.3 Results

The code successfully generates three sinusoidal waves. The DAC output observed on the DSO confirms the 120° phase shift and accurate frequency generation of 50 Hz.

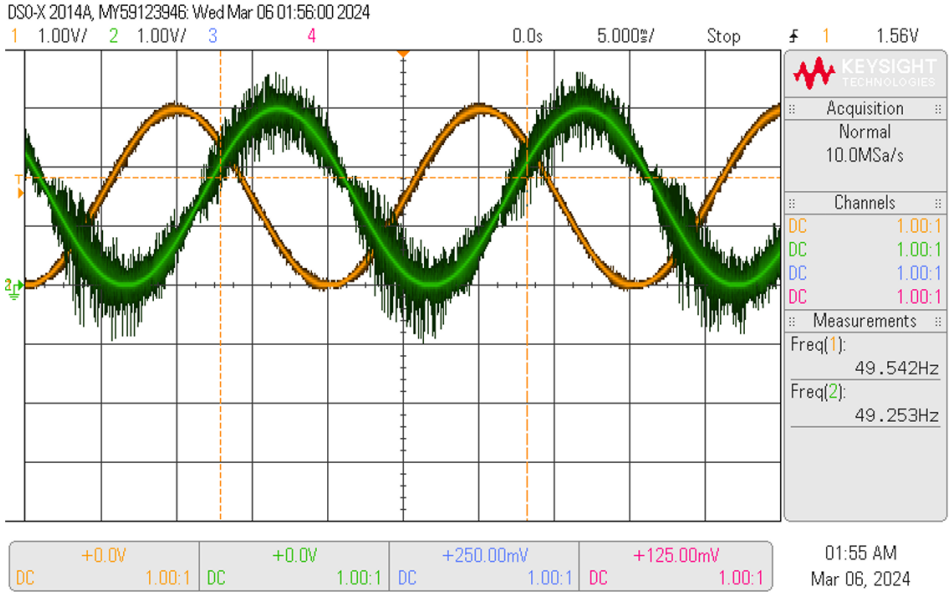

Figure 1: Open-loop SPWM generation showing balanced 3-phase voltages.

3 Section 2: Grid Synchronization (SRF-PLL)

3.1 SRF-PLL Theory

Grid synchronization is achieved using a Synchronous Reference Frame PLL. The three-phase grid voltages v_{abc} are transformed to the rotating dq frame. The PLL controller regulates the q-axis voltage v_q to zero. When $v_q = 0$, the d-axis is aligned with the grid voltage vector, and the estimated angle θ_{PLL} tracks the grid angle.

3.2 Discrete Implementation

The PI controller for the PLL is discretized using the Tustin transformation. The coefficients b_0 and b_1 are pre-calculated based on the sampling time ($T_s = 100\mu s$) and PI gains ($K_p = 0.687, K_i = 77.47$).

```
// ABC to DQ Transformation
vd = (2.0/3.0)*((va*sin(thetaPLL)) + (vb*sin(thetaPLL - 2*PI/3))
                + (vc*sin(thetaPLL - 4*PI/3)));
vq = (2.0/3.0)*((va*cos(thetaPLL)) + (vb*cos(thetaPLL - 2*PI/3))
                + (vc*cos(thetaPLL - 4*PI/3)));

```

```
// PI Controller (Tustin)
delta_f = delta_f_pre + b0*vq + b1*vq_pre;

// Frequency and Angle Integration
f = delta_f + line_freq; // Feedforward 50Hz
thetaPLL1 = thetaPLL_pre + PI*sampling_time*(f + f_pre);

// Wrap angle to 0-2pi
thetaPLL = (thetaPLL1 <= 2*PI) ? thetaPLL1 : (thetaPLL1 - 2*PI);

// Update History Terms
vq_pre = vq;
delta_f_pre = delta_f;
thetaPLL_pre = thetaPLL;
f_pre = f;
```

Listing 2: SRF-PLL Implementation

3.3 Results

The PLL accurately locks onto the generated grid voltage. The oscilloscope results (Figure 2) show the grid voltage (Channel 1) and the PLL angle sawtooth wave (Channel 2) are perfectly synchronized.

