Design Guide: TIDA-01606

11-kW, Bidirectional Three-Phase Three-Level (T-type) Inverter and PFC Reference Design

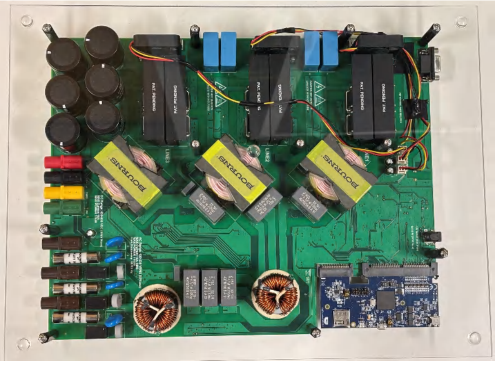
Description

This reference design provides an overview on how to implement a bidirectional three-level, three-phase, SiC-based active front end (AFE) inverter and power factor correction (PFC) stage. The design uses switching frequency up to 90 kHz and an LCL output filter to reduce the size of the magnetics. A peak efficiency of 98.6% is achieved. The design shows how to implement a complete three-phase AFE control in the DQ domain. This bidirectional converter enables both DC fast charging and vehicle-to-grid (V2G) applications.

Resources

TIDA-01606	Design Folder
TMS320F28379D, TMS320F280039C	Product Folder
UCC21710, UCC5350, AMC3306M05	Product Folder
TMCS1123, AMC0330R, AMC0381D	Product Folder
OPA4388, OPA397, UCC14341B	Product Folder
UCC33421, C2000WARE	Product Folder
TMDSCNCD28379D	Tool Folder
TMDSCNCD280039C	Tool Folder

Ask our TI E2E™ support experts

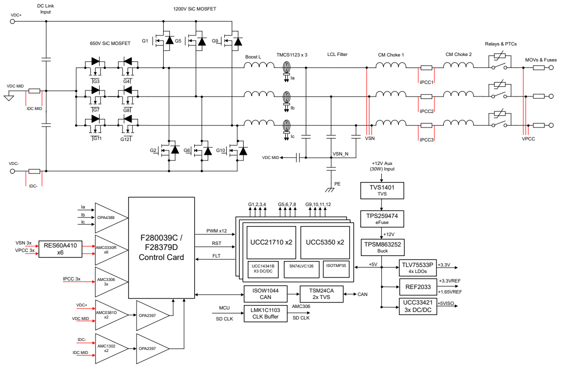


Features

- Rated nominal and maximum input voltage at 800-V and 900-V DC, maximum power 11-kW, 11-kVA at 400-V_{L-L} AC, 50 Hz or 60 Hz
- Compact output LCL filter with switching frequency of 90 kHz
- < 2.5% output current Total Harmonic Distortion (THD) at full load
- Isolated driver UCC21710 with reinforced isolation for driving high-voltage SiC MOSFET and UCC5350 for driving middle SiC MOSFET
- Isolated current sensing using TMCS1123 for load current control and monitoring
- TMS320F28379D and TMS320F280039C control cards for digital control with Trigonometric Math Unit (TMU) to accelerate Phase-Locked Loop (PLL) computation, comparator sub-system for protection implementation and Control Law Accelerator (CLA) to offload the control loops to the coprocessor

Applications

- DC fast charging station
- DC wallbox charger
- On-board (OBC) and wireless charger
- Power conversion system (PCS)
- String inverter
- Central inverter



1 System Description

Modern commercial scale solar inverters are seeing innovation on two fronts, which lead to smaller, higher efficiency products on the market:

- The move to higher voltage solar arrays
- Reducing the size of the onboard magnetics

By increasing the voltage to 1000-V or 1500-V DC from the array, the current can be reduced to maintain the same power levels. This reduction in current results in less copper and smaller power conducting devices required in the design. The reduction in di/dt also reduces the stress on electrical components. However, sustained DC voltages of > 1 kV can be difficult to design to, or even find components that can survive.

To compensate for the voltage stresses generated by high-voltage solar arrays, new topologies of solar inverters have been designed. Traditional half bridges block the full input voltage on each switching device. By adding additional switched blocking and conduction components, the overall stress on the device can be significantly reduced. This reference design shows how to implement a three-level converter. Higher level converters are also possible, further increasing the voltage handling capability.

Additional power density in solar electronics is also being enabled by moving to higher switching speeds in the power converters. As this design shows, even a modestly higher switching speed reduces the overall size requirement of the output filter stage—a primary contributor to the design size.

Traditional switching devices have a limit in how quickly these devices can switch high voltages, or more appropriately, the dV/dt ability of the device. This slow ramp up and down increases conduction loss because the device spends more time in a switching state. This increased switch time also increases the amount of dead time required in the control system to prevent shoot-through and shorts. The answer to this was developed in newer switching semiconductor technology like SiC and GaN devices with high electron mobility. This reference design uses SiC MOSFETs alongside TI's SiC gate driver technology to demonstrate the potential increase in power density.

Similarly for Onboard Chargers (OBC) higher power chargers (11 kW and 22 kW) are increasingly required. For which three phase PFC is necessary, this design shows implementation of three phase PFC using DQ control and presents the complete control loop model.

1.1 Key System Specifications

Table 1-1. Key System Specifications

PARAMETER	SPECIFICATIONS	DETAILS
Output power	11 kW	Section 2.3
Output voltage	Three-phase 400 V_{RMS} (Maximum V_{L-L})	Section 2.3
Output frequency	50 or 60 Hz	Section 2.3
Output current	16 A_{RMS} (maximum)	Section 2.3
Nominal input voltage	800-V DC	Section 2.3
Input voltage range	600-V to 900-V DC	Section 2.3
Inverter switching frequency	50–90 kHz	Section 2.3
Efficiency	98.6%	Section 2.3.1.5
THD	< 3% (11 kW)	
Power density	2.2 kW/L+	
Dimensions	27 cm × 35 cm × 5 cm	

2 System Overview

2.1 Block Diagram

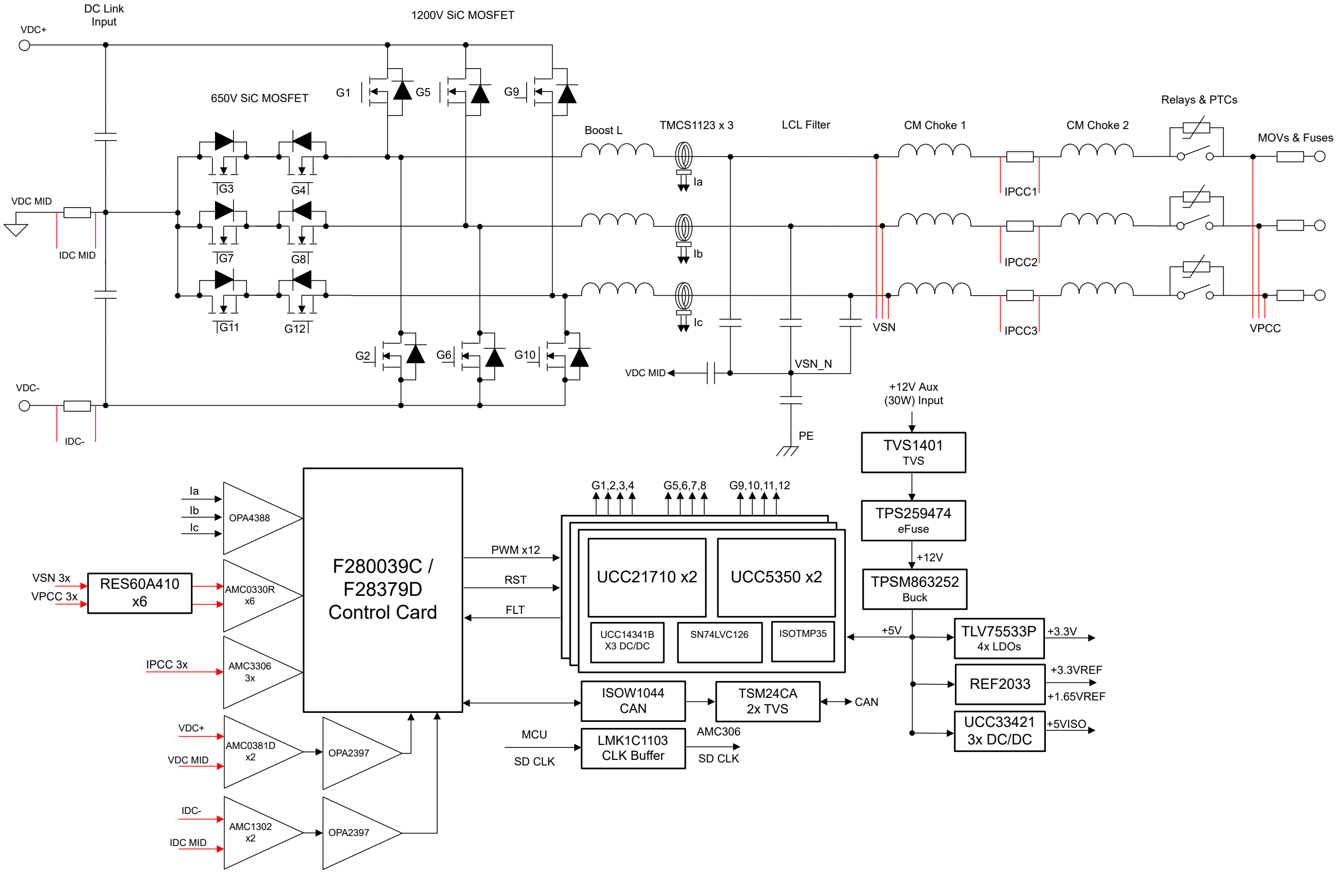


Figure 2-1. TIDA-01606 Block Diagram

This reference design is comprised of two separate boards that intercommunicate. The following boards work in tandem to form this three-phase inverter reference design:

- A power board, comprising all of the switching devices, gate drivers, LCL filter, sensing electronics, and power structure
- A TMS320F28379D control card or a TMS320F280039C to support the DSP

2.2 Highlighted Products

2.2.1 UCC21710

The UCC21710 device is a 5.7-kV$_{RMS}$, reinforced isolated gate driver for Insulated-Gate Bipolar Transistors (IGBT) and SiC MOSFETs with split outputs, providing 10-A source and 10-A sink current. The input side operates from a single 3-V to 5.5-V supply. The output side allows for a supply range from minimum 13 V to maximum 33 V. Two complementary CMOS inputs control the output state of the gate driver. The short propagation time of 130 ns provides accurate control of the output stage. UCC21710 integrates short circuit protection, detected via Overcurrent detection, with a fast response time needed to protect SiC MOSFETs.

- 150-kV/µs minimum common-mode transient immunity (CMTI)
- Split outputs to provide 10-A peak source and 10-A peak sink currents
- Short propagation delay: 90 ns (typ), 130 ns (max)
- 4-A active Miller clamp
- Output short-circuit clamp
- Soft turn off (STO) during short circuit
- Fault alarm upon desaturation detection is signaled on FLT and reset through RST
- Input and output undervoltage lockout (UVLO) with Ready (RDY) pin indication

- Active output pulldown and default low outputs with low supply or floating inputs
- 2.25-V to 5.5-V input supply voltage
- 15-V to 30-V output driver supply voltage
- CMOS compatible inputs
- Rejects input pulses and noise transients shorter than 40 ns
- Operating temperature: –40°C to +150°C ambient
- Isolation surge withstand voltage of 12800-V_{PK}

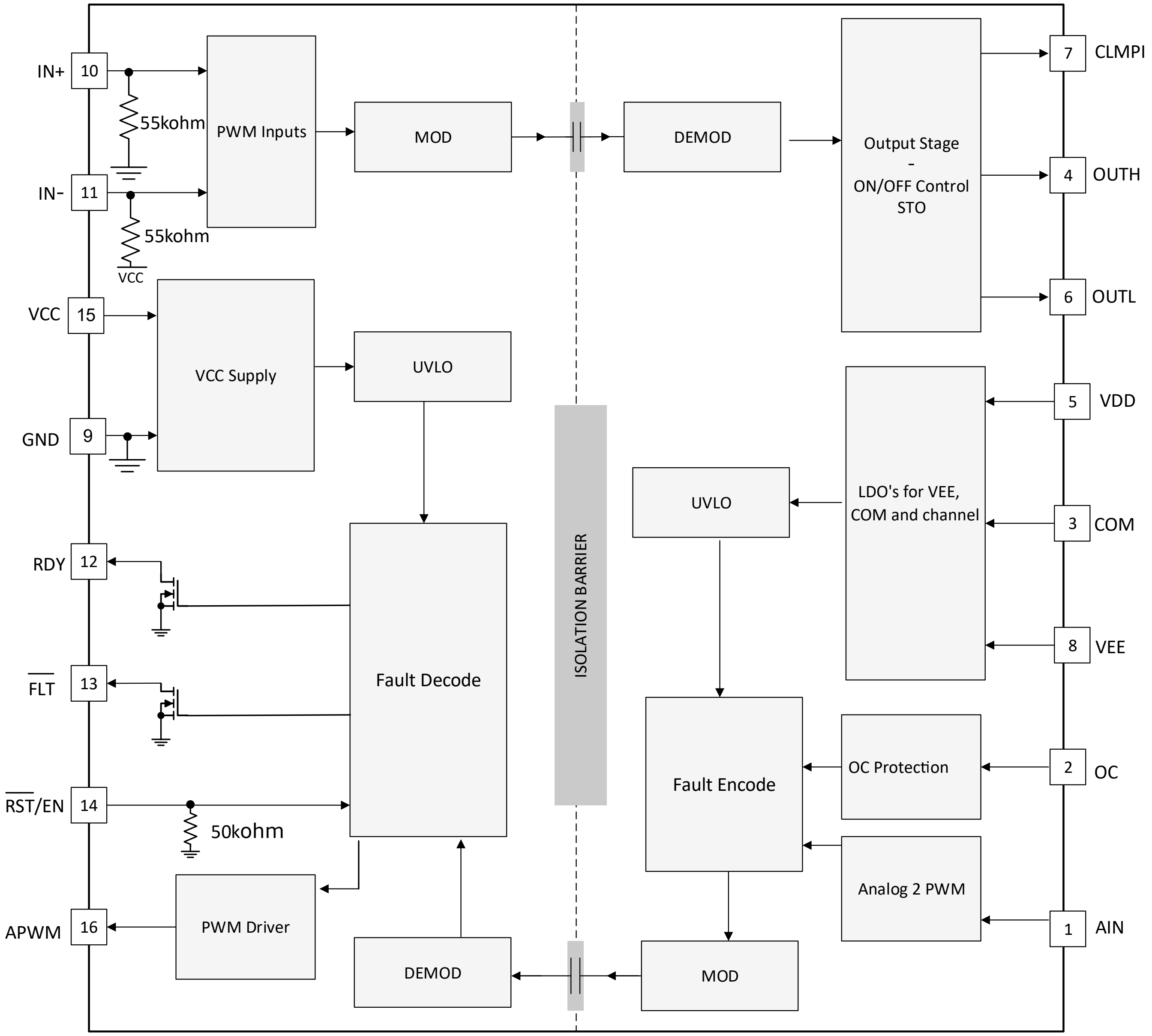


Figure 2-2. UCC21710 Functional Block Diagram

2.2.2 UCC5350

The UCC53x0 is a family of compact, single-channel, isolated IGBT, SiC, and MOSFET gate drivers with best-in-class isolation ratings and variants for pinout configuration and drive strength.

The UCC53x0 is available in an 8-pin SOIC (DWV) package. This package has a creepage and clearance of 8.5 mm and can support isolation voltage up to 5 kV_{RMS}, which is good for applications where reinforced isolation is needed. With these various options and wide power range, the UCC53x0 family is a good fit for motor drives and industrial power supplies.

- 3-V to 15-V input supply voltage
- 13.2-V to 33-V output driver supply voltage
- Feature options:
  - Split outputs (UCC5320S and UCC5390S)
  - UVLO with respect to MOSFET collector (UCC5320E and UCC5390E)
  - Miller clamp option (UCC5310M and UCC5350M)
- Negative 5-V handling capability on input pins
- 60-ns (typical) propagation delay for UCC5320S, UCC5320E, and UCC5310M
- 100-kV/µs minimum CMTI
- Isolation surge withstand voltage: 4242 V_{PK}
- Safety-related certifications:
  - 4242-V_{PK} isolation per DIN V VDE V 0884-10 and DIN EN 61010-1 (planned)
  - 3000-V_{RMS} isolation for 1 minute per UL 1577 (planned)
  - CSA Component Acceptance Notice 5A, IEC 60950-1 and IEC 61010-1 End Equipment Standards (Planned)
  - CQC Certification per GB4943.1-2011 (Planned)
- 4-kV ESD on all pins
- CMOS inputs
- 8-pin narrow body SOIC package
- Operating temperature: –40°C to +125°C ambient

2.2.3 TMS320F28379D

The Delfino™ TMS320F2837xD is a powerful 32-bit floating-point microcontroller unit (MCU) designed for advanced closed-loop control applications such as industrial drives and servo motor control; solar inverters and converters; digital power; transportation; and power line communications. Complete development packages for digital power and industrial drives are available as part of the powerSUITE and DesignDRIVE initiatives. While the Delfino product line is not new to the TMS320C2000™ portfolio, the F2837xD supports a new dual-core C28x architecture that significantly boosts system performance. The integrated analog and control peripherals also let designers consolidate control architectures and eliminate multiprocessor use in high-end systems.

- Dual-core architecture:
  - Two TMS320C28x 32-bit CPUs
  - 200 MHz
  - IEEE 754 single-precision floating-point unit (FPU)
  - Trigonometric math unit (TMU)
  - Viterbi/complex math unit (VCU-II)
- Two programmable control law accelerators (CLAs)
  - 200 MHz
  - IEEE 754 single-precision floating-point instructions
  - Executes code independently of main CPU
- On-chip memory
  - 512KB (256 kW) or 1MB (512 kW) of Flash (ECC-protected)
  - 172KB (86 kW) or 204KB (102 kW) of RAM (ECC-protected or parity-protected)
  - Dual-zone security supporting third-party development
- Clock and system control:
  - Two internal zero-pin 10-MHz oscillators
  - On-chip crystal oscillator
  - Windowed watchdog timer module
  - Missing clock detection circuitry
- 1.2-V core, 3.3-V I/O design
- System peripherals:
  - Two external memory interfaces (EMIFs) with ASRAM and SDRAM support
  - Dual six-channel direct memory access (DMA) controllers
  - Up to 169 individually programmable, multiplexed general-purpose input/output (GPIO) pins with input filtering
  - Expanded peripheral interrupt controller (ePIE)

  - Multiple low-power mode (LPM) support with external wakeup
- Communications peripherals:
  - USB 2.0 (MAC + PHY)
  - Support for 12-pin 3.3-V compatible universal parallel port (uPP) interface
  - Two controller area network (CAN) modules (pin-bootable)
  - Three high-speed (up to 50-MHz) SPI ports (pin-bootable)
  - Two multichannel buffered serial ports (McBSPs)
  - Four serial communications interfaces (SCI/UART) (pin-bootable)
  - Two I^{2}C interfaces (pin-bootable)
- Analog subsystem:
  - Up to four analog-to-digital converters (ADCs):
    - 16-bit mode
      - 1.1 MSPS each (up to 4.4-MSPS system throughput)
      - Differential inputs
      - Up to 12 external channels
    - 12-bit mode
      - 3.5 MSPS each (up to 14-MSPS system throughput)
      - Single-ended inputs
      - Up to 24 external channels
    - Single sample-and-hold (S/H) on each ADC
    - Hardware-integrated post-processing of ADC conversions:
      - Saturating offset calibration
      - Error from setpoint calculation
      - High, low, and zero-crossing compare, with interrupt capability
      - Trigger-to-sample delay capture
  - Eight windowed comparators with 12-bit digital-to-analog converter (DAC) references
  - Three 12-bit buffered DAC outputs
- Enhanced control peripherals:
  - 24 pulse width modulator (PWM) channels with enhanced features
  - 16 high-resolution pulse width modulator (HRPWM) channels:
    - High resolution on both A and B channels of eight PWM modules
    - Dead-band support (on both standard and high resolution)
  - Six enhanced capture (eCAP) modules
  - Three enhanced quadrature encoder pulse (eQEP) modules
  - Eight sigma-delta filter module (SDFM) input channels, two parallel filters per channel:
    - Standard SDFM data filtering
    - Comparator filter for fast action for out of range

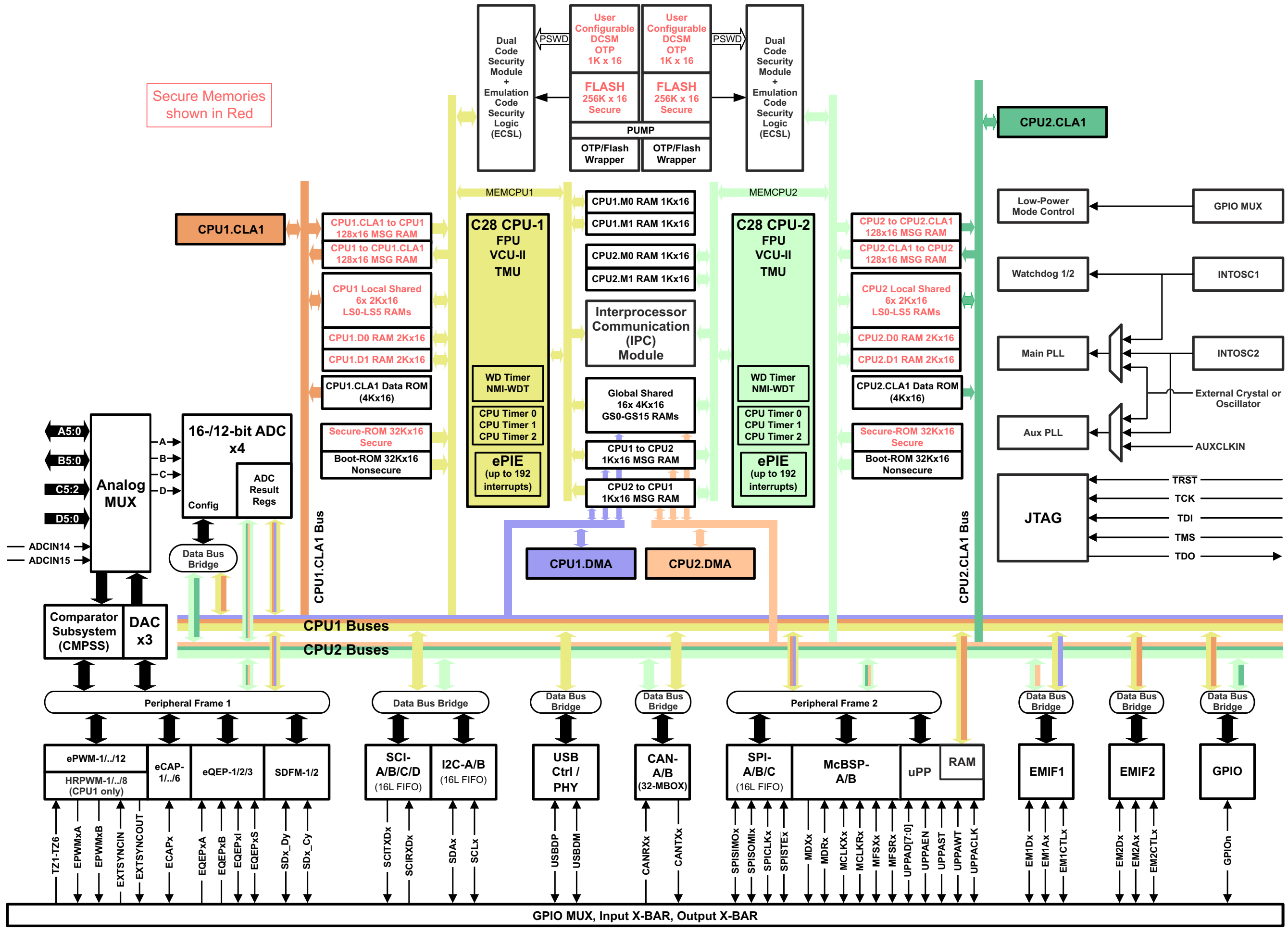


Figure 2-3. TMS320F28379D Functional Block Diagram

2.2.4 AMC3306M05

The AMC3306M05 is a precision, isolated delta-sigma (ΔΣ) modulator, optimized for shunt-based current measurements. The fully integrated, isolated DC/DC converter allows single-supply operation from the low-side of the device which makes the device a unique answer for space-constrained applications. The reinforced capacitive isolation barrier is certified according to VDE V 0884-11 and UL1577 and supports a working voltage of up to 1.2 kV_{RMS}.

2.2.5 OPA4388

The OPAx388 (OPA388, OPA2388, and OPA4388) series of precision operational amplifiers are ultra-low noise, fast-settling, zero-drift, zero-crossover devices that provide rail-to-rail input and output operation. These features and excellent AC performance, combined with only 0.25 µV of offset and 0.005 µV/°C of drift overtemperature, make the OPAx388 a great choice for driving high-precision, analog-to-digital converters (ADCs) or buffering the output of high-resolution, digital-to-analog converters (DACs). This design results in excellent performance when driving ADCs without degradation of linearity. The OPA388 (single version) is available in the VSSOP-8, SOT23-5, and SOIC-8 packages. The OPA2388 (dual version) is offered in the VSSOP-8 and SO-8 packages. The OPA4388 (quad version) is offered in the TSSOP-14 and SO-14 packages. All versions are specified over the industrial temperature range of –40°C to +125°C.

2.2.6 TMCS1123

The TMCS1123 is a galvanically isolated Hall-effect current sensor with industry-leading isolation and accuracy. An output voltage proportional to the input current is provided with excellent linearity and low drift at all sensitivity options. Precision signal conditioning circuitry with built-in drift compensation is capable of less than 1.4% maximum sensitivity error over temperature and lifetime with no system-level calibration, or less than 0.9% maximum sensitivity error including both lifetime and temperature drift with a one-time calibration at room temperature.

2.2.7 AMC0330R

The AMC0x30R-Q1 is a precision, galvanically isolated amplifier with a ±1 V, high impedance input and single-ended, ratiometric output. The high-impedance input is optimized for connection to high-impedance resistive dividers or other voltage signal sources with high output resistance.

2.2.8 AMC0381D

The AMC0381D-Q1 is a precision, galvanically isolated amplifier with a high-voltage DC, high impedance input, and fixed-gain, differential output. The input is designed to connect directly to a high-voltage signal source.

2.2.9 UCC14341

UCC14341 is a high-isolation voltage DC/DC module designed to provide power to IGBT or SiC gate drivers. The module integrates a transformer and DC/DC controller with a proprietary architecture to achieve high efficiency with very low emissions. The high-accuracy output voltages provide higher system efficiency without over-stressing the power device gate.

2.2.10 UCC33421

UCC33421-Q1 is an automotive qualified DC/DC power module with integrated transformer technology designed to provide 1.5 W of isolated output power. The device supports an input voltage operation range of 4.5 V to 5.5 V and regulates 5.0 V output voltage with a selectable headroom of 5.5 V.

2.3 System Design Theory

2.3.1 Three-Phase T-Type Inverter

2.3.1.1 Architecture Overview

To understand the impetus behind a three level t-type inverter, some background on a traditional two-level inverter is required. Figure 2-4 shows a typical implementation of this architecture.

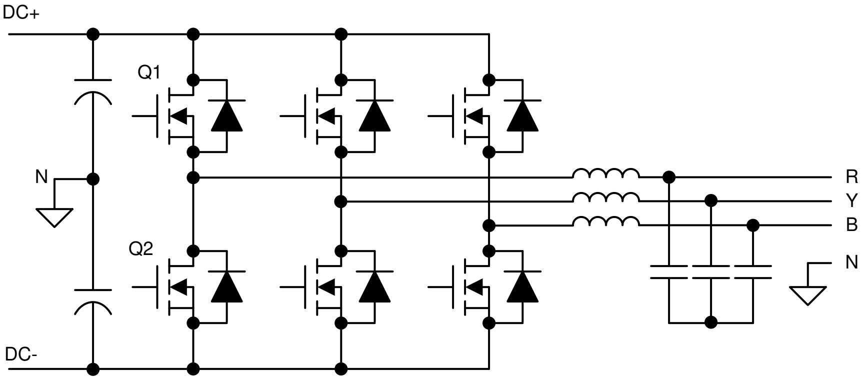


Figure 2-4. Two-Level, Three-Phase Inverter Architecture

To simplify the analysis, a single leg can be isolated.

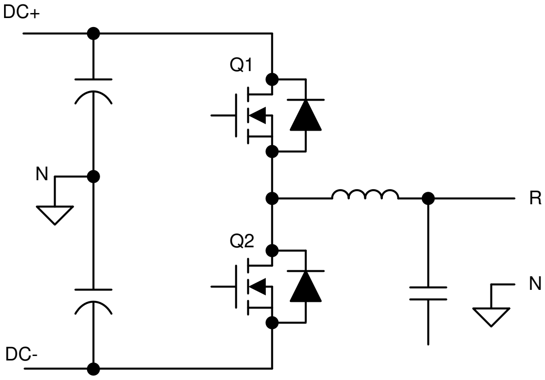


Figure 2-5. Two-Level, Single-Phase Inverter Leg

In this example, the two switching devices as a pair have four possible conduction states, independent of the other phases.