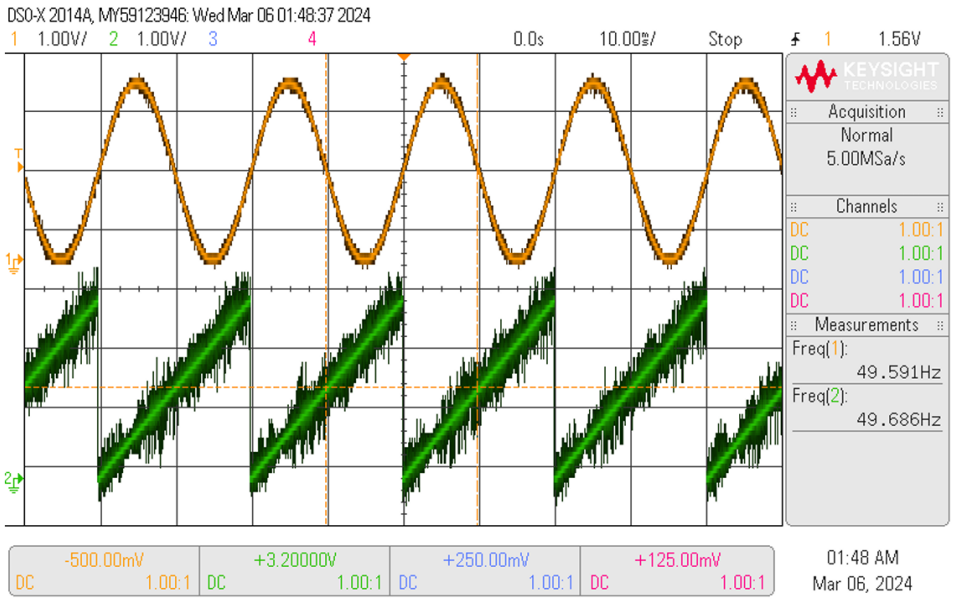

Figure 2: Grid phase angle tracking using SRF-PLL. Channel 1: Grid Voltage (Yellow), Channel 2: PLL Angle (Green).

4 Section 3: Closed Loop Control (PIL)

4.1 Control Structure

The complete control system consists of:

1. **Outer Power Loop:** Generates i_d^* and i_q^* references to track P_{ref} and Q_{ref}.

2. **Inner Current Loop:** Regulates i_d and i_q using decoupling terms to handle cross-coupling between axes.

3. **Plant Model (PIL):** Simulates the behavior of the L-filter and grid interaction directly on the MCU.

4.2 Discrete Plant Model

To validate the controller without high-voltage hardware, the inverter and grid dynamics are modeled using difference equations derived from:

$$v_{inv} - v_{grid} = L\frac{di}{dt} + iR \tag{1}$$

```
void inverter(void) {
    // Reconstruct Inverter Phase Voltages
    vao = vdc * refa;
    vbo = vdc * refb;
    vco = vdc * refc;
    vno = (vao + vbo + vco) / 3.0; // Neutral Voltage

    // Voltage across inductor
    vla = vao - va - vno;
    vlb = vbo - vb - vno;
    vlc = vco - vc - vno;

    // Update Currents (Discrete Integration)
    ia = (-b3*ia + (vla + vla_pre)) / b2;
    ib = (-b3*ib + (vlb + vlb_pre)) / b2;
    ic = (-b3*ic + (vlc + vlc_pre)) / b2;

    // Store previous values
    vla_pre = vla;
    vlb_pre = vlb;
    vlc_pre = vlc;
}
```

Listing 3: Inverter Plant Model (PIL)

4.3 Power and Current Control

The controller implements decoupling to ensure independent control of Active and Reactive power.

```
// --- Power Loop ---
// Calculate P and Q
pdq = 1.5 * vd * id;
qdq = 1.5 * vd * iq;

// PI Control for References
error_p = pref - pdq;
idref = idref + (b4*error_p) + (b5*error_p_pre);

error_q = qref - qdq;
iqref = iqref + (b6*error_q) + (b7*error_q_pre);

```

```
// --- Current Loop ---
error_id = idref - id;
ud = ud + (b8*error_id) + (b9*error_id_pre);

error_iq = iqref - iq;
uq = uq + (b8*error_iq) + (b9*error_iq_pre);

// --- Decoupling & Inverse Transform ---
md = (ud - iq*omegaL + vd) * 2 / vdc;
mq = (uq + id*omegaL + vq) * 2 / vdc;

// Inverse Park Transform (dq to abc)
ma = sin(thetaPLL)*md + cos(thetaPLL)*mq;
mb = sin(thetaPLL + 2*PI/3)*md + cos(thetaPLL + 2*PI/3)*mq;
mc = sin(thetaPLL - 4*PI/3)*md + cos(thetaPLL - 4*PI/3)*mq;

// PWM Duty Cycle Generation
refa = (ma + 1) / 2;
refb = (mb + 1) / 2;
refc = (mc + 1) / 2;
```

Listing 4: Current and Power Control Loops

5 Results and Conclusion

5.1 Controller Performance

The system was tested with a reference of $P_{ref} = 3000$ W and $Q_{ref} = 3000$ VAR.