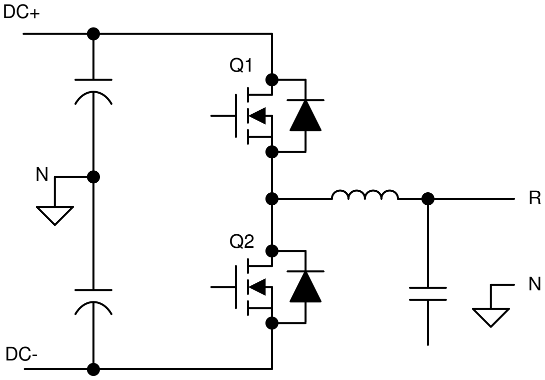


Figure 2-6. Q1 and Q2 off

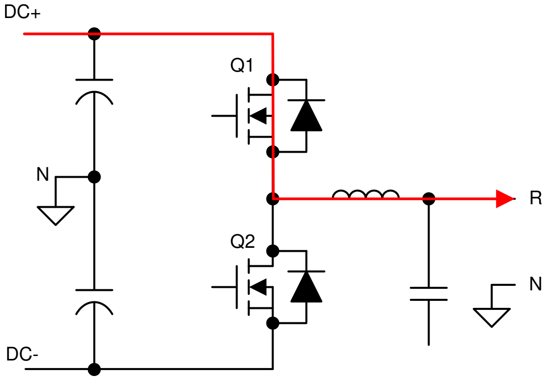


Figure 2-7. Q1 on, and Q2 off

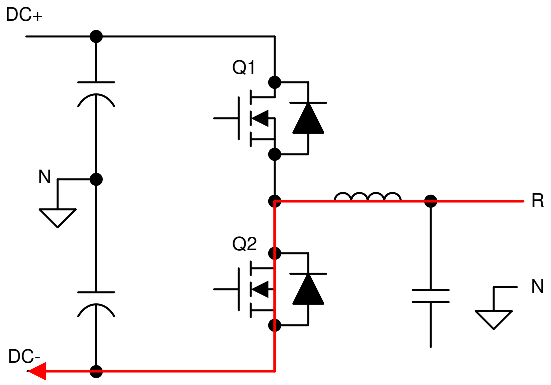


Figure 2-8. Q1 off, and Q2 on

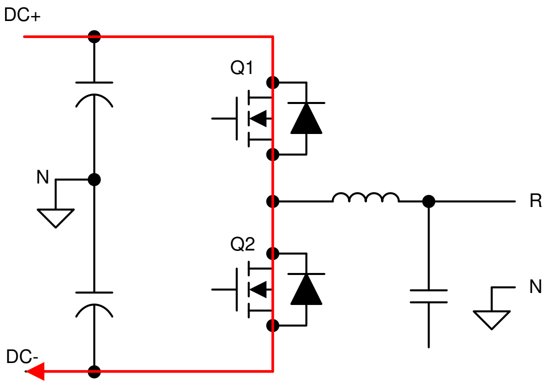


Figure 2-9. Q1 and Q2 on (Invalid)

By observing the current path through the inverter, each switching device must be capable of blocking the full DC link voltage present between DC+ and DC–. In traditional low-voltage systems (< 600 V), this capability is fairly trivial with common off-the-shelf IGBTs. However, if the DC link voltage is pushed higher to increase the power throughput without increasing current, as is a common trend in power electronics, this limitation puts an upper level on the supported voltage ranges.

Additionally, the increased voltage does result in increased switching losses in the traditional IGBTs. The low dV/dt exacerbates in these devices, even when able to support the higher voltages. This dV/dt is what determines how quickly one device can transition from on to off (or vice versa), thus dictating the dead time between each of these states. An elongated switch time or dead time means the switches spend less time at full conduction, resulting in decreased efficiency.

These two primary drawbacks of a two-level inverter are what drives the implementation in this design.

The next step up from a standard two-level inverter is a T-type three-level inverter. This type is implemented by inserting two back-to-back switching devices between the switch node and the neutral point of the DC link created by the bulk input capacitors. These two switch devices are placed in a common emitter configuration so that current flow can be controlled by switching one or the other on or off. This configuration also enables both of them to share a common bias supply as the gate-emitter voltage is identically referenced. Figure 2-10 shows a simplified view of the implementation.

Note

The E6 hardware middle switches are configured as common source switches. However, the E7 hardware middle switches are configured as common drain switches. The following T-type example illustrations are done using common source configuration.

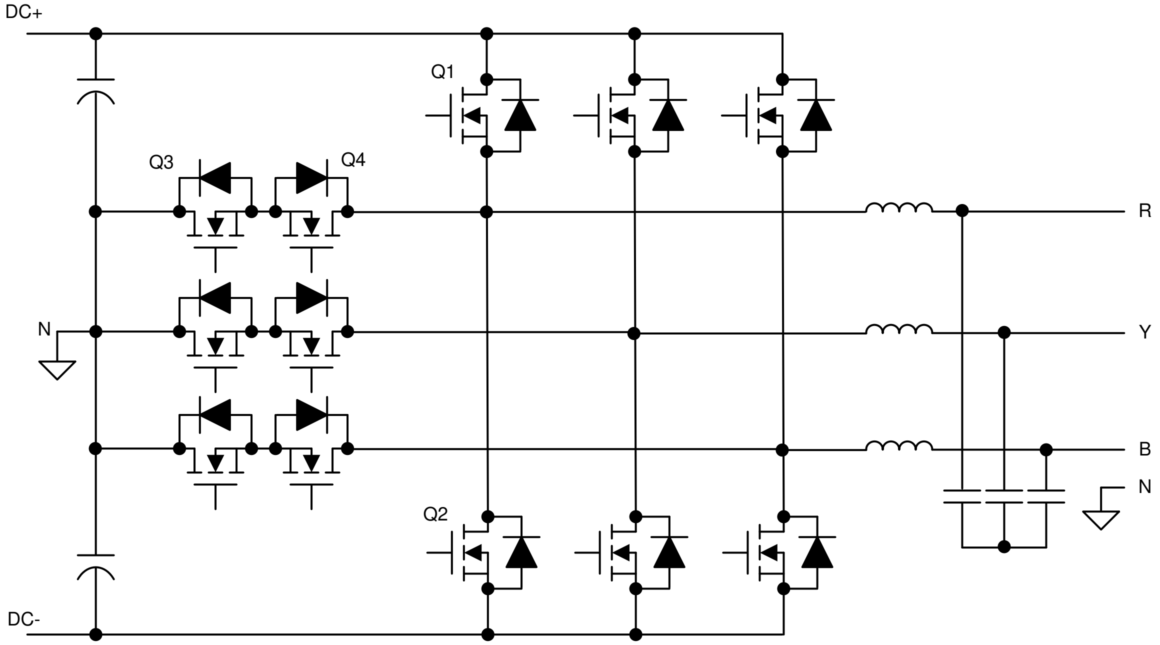


Figure 2-10. Three-Level T-Type, Three-Phase Inverter Architecture

To assist in understanding the benefits of the architecture, the inverter is again reduced to a single leg.

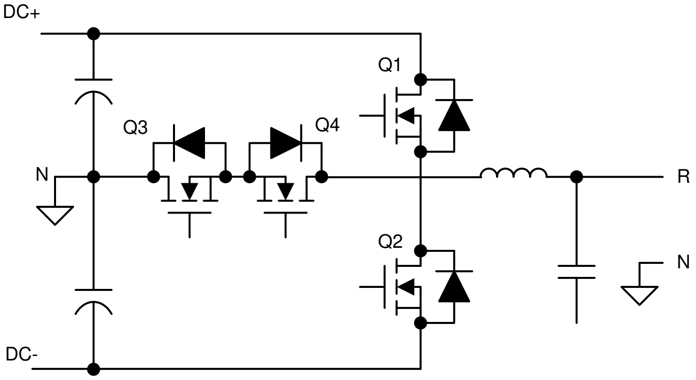


Figure 2-11. Three-Level T-Type, Single-Phase Inverter Leg

Adding two extra switching devices complicates the control of the system, but the same process of evaluating current flow during various modulation points illustrates the architecture benefits. Additionally, a simplified commutation scheme can be demonstrated, illustrating that control of a T-type inverter is not substantially more difficult than a traditional two-level architecture.

A single leg has three potential connection states: DC+, DC–, or N. This connection can be accomplished by closing Q1, closing Q3 and Q4, and closing Q2, respectively. However, this scheme depends on the current path in the system. Rather, for a DC+ connection, Q1 and Q3 can be closed, Q2 and Q4 for a neutral connection, and Q2 and Q4 for a DC– connection. This scheme acts independent of current direction as shown in the following figures.

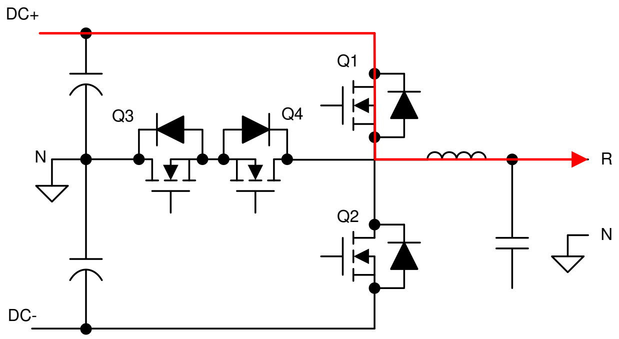


Figure 2-12. Q1 on, Q2 off, Q3 on, and Q4 off

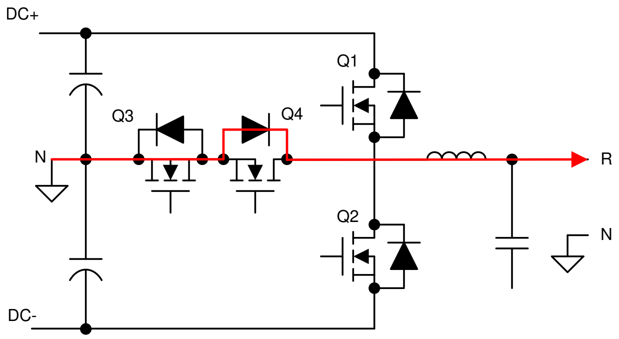


Figure 2-13. Q1 off, Q2 off, Q3 on, and Q4 off

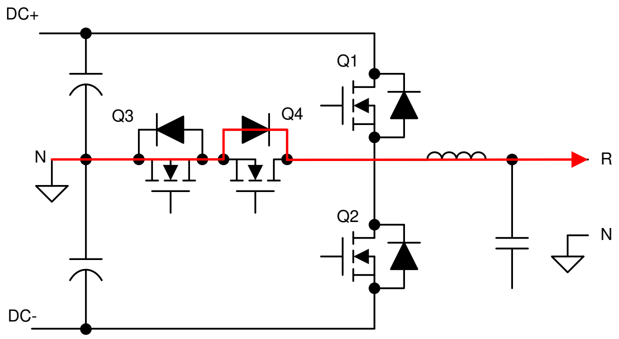


Figure 2-14. Q1 off, Q2 off, Q3 on, and Q4 on

This example starts with the output phase connected to DC+ by closing Q1 and Q3, resulting in current output from the system. To transition to an N connection, Q1 is opened and after a dead-time delay, and Q4 is closed. This setup allows current to naturally flow through Q3 and the diode of Q4.

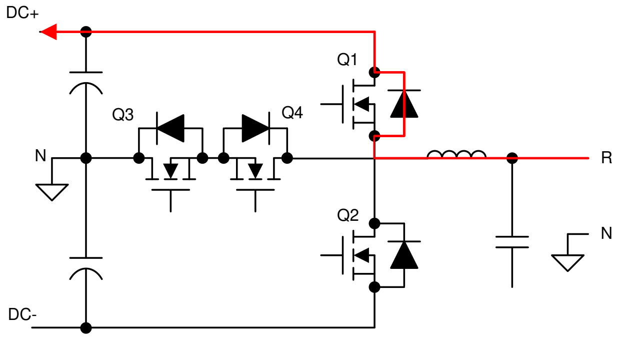


Figure 2-15. Q1 on, Q2 off, Q3 on, and Q4 off

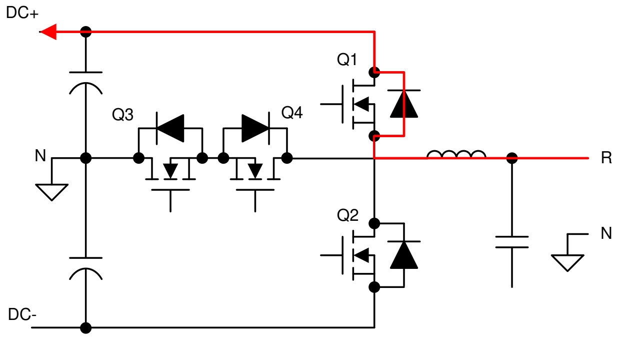


Figure 2-16. Q1 off, Q2 off, Q3 on, and Q4 off

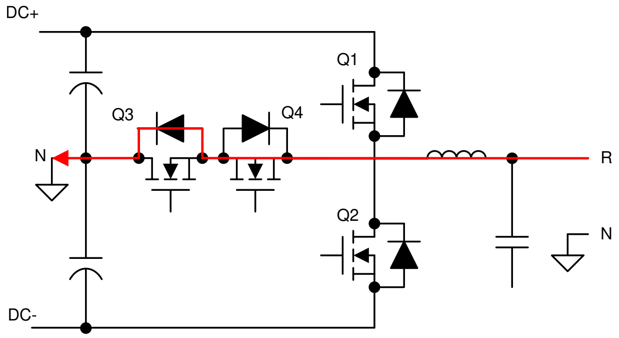


Figure 2-17. Q1 off, Q2 off, Q3 on, and Q4 on

For a negative current, the same sequence can be used. Once Q4 is closed, current then flows through Q4 and the diode of Q3 rather than the diode of Q1.

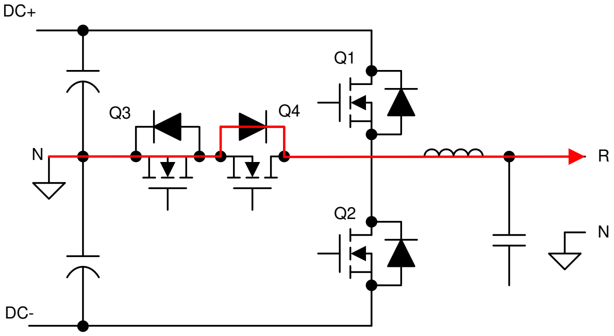


Figure 2-18. Q1 off, Q2 off, Q3 on, Q4 on

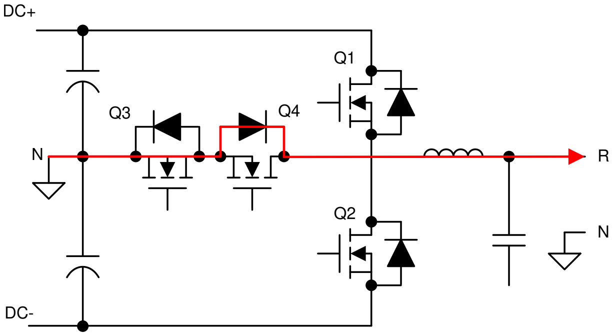


Figure 2-19. Q1 off, Q2 off, Q3 on, Q4 off

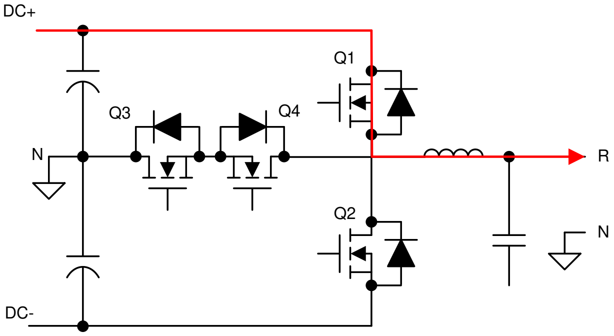


Figure 2-20. Q1 on, Q2 off, Q3 on, Q4 off

A similar natural current flow can be observed when connecting the output leg from N to DC+ with a positive current. Q3 and Q4 start closed with a full N connection. Q4 is switched off, but current still flows through the associated diode. Closing Q1 now naturally switches the current flow from N to DC+.

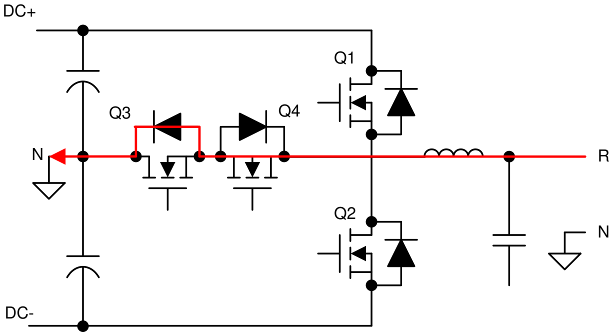


Figure 2-21. Q1 off, Q2 off, Q3 on, Q4 on

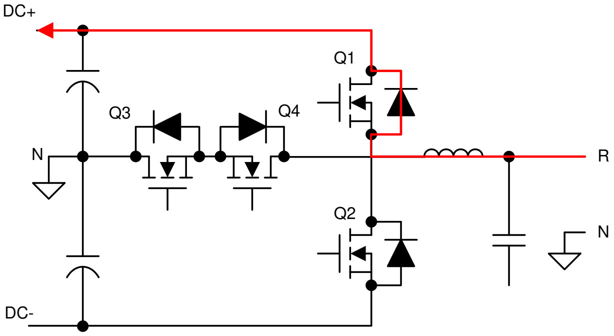


Figure 2-22. Q1 off, Q2 off, Q3 on, Q4 off

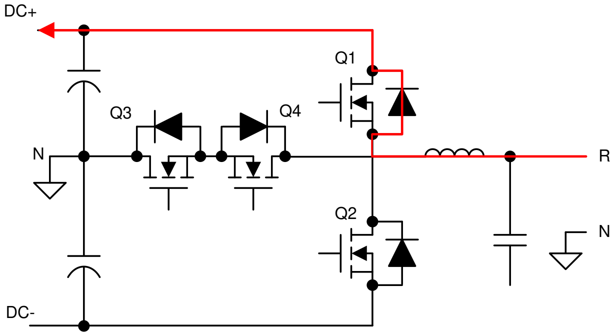


Figure 2-23. Q1 on, Q2 off, Q3 on, Q4 off

As in the earlier example when moving from a DC+ to N connection on a negative current, the same scheme can also be used here for a positive current. Q3 and Q4 begin closed, conducting current into N. Q4 is opened, causing current to flow through the diode of Q1. Lastly, Q1 is closed, and current continues flowing in the same direction.

All four of these transition states (DC+ to N, N to DC+, with both forward and reverse current) all share two simple switching schemes. This also holds true for transitions to and from DC– through Q2. By maintaining this scheme through all switching cycles, a simple dead-zone delay between switching events is all that is needed to avoid shoot-though; however, additional protection can be added in the control software with relative ease.

An additional benefit from this modulation scheme is that Q3 and Q4 never switch at the same time. This benefit reduces voltage stress on the devices as well as the power rating of the bias supply to drive these devices effectively. As previously mentioned, Q3 and Q4 can share a single supply sized for one driver rather than two.

Q1 and Q2 still need to block the full DC link voltage as the inverters do in the traditional architecture. To use a higher DC bus voltage, full-voltage FETs must still be in place; however, because the inverters are back to back and do not switch at the same time, the two switches on the center leg can be at a lower rating.

2.3.1.2 LCL Filter Design

Any system of power transfer to the grid is required to meet certain output specifications for harmonic content. In voltage sourced systems like modern photo-voltaic inverters, a high-order LCL filter typically provides sufficient harmonic attenuation, along with reducing the overall design size versus a simpler filter design. However, due to the higher order nature, take some care in the design to control resonance. Figure 2-24 shows a typical LCL filter.

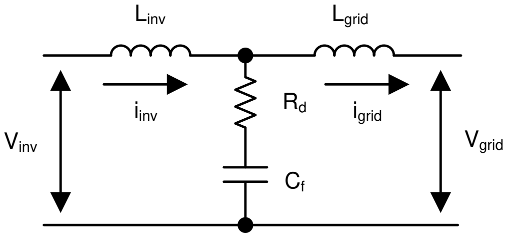


Figure 2-24. LCL Filter Architecture

One of the key benefits of using SiC MOSFETs (as this reference design does) is the ability to increase the switching frequency of the power stage significantly versus traditional Si-based switching elements. This increased switching frequency has a direct impact on the output filter resonant design of the inverter, which needs to be accounted for. To make sure that the filter is designed correctly around this switch frequency, this known mathematical model is used in this design.

The primary component is the inverter inductor, or L_{inv}, which can be derived using Equation 1:

$$L_{inv} = \frac{V_{DC}}{8 \times f_{SW} \times I_{grid\_rated} \times \%ripple} \tag{1}$$

where,

- f_{SW} is the PWM switching frequency
- I_{grid_rated} is the grid RMS current rating
- % ripple is the ripple current percentage of the rated grid current

The sizing of the primary EMI filter capacitor, C_f, is determined by Equation 2:

$$\frac{1}{L_{inv} \times (2\pi \times f_{SW})^2} < < C_f < \frac{\% \times Q_{rated}}{2\pi \times f_{grid} \times V_{grid}^2} \tag{2}$$

where

- % Q_{rated} is the percentage of the rated reactive power to limit the C_f capacitor
- F_{grid} is the grid electrical frequency
- V_{grid} is the grid phase voltage

Equation 3 shows an example calculation:

$$\frac{1}{130\mu F \times (2\pi \times 90kHz)^2} < < C_f < \frac{5\,\% \times \frac{11kVA}{3}}{2\pi \times 60Hz \times 230V^2} \tag{3}$$
$$24nF \quad < < C_f < \quad 9.2\mu F$$

In E7, 4.7 µF was selected for C_f. This equates to approximately 6.4 kHz for the cutoff frequency.

2.3.1.3 Inductor Design

Note

The boost inductors used in E7 are custom Bourns inductors. The key specifications for the boost inductor are the inductance, current rating, dimensions, and operating temperature. TI recommends working with experienced magnetic manufacturers (for example, Bourns Inc.) to help with designing custom inductors.

2.3.1.4 SiC MOSFETs Selection

As shown in the *Architecture Overview*, the main switching device needs to support the full switching voltage. To support the 1000-V DC link voltage of this design, use 1200-V FETs; however, at this voltage, the migration to SiC is necessitated by several factors:

- The switching speed of a 1200-V SiC MOSFET is significantly faster than a traditional IGBT, leading to a reduction in switching losses.
- The reverse recovery charge is significantly smaller in the SiC MOSFET, resulting in reduced voltage and current overshoot.
- A lower temperature dependence at full load due to reduced conduction loss.

The middle switches are only exposed to half of the DC link voltage, or 500 V in this design. As such, a 650-V device is acceptable. A full SiC design provides the best performance due to these same features. For this design, the reverse recovery loss and voltage overshoot limits the device selection. As such, a 1200-V SiC MOSFET + 650-V MOSFET design is used.

Conduction loss is mainly determined by the $R_{DS(on)}$ of the 1200-V SiC MOSFET and the on $R_{DS(on)}$ of the 650-V SiC MOSFET. The 75-mΩ SiC devices have a good high-temperature performance, and the $R_{DS(on)}$ only increases 40% at 150°C junction temperature. With the high temperature I-V curve in the data sheet, calculate the conduction loss on the devices.

Switching loss is a function of the switching frequency and switching energy of each switching transient, the switching energy is related with device current and voltage at the switching transient. Using the switching energy curve in the data sheet, one can estimate the total switching loss.

Similarly, the conduction loss and switching loss can be estimated for all the devices and efficiency can be estimated. With the thermal impedance information of the thermal system design, the proper device rating can be selected. The 1200-V/75-mΩ SiC MOSFET and 650-V/60-mΩ SiC MOSFET is a good tradeoff among thermal, efficiency and cost.

2.3.1.5 Loss Estimations

The primary source of lost efficiency in any inverter is going to be a result of the losses incurred in the switching devices. These losses are broken into three categories for each device:

- Conduction loss: When the device is on and conducting normally
- Switching loss: When the device is switching between states
- Diode conduction loss: Related to voltage drop and current when in conduction

Each of these are dictated by a unique equation, and can be determined from the device data sheet and design parameters that were already set.

Conduction loss is driven by the on-time of the FET, the switched current, and the on-resistance:

$$P_{cond\_loss} = \frac{1}{T}\int_0^T V_{ce}(t) \times I_c(t) \times D_Q(t)dt \tag{4}$$

where

- V_{ce} is the conduction voltage drop
- I_c is the conduction current
- D_Q is the duty cycle
- T represents one modulation cycle

Switching loss is determined by the switching energy of the device and the switching voltage at a selected test point. Determine the value of the switching energy from the device data sheet using the value of the designed external gate resistor. The remainder of the values needed were determined earlier in the design phase.

$$P_{sw\_loss} = \frac{(E_{on} + E_{off}) \times I_{peak} \times f_{SW} \times V_{DC}}{\pi \times I_{avg} \times V_{nom}} \tag{5}$$

Figure 2-25 shows an example of the graph used to extract the switching energy values from the device data sheet is shown for an C3M0060065D SiC MOSFET.

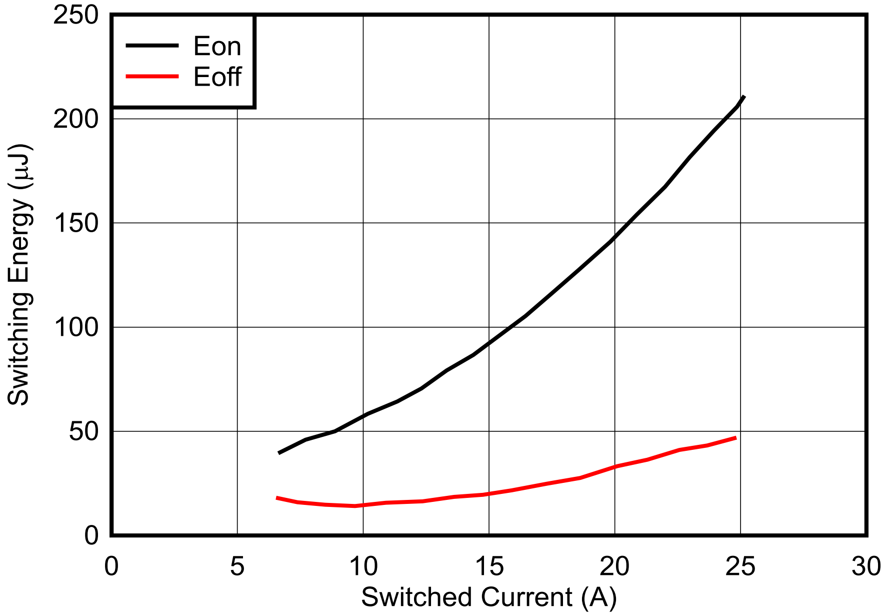


Figure 2-25. Switching Energy vs Switched Current for C3M0060065D

The diode conduction loss is similarly calculated using known values:

$$P_{sw\_diode} = \frac{1}{T}\int_0^T Vf(t) \times If(t) \times D_D(t)dt \quad (6)$$

where

- V_f is the voltage drop
- I_f is the diode current
- D_D is the duty cycle
- T represents one modulation cycle

Using these three equations, the expected losses of the design are computed for both the SiC MOSFETs as shown in Table 2-1.

Table 2-1. Expected Losses of Switching Devices

PARAMETER	C3M0075120D (Q1)	C3M0060065D (Q3)
Conduction loss	5.76 W	4.5 W
Switching loss	1.8 W	1.13 W
Diode loss	0 W	0 W
Total	7.56 W	5.63 W

The final piece of the total system loss estimation is the inductor losses. These losses are determined using the value of the inductor DC and AC resistance and expected inductor current from Section 2.3.1.3.

$$Pind\_loss = I_{ind\_ac\_rms}{}^2 \times R_{DC} + I_{ind\_ripple\_rms}{}^2 \times R_{AC} \quad (7)$$

$$P_{ind\_loss} = (0.81\ A)^2 \times 0.024\ \Omega + (15.155)^2 \times 0.076\ \Omega = 5.64\ W \quad (8)$$

The total major energy loss for this design is then:

$$P_{loss\_total} = 6 \times \left(P_{Q1\_total} + P_{Q3\_total}\right) + 3 \times P_{int\_loss} \quad (9)$$

$$P_{loss\_total} = 6 \times \left(5.631\text{ W} + 7.56\text{ W}\right) + 3 \times 5.64\text{ W} = 96.102\text{ W} \tag{10}$$

Use Equation 10 to determine the total expected inverter efficiency. Note that this is an estimation, but the estimate allows the design to be validated up to this point.

$$\eta = \frac{P_{out}}{P_{out} + P_{loss\_total}} \times 100 \tag{11}$$

$$\eta = \frac{10\text{ kW}}{10\text{ kW} + 96.102\text{ W}} = 99.048\% \tag{12}$$

2.3.2 Voltage Sensing

Voltage sensing happens at two points in the inverter signal path to aid in control: before and after the primary output relay. By enabling measurement on both sides of the relay, the control system can lock into the grid voltage and frequency before connecting, thus preventing any mismatch issues.

Both sensing topologies are similar. First, V_PCC_N and V_SN_N are used as virtual neutral using a resistor network. The high-voltage signal is attenuated using a series of large value resistances. An offset of 1.65 V is added to the attenuated phase voltage to center the voltage relative to the ADC VREF (that is, 3.3 V).

2.3.3 Current Sensing

Critical to attaining a closed-loop control system is accurate current measurement of the inverter. In this design, current measurement is done at two locations with different sensing technologies. The first location is on the grid output using shunt resistors. Because the output is high voltage and the controller needs to remain isolated, the AMC3306M05 reinforced modulator is used to measure the resistor voltage drop. To keep system losses low, the AMC3306M05 has a ±50-mV input range. When compared to other devices with a typical input range of ±250 mV, the total power loss across the shunt is significantly reduced.

Sizing the shunt resistor for this design is a trade-off between sensing accuracy and power dissipation. A 0.002-Ω shunt provides a ±50-mV output signal at the approximate ±25-A output inverter but also only generates 0.5 W of heat at full load. When choosing an actual device, select a high accuracy value to eliminate the need to calibrate each sensor path.