- **Current Tracking:** The generated current references ($i_{d_{ref}}, i_{q_{ref}}$) stabilized at approximately 6.9 A, consistent with theoretical expectations for the given power levels.
- **Waveforms:** The oscilloscope results (Figure 3) display the grid voltage (Channel 1) and the injected current (Channel 2). The waveforms are sinusoidal and stable, indicating successful closed-loop control.

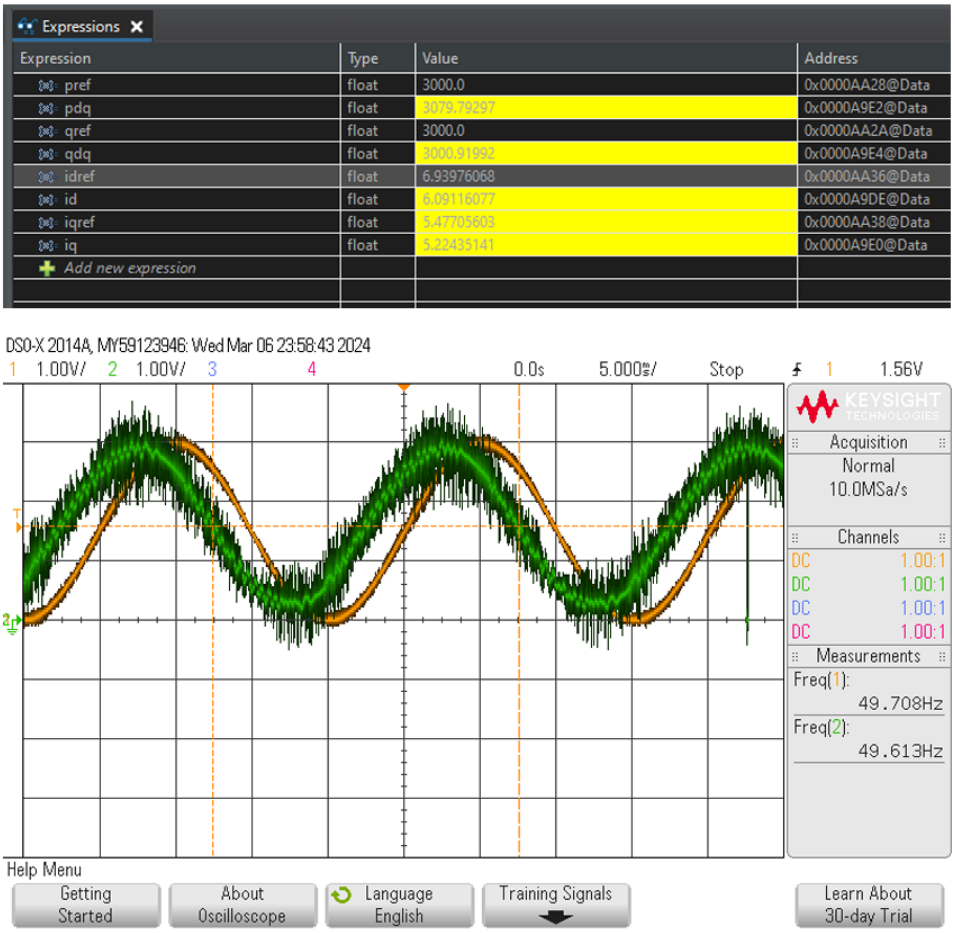

Figure 3: Closed-loop GCI operation. Channel 1 (Yellow): Grid Voltage, Channel 2 (Green): Injected Grid Current.

5.2 Conclusion

The project successfully demonstrated the implementation of a full digital control stack for a Grid-Connected Inverter on the C2000 F28379D MCU. The use of Processor-in-the-Loop simulation allowed for the verification of the PLL, current loops, and power loops in a safe environment. The results confirm that the controller can accurately regulate power injection into the grid with fast dynamic response and robust synchronization.