The voltage across the shunt resistor is fed into the AMC3306M05 sigma-delta modulator, which generates the sigma-delta stream that is decoded by the SDFM demodulator present on the C2000™ MCU. The clock for the modulator is generated from the eCAP peripheral on the C2000 MCU, and the AMC3306M05 data are decided using the built-in SDFM modulator.

The second location is a Hall-effect sensor TMCS1123, which is used to sense the current through the inductor. OPA4388 is used to filter the outputs and center all three phase measurements on the same 1.65 V offset reference.

2.3.4 System Auxiliary Power Supply

A 12-V, 2-A auxiliary power supply is required to power the supporting devices including isolated bias supplies, fans, relays, low-voltage analog and digital circuits, and the C2000 control card. There are a number of onboard voltage converters and regulators to provide the 5 V, 3.3 V, and 1.65 V rails from the 12-V input. TI recommends using a 12-V power supply with overvoltage and overcurrent protection features for added protection during testing.

2.3.5 Gate Drivers

2.3.5.1 1200-V SiC MOSFETs

VCC and GND are the supply pins for the input side of the UCC21710 device. The supply voltage at VCC can range from 3.0 V to 5.5 V with respect to GND. VDD and COM are the supply pins for the output side of the UCC21710 device. VEE is the supply return for the output driver and COM is the reference for the logic circuitry.

The supply voltage at VDD can range from 15 V up to 30 V with respect to VEE. The PWM is applied across the IN+ and IN– pins of the gate driver.

On the secondary-side of the gate driver, gate resistors (for example, R203 and R204) control the gate current of the switching device. The DESAT fault detection prevents any destruction resulting from excessive collector currents during a short-circuit fault. To prevent damage to the switching device, the UCC21710 slowly turns off the SiC MOSFET in the event of a fault detection. A slow turnoff makes sure the overcurrent is reduced in a controlled manner during the fault condition. The DESAT diode (for example, D10) conducts the bias current from the gate driver, which allows sensing of the MOSFET-saturated collector-to-emitter voltage when the SiC MOSFET is in the ON condition.

2.3.5.2 650-V SiC MOSFETs

The UCC5350 primary side is powered by a 3.3-V rail. A 0.1-µF ceramic capacitor is placed close to the VCC1 pin for noise decoupling. The positive-going UVLO threshold on the supply is 2.6 V and the negative-going threshold is 2.5 V.

The PWM input to the gate driver is provided by the controller PWM output peripheral. Dead time must be inserted between the low-side and high-side PWM signals to prevent both switches turning on at the same time. The signal is single ended and is filtered by RC low-pass filter before connecting to the gate driver input. The filter attenuates high-frequency noise and prevents overshoot and undershoot on the PWM inputs due to longer tracks from the controller to the gate driver. The inverting PWM input IN– is not used in the design and is connected to primary side ground.

A 3.3-Ω gate resistor (for example, R258) is used for MOSFETs turn-on and turn-off. A 10-kΩ resistor (for example, R85) is connected across the MOSFET gate to collector pins close to the MOSFET on the main power board. This connection makes sure that the MOSFET remains in the off state in case the gate driver gets disconnected from the MOSFET due to faults.

2.3.5.3 Gate Driver Bias Supply

The gate drivers rely on UCC14341B isolated bias voltages (+15 V and –4 V) to drive the gates across the high-voltage barrier. In this architecture, there are nine total gate drivers and the default number of isolated bias supplies is nine. There is a possibility to reduce the number of isolated bias supplies to five. For the nine-supply option:

1. 3 × low-side supplies referenced to REF_VDC-n, where n = 1, 2, and 3
2. 3 × middle-side supplies referenced to REF_MIDn, where n = 1, 2, and 3
3. 3 × switching-node supplies referenced to REF_SN_Ln, where n = 1, 2, and 3

For the five-supply option:

1. 1 × low-side supply referenced to REF_VDC-n, where n = 1
2. 1 × middle-side supply referenced to REF_MIDn, where n = 1
3. 3 × switching-node supplies referenced to REF_SN_Ln, where n = 1, 2, and 3

For the five-supply option to work, assume there is a negligible voltage drop between the reference points across all three phases.

2.3.6 Control Design

Terminology:

V_{bus} or V_{dc} bus voltage for the inverter

L_i and R_i inductance of the inverter side inductor and series resistance

L_g and R_g inductance of the grid side inductor, and series resistance

C_f and R_f capacitance value and series resistor plus any damping resistor

v_{i_a},v_{i_b},v_{i_c} output voltage from the three phase bridge, this voltage is what is controlled using the duty cycle control of the three phase bridge

Therefore, for control purposes it is assumed now only the modulation needs to change between 2-level and 3-level inverter for the power stage and control design can remain the same. Where ,D_a, D_b, D_c, are the control variable generated such that the output voltage of the inverter can be represented as Equation 13:

$$v_{i\_a} = D_a \times {}^{V_{DC}}\!/_{2} \tag{13}$$

i_{i_a}, i_{i_b}, i_{i_c} current through the inverter side inductor

v_{x_a}, v_{x_b}, v_{x_c} voltage across the filter capacitor

i_{g_a}, i_{g_b}, i_{g_c} voltage across the filter capacitor

v_{g_a}, v_{g_b}, v_{g_c} grid frequency

2.3.6.1 Current Loop Design

For the inverter filter shown in Figure 2-26, using KCL and KVL Equation 14 can be written.

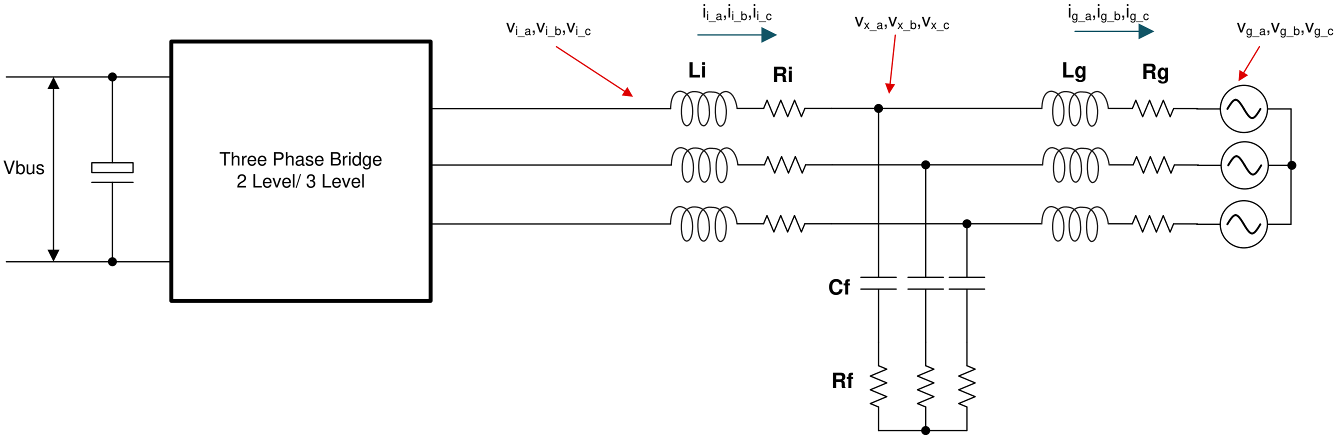


Figure 2-26. Inverter Model

$$v_{i\_a} - L_i \frac{di_{i\_a}}{dt} - R_i i_{i\_a} = v_{x\_a} \tag{14}$$

Upon re-arranging, Equation 14 can be written as Equation 15:

$$\frac{di_{i\_a}}{dt} = \frac{1}{L_i} v_{i\_a} - \frac{1}{L_i}\left(R_i i_{i\_a} + v_{x\_a}\right) \tag{15}$$

Similarly on another node, using KCL and KVL, Equation 16 can be written as Equation 16:

$$\frac{di_{g\_a}}{dt} = \frac{1}{L_2} v_{x\_a} - \frac{1}{L_2}\left(R_2 i_{g\_a} + v_{g\_a}\right) \tag{16}$$

Assuming R_f is negligible Equation 17 can be written for the capacitor voltage:

$$\frac{dv_{x\_a}}{dt} = \frac{1}{C_f}\left(i_{i_a} - i_{g_a}\right) \tag{17}$$

Typically a synchronous reference frame control is designed, where a d_q rotating reference frame at grid frequency speed, and oriented such that the d axis is aligned to the grid voltage vector is used. Using basic trigonometric identities, i_d and i_q can be written as Equation 18 and Equation 19.

$$i_d = \frac{2}{3}\left(i_a \cos\omega t + i_b \cos\left(\omega t - 120\right) + i_c \cos(\omega t + 120)\right) \tag{18}$$

$$i_q = -\frac{2}{3}\left(i_a \sin\omega t + i_b \sin\left(\omega t - 120\right) + i_c \sin(\omega t + 120)\right) \tag{19}$$

Taking the derivative, and using the partial derivative theorem, Equation 20 is written:

$$\begin{aligned} &\xrightarrow{\text{yields}} \frac{di_d}{dt} = \frac{2}{3}\left(\frac{di_a}{dt}\cos\omega t + \frac{di_b}{dt}\cos\left(\omega t - 120\right) + \frac{di_c}{dt}\cos(\omega t + 120)\right) + \omega i_q \\ &\frac{di_d}{dt} = \frac{2}{3}\left(\frac{di_a}{dt}\cos\omega t + \frac{di_b}{dt}\cos\left(\omega t - 120\right) + \frac{di_c}{dt}\cos(\omega t + 120)\right) - \frac{2}{3}\omega\left(i_a \sin\omega t + i_b \sin\left(\omega t - 120\right) + i_c \sin(\omega t + 120)\right) \end{aligned} \tag{20}$$

The following state equations can be written:

$$\frac{di_{i\_d}}{dt} = \frac{1}{L_i} v_{i\_d} + \omega i_{i\_q} - \frac{1}{L_i}\left(R_i i_{i\_d} + v_{x\_d}\right) \tag{21}$$

$$\frac{di_{i\_q}}{dt} = \frac{1}{L_i} v_{i\_q} - \omega i_{i\_d} - \frac{1}{L_i}\left(R_i i_{i\_q} + v_{x\_q}\right) \tag{22}$$

Hence, using these equations, and substituting in Equation 23:

$$\left(sL_i\right) i_{i\_d}\left(s\right) = v_{i\_d}\left(s\right) + \left(\omega L_i\right) i_{i\_q}\left(s\right) - \left(R_i i_{i\_d}\left(s\right) + v_{x\_d}\left(s\right)\right) \tag{23}$$

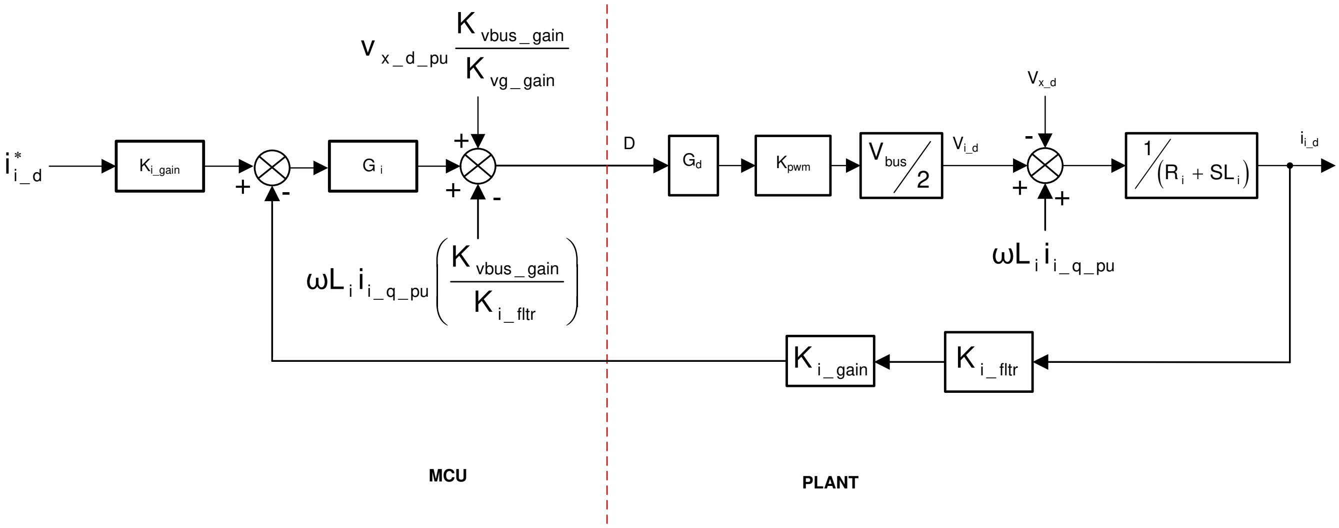
Taking the Laplace function on the previous equations:

$$\left(sL_i\right) i_{i\_q}\left(s\right) = v_{i_q}\left(s\right) - \left(\omega L_i\right) i_{i\_d}\left(s\right) - \left(R_i i_{i\_q}\left(s\right) + v_{x\_q}\left(s\right)\right) \tag{24}$$

When written in control diagram format, this looks like the following. Feedforward elements are added to remove additional sources of disturbances and errors in the model, two feedforward elements are added,

1. For the coupling term from the other axis in synchronous frame
2. For the output grid voltage

The diagram is drawn as shown in Figure 2-27.


$i^*_{i\_d}$
K_{i_gain}
+
-
G_i
$v_{x\_d\_pu}\frac{K_{vbus\_gain}}{K_{vg\_gain}}$
+
+
-
$\omega L_i i_{i\_q\_pu}\left(\frac{K_{vbus\_gain}}{K_{i\_fltr}}\right)$
D
G_d
K_{pwm}
$V_{bus}/2$
V_{i_d}
V_{x_d}
-
+
+
$\omega L_i i_{i\_q\_pu}$
$1/(R_i + SL_i)$
i_{i_d}
K_{i_gain}
K_{i_fltr}
MCU
PLANT

where:

- $i^*_{i\_d}$ is the current reference
- K_{i_gain} is the current sense scalar which is one over max current sense
- K_{i_fltr} is the filter that is connected on the current sense path. current sense scalar which is one over max current sense
- K_{vbus_gain} is the voltage sense scalar for the bus, which is one over max voltage sensed
- K_{vg_gain} is the voltage sense scalar for the grid voltage, which is one over max voltage sensed

Figure 2-27. Id Current Loop Model

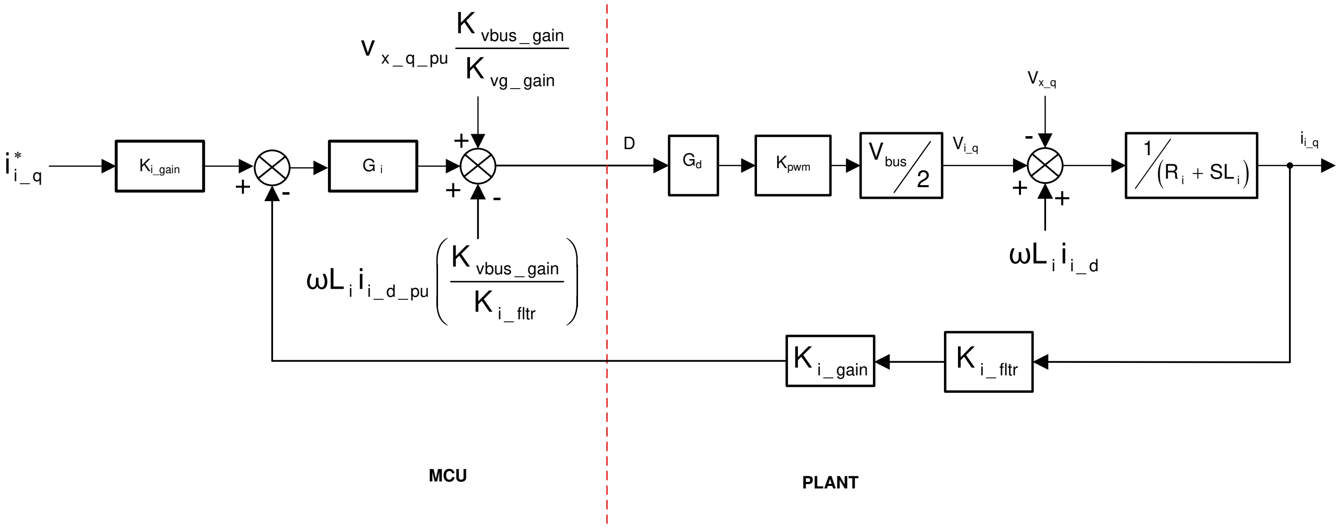


Figure 2-28. Iq Current Loop Model

With the feedforward elements, the small signal model can be written as Equation 25(Note: Separate scaling factors are applied to bus voltage and grid voltage due to the differences in the sensing range.):

$$\frac{\hat{i}_{i\_d\_pu}}{\hat{d}} = G_d K_{pwm} \frac{1}{K_{vbus\_gain}} K_{i\_gain} K_{i\_fltr} \frac{1}{(R_i + sL_i)} \tag{25}$$

In the case of an LCL filter, the following can be assumed as a simplified model as in Equation 26:

$$\frac{\hat{i}_{i\_d\_pu}}{\hat{d}} = G_d K_{pwm} \frac{1}{K_{vbus\_gain}} K_{i\_gain} \; K_{i\_fltr} \frac{1}{\left(Z_i + \frac{Z_c Z_f}{(Z_c + Z_f)} \right)} \tag{26}$$

The current loop plant is compared with the Software Frequency Response Alert (SFRA) measured data for the current loop as illustrated in Figure 2-29.

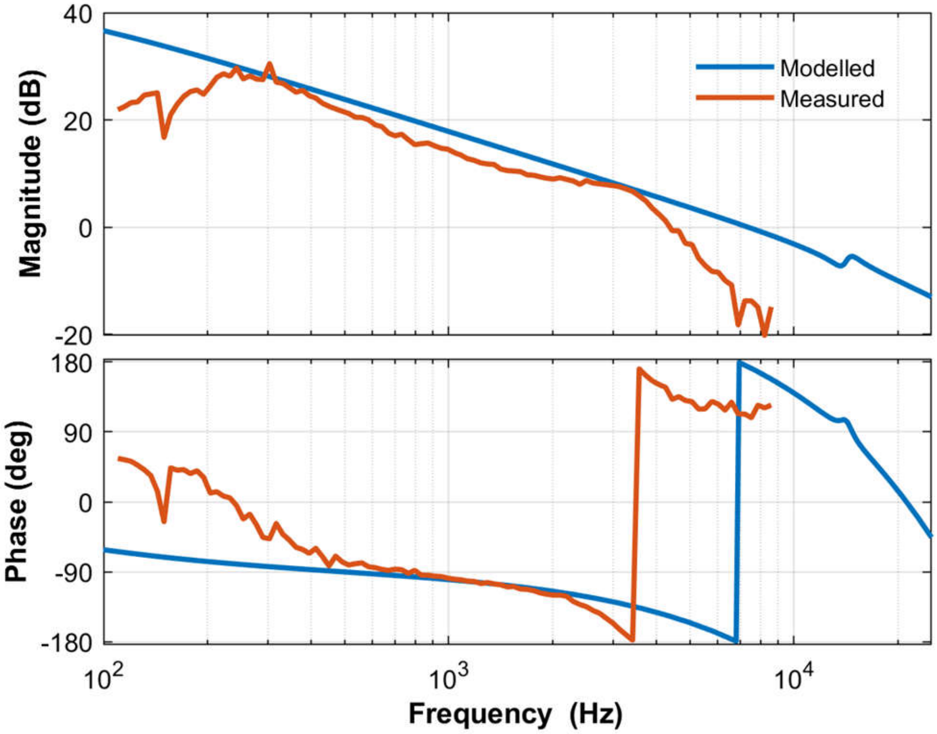


Figure 2-29. Current Loop Plant Frequency Response Modeled vs Measured

Equation 27 represents the compensator designed for the closed-loop operation:

$$G_i = 0.3 \times \frac{(s + 2\pi \times 95.6)}{s} \tag{27}$$

With which the open loop plot in Figure 2-30 is achieved, gives roughly > 1-kHz bandwidth in the I_d and I_q loop.

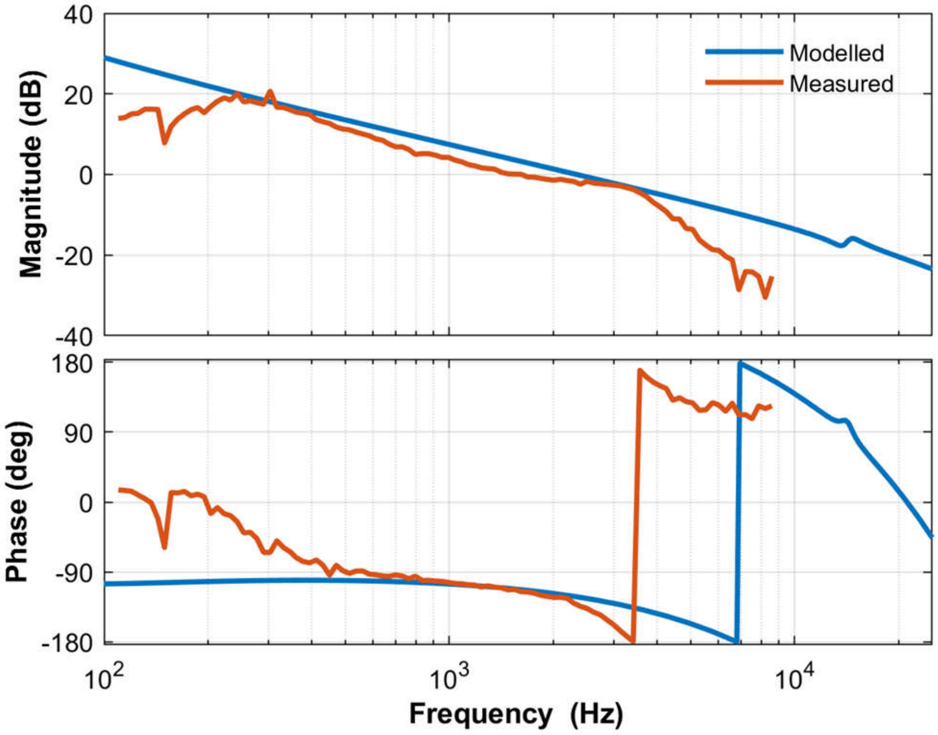


Figure 2-30. Current Loop, Open Loop Response Modeled vs Measured

2.3.6.2 PFC DC Bus Voltage Regulation Loop Design

Before looking at the voltage loop model, the power measurement from DQ domain can be written as:

$$P = v_a i_a + v_b i_b + v_c i_c = \begin{bmatrix} v_a & v_b & v_c \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \left(T^{-1}_{abc->dq0} \overrightarrow{v_{dq0}}\right)^T \left(T^{-1}_{dq0->abc} \overrightarrow{i_{dq0}}\right) = \overrightarrow{v_{dq0}} \left(T^{-1}_{abc->dq0}\right)^T T^{-1}_{abc->dq0} \overrightarrow{i_{dq0}} = \frac{3}{2}\left(v_{gd} i_d + v_{gq} i_q + v_{g0} i_0\right) \tag{28}$$

where

$$T_{abc->dq0} = \frac{2}{3} \begin{bmatrix} \cos(\omega t) & \cos(\omega t - \frac{2\pi}{3}) & \cos(\omega t + \frac{2\pi}{3}) \\ -\sin(\omega t) & -\sin(\omega t - \frac{2\pi}{3}) & -\sin(\omega t + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

Hence:

$$P = \frac{3}{2}\left(v_{gd} i_d + v_{gq} i_q\right) \tag{29}$$

$$Q = -v_{gd} i_d + v_{gq} i_q \tag{30}$$

The DC Bus regulation loop is assumed to be providing the power reference, which is divided by the square of the line voltages RMS to provide the conductance. When further multiplied by the line voltage gives the instantaneous current.

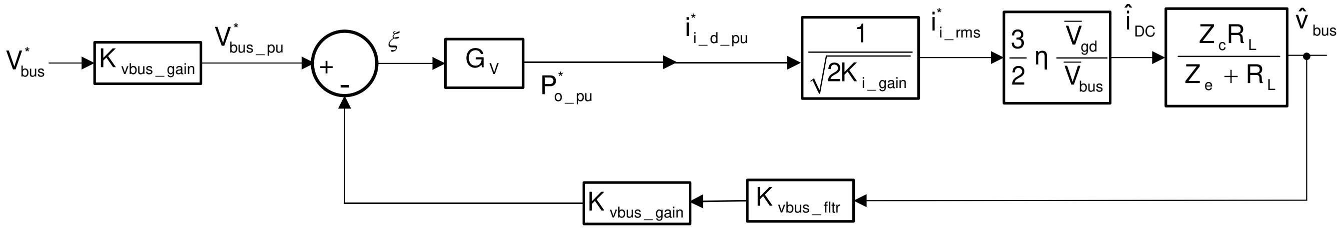


Figure 2-31. Voltage Loop Model

A small-signal model of the DC bus regulation loop is developed by linearizing Equation 31 around the operating point:

$$i_{DC} v_{bus} = 3\eta v_{g\_rms} i_{g\_rms} \Rightarrow \hat{i}_{DC} = 3\eta \frac{\bar{V}_{g\_rms}}{\bar{V}_{bus}} \hat{i}_{g\_rms} \tag{31}$$

Because transformation is an amplitude invariant, translating from RMS to peak quantities using $\hat{i}_{g\_rms} = \frac{1}{\sqrt{2}} \hat{i}_{gd}$ and $\hat{v}_{g\_rms} = \frac{1}{\sqrt{2}} \hat{v}_{gd}$, Equation 32 can be derived.

$$\hat{i}_{DC} = \frac{3}{2} \frac{\bar{V}_{gd}}{\bar{V}_{bus}} \hat{i}_{gd} \tag{32}$$

Also for resistive load on the DC Bus: $\hat{v}_{bus} = \frac{Z_c R_L}{Z_c + R_L} \hat{i}_{DC}$.

Therefore, the voltage loop plant can be written as Equation 33

$$G_{v\_bus} = \frac{Z_c R_L}{Z_c + R_L} \times p \times \frac{K_{vbus_{fltr}} \times K_{vbus_{gain}}}{K_{i_{gain}}} \times \frac{1}{\left(V_{bus} \times K_{vg_{gain}}\right)} \times \frac{3}{2} \times \left(\frac{V_{g\_peak}}{V_{g\_Sense\_max}}\right) \tag{33}$$

Using the previous model the following compensator, Equation 34 is designed for the voltage loop:

$$G_v = 1.8581 \times \frac{(s + 2\pi \times 35)}{s} \tag{34}$$

SFRA is used to measure the voltage loop bandwidth, and compare against the model which shows good correlation to the model. Figure 2-32 shows the plant frequency response comparison and Figure 2-33 shows the open-loop frequency response comparison of modeled versus measured.

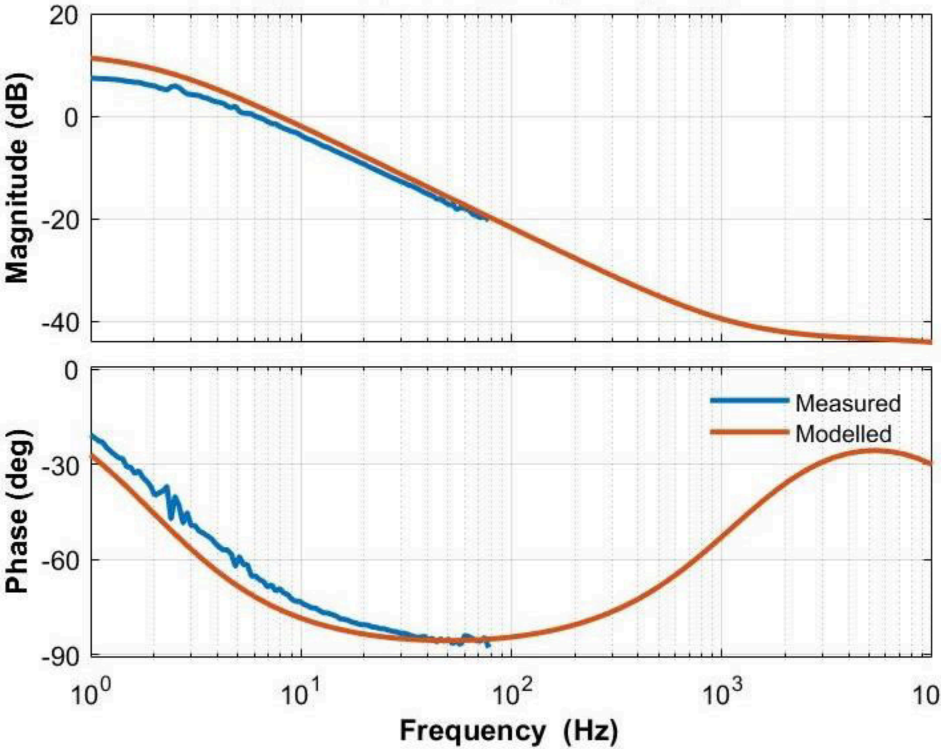


Figure 2-32. Voltage Loop Plant Frequency Response Measured vs Modeled

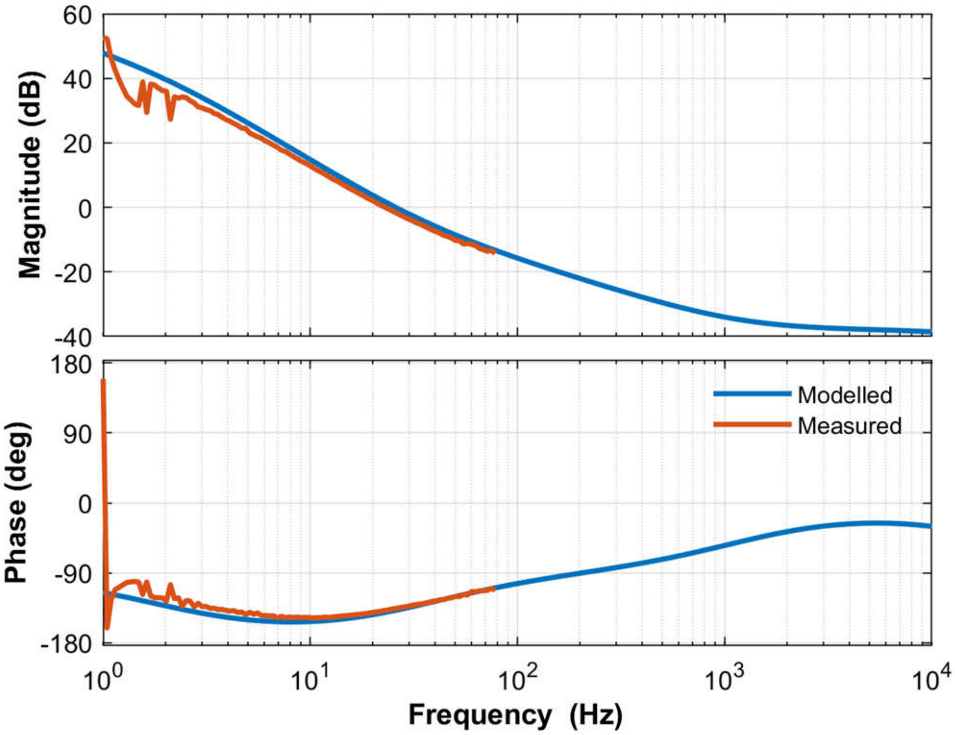


Figure 2-33. Voltage Loop, Open-Loop Frequency Response Modeled vs Measured

3 Hardware, Software, Testing Requirements, and Test Results

3.1 Required Hardware and Software

This section details the hardware and explains the different sections on the board and how to set them up for the experiments as outlined in this design guide.

3.1.1 Hardware

3.1.1.1 Test Hardware Required

The design under test (DUT) in this design is set up and is operated in several pieces:

- One E7 TIDA-01606 motherboard
- TMS320F28379D Control Card or TMS320F280039C Control Card
- Mini USB cable with a snap ferrite (for example, ZCAT3035-1330-BK, recommended for noise immunity)
- One USB isolator (for safety and noise immunity)
- Laptop or other computer (for example, Microsoft® Windows® 11 operating system)

The test equipment required to power and evaluate the design is as follows:

1. 12-V, 2-A bench style supply for primary board power. TI recommends a supply with overvoltage protection (OVP) and overcurrent protection (OCP) features.
2. For PFC mode:
   - 400 V_{L-L} capable three-phase AC source
   - 11-kW equivalent e-load to be connected at the DC output
3. For Inverter mode:
   - Star-connected resistive load network (for example, 11-kW load bank)
   - 800-V, 12-A power supply for DC link input
4. Four-channel, power quality analyzer (for example, WT5000)
5. Oscilloscope, voltage meter, differential probes (for example, P5200A) and current probes (for example, TCP0030)

The design follows a high-speed edge card (HSEC) concept. This design can be scaled across multiple devices from the C2000™ MCU product family with a compatible HSEC control. The key resources used for controlling the power stage on the MCU are listed in Table 3-1.

3.1.1.2 Microcontroller Resources Used on the Design (TMS320F28379D)

Table 3-1 details the key controller peripherals used for control of the power stage on the board and Table 3-2 lists the key connectors and functions.

Table 3-1. Key Controller Peripherals Used for Control of Power Stage on Board

PIN NUMBER	DESCRIPTION	SOFTWARE NAME
15, 31, 28	Grid Voltage Sense Phase A, B, C	TINV_VGRID_A, B, C
21, 33, 30	Inverter Side Voltage Phase A, B, C	TINV_VINV_A, B, C
25, 37, 34	Inverter Side Current Phase A	TINV_IINV_A, B, C
42	Bus Voltage Sensing	TINV_VBUS
40	Bus Voltage Midpoint Sensing	TINV_VBUS_MID
12, 14, 18, 20	Temperature A, B, C and Ambient	TINV_TEMP_A, B, C, AMB
49, 50, 58	PWM1 Phase A, B, C	TINV_Q1_A, B, C
51, 52, 60	PWM3 Phase A, B, C	TINV_Q3_A, B, C
53, 54, 62	PWM2 Phase A, B, C	TINV_Q2_A, B, C
99, 103, 107	SDFM Data IG A, B, C	TINV_IGRID_A, B, C
101, 105, 109	SDFM Clock IG A, B, C	
57, 75	SDFM Clock Source	
89, 87 , 85	SiC Fault Signal A, B, C (active Low)	TINV_FAULT_A, B, C
92	Control relays on A, B, C. E7 uses one GPIO to control all relays.	TINV_RELAY_N
61, 63	Gate driver supply PWM	TINV_GATE_DRIVE

Table 3-1. Key Controller Peripherals Used for Control of Power Stage on Board (continued)

PIN NUMBER	DESCRIPTION	SOFTWARE NAME
71	Control GPIO for FAN. Previously, in E6 the control GPIO for FAN was pin 59.	TINV_FAN
108, 110	These pins are used to see ISR nesting and so forth, on the docking station while starting firmware debug	TINV_PROFILING1, 2
95	Gate driver enable	TNV_PWM_EN
81	Gate driver reset	TINV_R

Note

There is a minor software change to support E7 hardware with the C2000Ware_DigitalPower_SDK. Make sure to change the code in `tinv_user_settings.h` to support the new fan control GPIO.

```
//#define TINV_FAN_GPIO                      9
//#define TINV_FAN_GPIO_PIN_CONFIG           GPIO_9_GPIO9

#define TINV_FAN_GPIO                      18
#define TINV_FAN_GPIO_PIN_CONFIG           GPIO_18_GPIO18
```

For the relay user control in the CCS watch window, use the `TINV_neutralRelaySet` function to control the relays (E7 hardware change). Additionally, the isolated bias supply enable pin is active low. When the user starts the CCS debugger, the supply is enabled by default. Do not use the `TINV_allRelaySet` function as this affects the bias supply enable pin. Make sure to enable the fans with the `TINV_fanSet` function before pushing high power as the FETs can get too hot.

Table 3-2. Key Connectors and Functions

CONNECTOR NAME	FUNCTION
J13, J15, J18	VDC+, VDC_MID, and VDC– terminals
J30, J14, J16, J17	PE (Protected Earth), L1, L2, and L3 terminals
J3	12-V auxiliary power supply
J1	Jumper for auxiliary power supply
J26, J29B	HSEC control card connector slot
J4, J5, J6	MCU GND reference options. Select one option at a time. Default is J6 (GRID_GND / PE)
J19, J20, J21	Fan connectors
S1–S5	Set all five switches to either position 1 for 379D or position 3 for 039C operation. The PCB has silkscreen labels to help the user configure the settings.

Note

The default MCU GND reference is to GRID_GND / PE. Using a ferrite in place of R10 helps with common noise rejection. If the user decides to select J4 or J5 as the MCU reference, make sure to use an isolated 12-V bench supply when testing with the high-voltage input source (for example, 3-P Chroma). Sometimes the switching noise can disrupt the fan function through conductive and radiated emissions. If snap ferrite chokes cannot mitigate the fan noise issue, use a separate 12-V power supply to keep the fan working properly during high-power tests. That is, disconnect the fans from J19, J20, and J21 and connect them to a separate 12-V (1 A) supply.

During high-power tests, TI recommends using differential probes to measure signals even on the MCU low-voltage side as most single-ended probe return clips are tied to earth GND. This provides a low-impedance path for the common-mode noise and is visible on the scope. Using a differential probe provides adequate impedance to earth GND thus decouples common-mode noise from the high side to the low side. This works well when a ferrite is used in place of R10.

3.1.1.3 F28377D, F28379D Control-Card Settings

Certain settings on the device control card are needed to communicate over JTAG and use the isolated UART port. The settings also provide a correct ADC reference voltage. The following are the settings required on revision 1.1 of the F28379D control card.

See also the information sheet located inside C2000Ware at `<sdk_install_path>\c2000ware\boards\controlCARDs\TMDSCNCD28379D`:

1. A:SW1 on the control card must be set on both ends to ON (up) position to enable the JTAG connection to the device and the UART connection for SFRA GUI. If this switch is OFF (down), the built-in, isolated JTAG on the control card cannot be used and the SFRA GUI cannot communicate to the device.
2. A:J1 is the connector for the USB cable that is used to communicate to the device from a host PC on which the Code Composer Studio (CCS) runs. Use a USB isolator to connect to the laptop, as previously discussed.
3. A 3.3-V reference is desired for the control loop tuning on this design; therefore, set the appropriate jumpers to provide a 3.3-V reference externally to the on-chip ADC. For version 1.3 of the F28379D control card, this means SW3 and SW2 are moved to the end with "." that is, to the left as defined, which puts 3.3-V VDDA as the reference for the ADC. See the information sheet for more information.
4. On jumper connector J3, connect together pin 2 and pin 3 by using a jumper.

3.1.1.4 Microcontroller Resources Used on the Design (TMS320F280039C)

Note

The TMDSCNCD280039C evaluation and development board is supported in the future. Configure S1–S5 on the main motherboard according to the control card of interest.

3.1.2 Software

Find related software information at the following link: www.ti.com/tool/C2000WARE-DIGITALPOWER-SDK

3.1.2.1 Getting Started With Firmware

3.1.2.1.1 Opening the CCS project

The software of this design is available inside C2000Ware_DigitalPower_SDK and is supported inside the powerSUITE framework. To open the project:

1. Install CCS (version 10.1 or above)
2. Install C2000Ware DigitalPower SDK from the tool page
3. Open CCS, and create a new workspace
4. Inside CCS, go to *View → Resource Explorer*. Under *Resource Explorer*, go to *Software → C2000Ware DigitalPower SDK - <version>* → solutions and select this solution; that is, **TIDA-01606**, and click import project. The code is available for both F28379D and F280039C.

Note

CCS can recommend installing a particular version of the compiler relevant to the imported project. If requested, find the compiler on ti.com to download and install. Configure the compiler version in the project properties menu after installing the compiler. Make sure the CCS project tool discovery path includes the path of your compiler installation.

3.1.2.1.2 Digital Power SDK Software Architecture

The general structure of the project is shown in <>. Once the project is imported, the Project Explorer appears inside CCS.

Solution-specific and device-independent files that consist of the core algorithmic code are in `<solution>.c` and `<solution>.h`.

Board-specific and device-specific files are in `<solution>_hal.c` and `<solution>_hal.h`. This file consists of device-specific drivers to run the scenario. If the user wants to use a different modulation scheme or a different device, the user is required only to make changes to these files, besides changing the device support files in the project.

The `<solution>-main.c` file consists of the main framework of the project. This file consists of calls to the board and the `solution` file that help in creating the system framework, along with the interrupt service routines (ISRs) and slow background tasks.

For this design, `<solution>` is *tinv* which is also referred to as the module name.

The powerSUITE page can be opened by clicking on the `main.syscfg` file, listed under the Project Explorer. The powerSUITE page generates the `<solution>_settings.h` file. This file is the only C based file used in the compile of the project that is generated by the powerSUITE page. The user must not modify this file manually, as the changes are overwritten by powerSUITE each time the project is saved. `<solution>_user_settings.h` is included by the `<solution>_settings.h` and can be used to keep any settings that are outside the scope of powerSUITE tools such as `#defines` for ADC mapping, GPIOs, and so forth.

The `Kit.json` and `solution.js` files are used internally by powerSUITE and also must not be modified by the user. Any changes to these files results in the project not functioning properly.

The design name is also used as the module name for all the variables and defines used in the design.

Therefore, all variables and function calls are prepended by the *TINV* name (for example, `TINV_vSecSensed_pu`). This naming convention lets the user combine different scenarios while avoiding naming conflicts.

3.1.2.1.3 Interrupts and Lab Structure

The project consists of two ISRs (ISR1 and ISR2) with ISR1 being the fastest and non-nestable ISR. ISR1 is reserved for the control loop and the PWM update. ISR1 is triggered by the PRIM_LEG1_PWM_BASE → EPWM_INT_TBCTR_U_CMPC event.

ISR2 is triggered by CPU Timer INT which is initiated by an overflow on CPU timer. It is used to run housekeeping functions such as doing a running average on the currents and voltage signals to remove noise and running the slew rate function for commanded references.

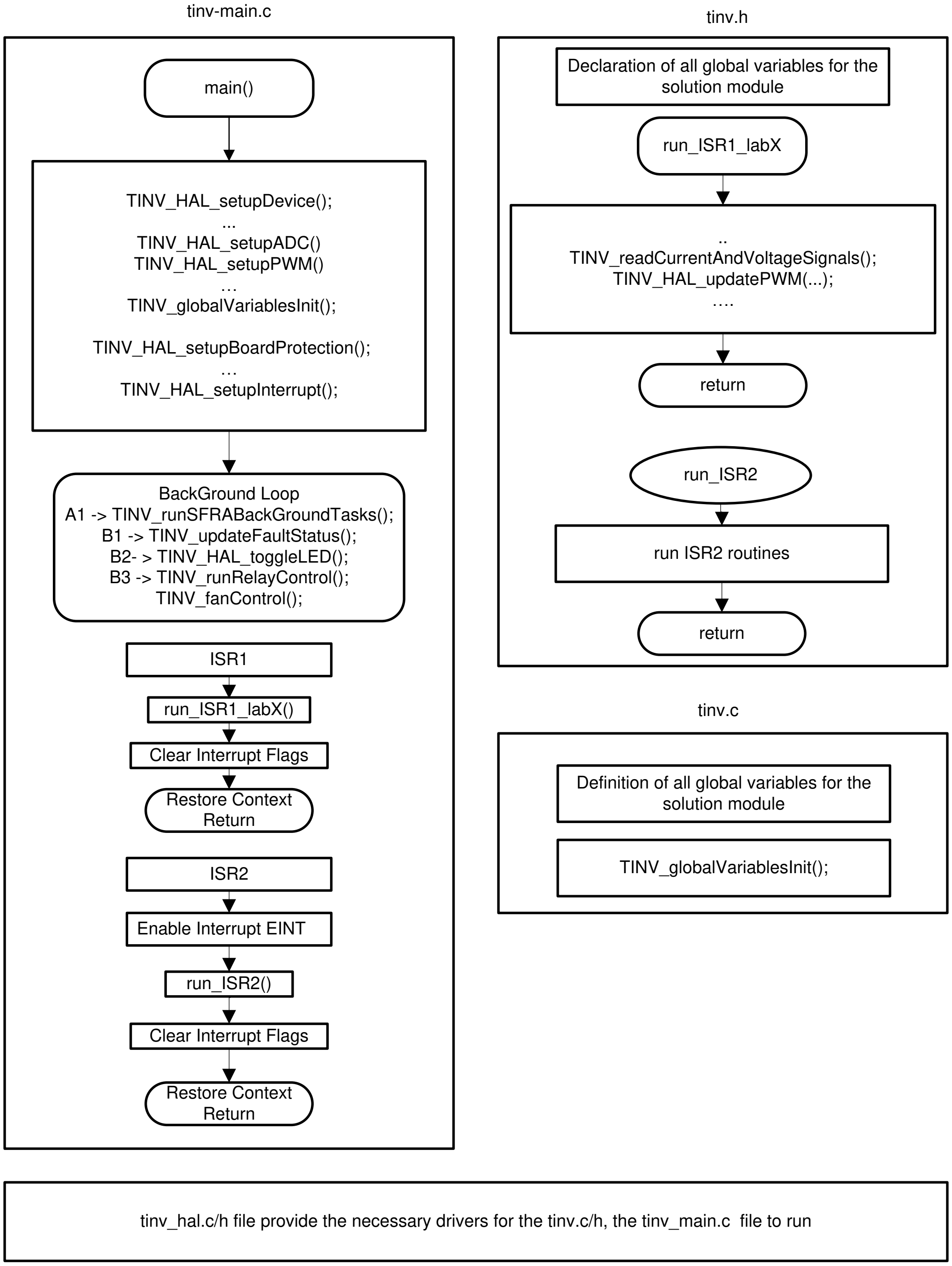

tinv-main.c
main()
TINV_HAL_setupDevice();
...
TINV_HAL_setupADC()
TINV_HAL_setupPWM()
…
TINV_globalVariablesInit();
TINV_HAL_setupBoardProtection();
…
TINV_HAL_setupInterrupt();
BackGround Loop
A1 -> TINV_runSFRABackGroundTasks();
B1 -> TINV_updateFaultStatus();
B2- > TINV_HAL_toggleLED();
B3 -> TINV_runRelayControl();
TINV_fanControl();
ISR1
run_ISR1_labX()
Clear Interrupt Flags
Restore Context Return
ISR2
Enable Interrupt EINT
run_ISR2()
Clear Interrupt Flags
Restore Context Return
tinv.h
Declaration of all global variables for the solution module
run_ISR1_labX
..
TINV_readCurrentAndVoltageSignals();
TINV_HAL_updatePWM(...);
….
return
run_ISR2
run ISR2 routines
return
tinv.c
Definition of all global variables for the solution module
TINV_globalVariablesInit();
tinv_hal.c/h file provide the necessary drivers for the tinv.c/h, the tinv_main.c file to run

Figure 3-1. Software Flow Diagram

The software of this reference design is organized in seven labs, Table 3-3 lists the labs and how they have been tested. All the labs can be run on the C28x Main CPU or the *Control Law Accelerator*.

Table 3-3. Overview of Labs to Test Reference Design

LAB NUMBER	DESCRIPTION	COMMENTS	TEST ENVIRONMENT
1	INV: PWM and ADC check	Test the PWM driver, ISR structure and execution rate, can be run on a control card. Unit test protection mechanisms. Test ADC mapping and reading of conversion data.	Control Card
2	INV: Open loop check	PWM Check, ADC check, Protection Check, inverter mode DC bus connected and resistive star network as load	Control Card + Power Stage Hardware
3	INV: Closed Current Loop, Resistive load connected at AC		Control Card + Power Stage Hardware
4	INV: Closed Current Loop, Grid connected test inverter mode		Control Card + Emulated power stage under Hardware In-the Loop
5	PFC: Three phase AC source , Resistive load at DC, open Loop check	1. Check if the vGridRms, iGridRms, and vBus measurements are correct 2. Check if PLL is locked.	Control Card + Power Stage Hardware
6	PFC: Closed Current Loop, Resistive load connected at DC, three phase AC ?		Control Card + Power Stage Hardware
7	PFC: Closed Voltage loop + Current Loop , Resistive load connected at DC, three phase AC ?		Control Card + Power Stage Hardware

3.1.2.1.4 Building, Loading, and Debugging the Firmware

To build the project, right-click on the project name and click *Rebuild Project*. The project builds successfully.

To load the project, first make sure in the Project Explorer the correct target configuration file is set as Active under targetConfigs (*.ccxml file). Then, click *Run* → *Debug* to launch a debugging session. In case of dual-CPU devices, a window can appear for the user to select the CPU on which the debug is to be performed. In this case, select CPU1. The project then loads on the device and the CCS debug view becomes active. The code halts at the start of the main routine.

To debug the system, monitor the variables in the watch/expressions window. To populate this window with the correct variables, click *View* → *Scripting Console* to open the scripting console dialog box. On the upper right corner of this console, click on *Open* and then browse to the `setupdebugenv_lab<Number>.js` script file located inside the project folder. This populates the watch window with the appropriate variables needed to debug the system. Enable Continuous Refresh button on the watch window to enable continuous update of values from the controller.

Real-time emulation is a special emulation feature that allows windows within Code Composer Studio to be updated while the MCU is running. This allows graphs and watch views to updated, but also allows the user to change values in watch or memory windows, and see the effect of these changes in the system without halting the processor. To enable real-time mode click on this button on the top bar of CCS. If a message box appears, select *YES* to enable debug events. This sets bit 1 (DGBM bit) of status register 1 (ST1) to a *0*. DGBM is the debug enable mask bit. When the DGBM bit is set to *0*, memory and register values can be passed to the host processor for updating the debugger windows.

In different labs, sometimes the currents and voltages measured or the control variables need to be verified by viewing the data in the graph window. For this Graph window can be used which in conjunction with a piece of code that runs on the controller can show a snapshot of how the values are being sensed by the controller. The values are logged by the datalogger typically in the slower ISRs. To import the graph into the CCS view select Tools → Graph → DualTime, and click Import and point to the `graph1.GraphProp` file inside the project folder.

Two graphs appear in CCS. Click Continuous Refresh on these graphs. A second set of graphs can also be added by importing the graph2.GraphProp file.

3.1.2.1.5 CPU Loading

The main control ISR with Lab 3 and Lab 7 takes approximately 54 MIPS at 50-kHz rate when running from CPU1, that is approximately 27% of the CPU when running from 200-MHz F2837x processor. This includes the ADC drivers, abc-dq0 and dq0-abc transformation, transformation, PWM generation , two current control loops, one voltage loop, and the SFRA call.

3.1.2.2 Protection Scheme

Figure 3-2 explains the software functions used to setup the trip behavior on the design.

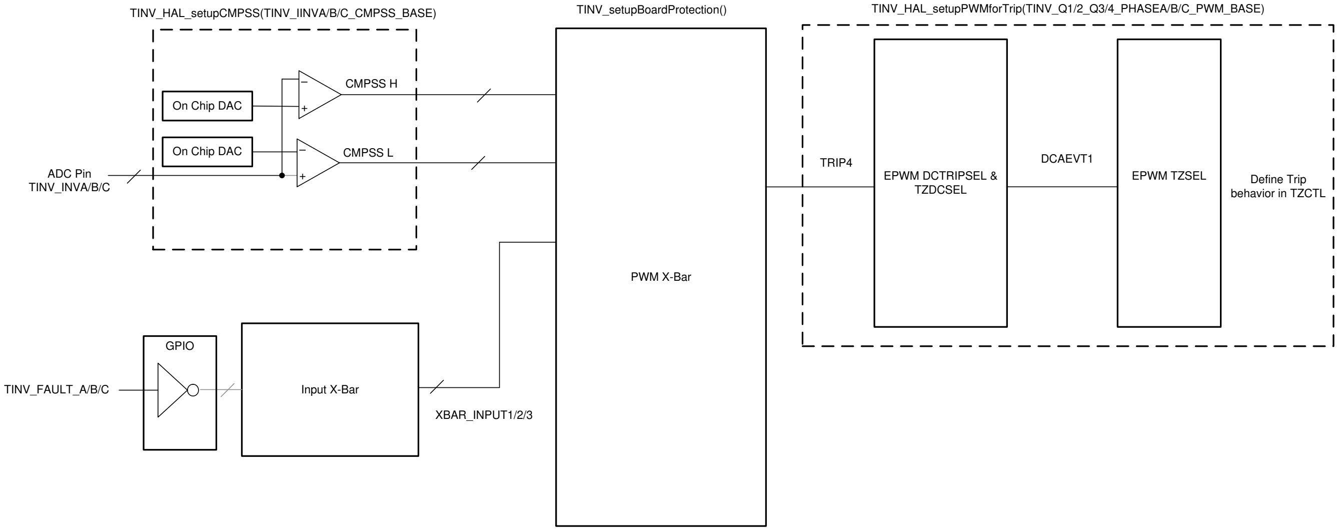

TINV_HAL_setupCMPSS(TINV_IINVA/B/C_CMPSS_BASE)
TINV_setupBoardProtection()
TINV_HAL_setupPWMforTrip(TINV_Q1/2_Q3/4_PHASEA/B/C_PWM_BASE)
On Chip DAC
On Chip DAC
CMPSS H
CMPSS L
ADC Pin
TINV_INVA/B/C
TRIP4
EPWM DCTRIPSEL & TZDCSEL
DCAEVT1
EPWM TZSEL
Define Trip behavior in TZCTL
PWM X-Bar
GPIO
TINV_FAULT_A/B/C
Input X-Bar
XBAR_INPUT1/2/3

Figure 3-2. Software Diagram for Trip Setup

`TINV_updateFaultStaus()` function is called periodically in a slow background task to updating Trip Flags and also resetting the latch if needed.

If a trip event has occurred, the PWM needs Trip flags need to be cleared separately. This part is typically handled in the ISR by calling `TINV_clearPWMTrips()`.

3.1.2.3 PWM Switching Scheme

Figure 3-3 is the PWM configuration used, only phase-A PWM modules time base is shown. Others are identical. EPWM11 is used for the SDFM sync on F28377.

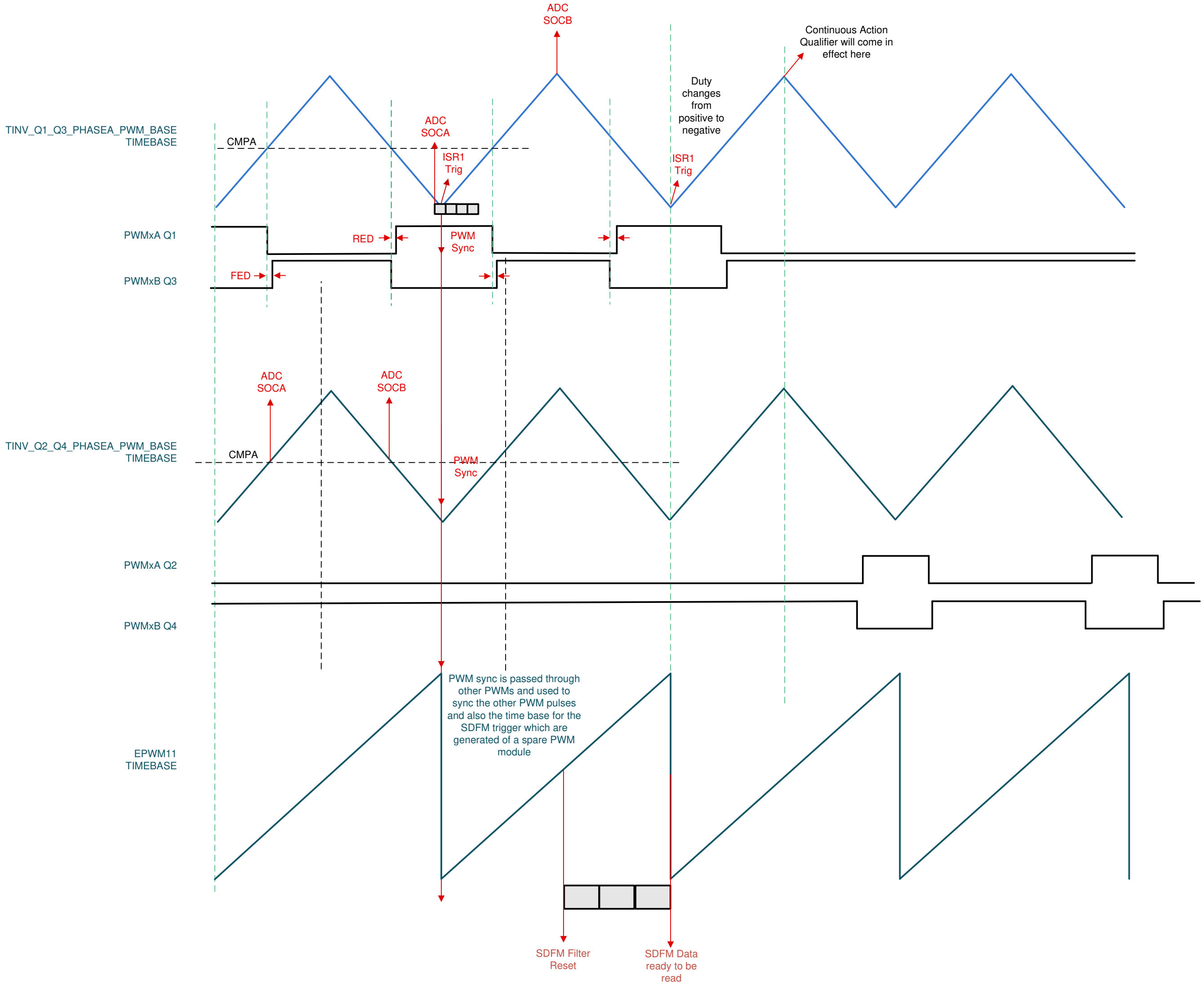

ADC SOCB
Continuous Action Qualifier will come in effect here
Duty changes from positive to negative
TINV_Q1_Q3_PHASEA_PWM_BASE TIMEBASE
CMPA
ADC SOCA
ISR1 Trig
ISR1 Trig
PWMxA Q1
RED
PWM Sync
PWMxB Q3
FED
ADC SOCA
ADC SOCB
TINV_Q2_Q4_PHASEA_PWM_BASE TIMEBASE
CMPA
PWM Sync
PWMxA Q2
PWMxB Q4
EPWM11 TIMEBASE
PWM sync is passed through other PWMs and used to sync the other PWM pulses and also the time base for the SDFM trigger which are generated of a spare PWM module
SDFM Filter Reset
SDFM Data ready to be read

Figure 3-3. PWM Configuration

3.1.2.4 ADC Loading

To maintain synchronous operation all conversions are triggered as following:

TINV_Q1_Q3_A_PWM_BASE; that is, EPWM1 TBCTR_D_CMPB → EPWM1_SOCA (green), triggered every cycle,

TINV_Q1_Q3_A_PWM_BASE; that is, EPWM1 TBCTR_D_CMPB → EPWM1_SOCB (), triggered every 10th cycle,

TINV_Q2_Q4_A_PWM_BASE; that is, EPWM2 TBCTR_U_CMPB → EPWM2_SOCA, triggered every cycle

TINV_Q2_Q4_A_PWM_BASE; that is, EPWM2 TBCTR_D_CMPB → EPWM2_SOCB, triggered every cycle

TINV_Q1_Q3_A_PWM_BASE; that is, EPWM3 TBCTR_PERIOD → EPWM3_SOCA, triggered every cycle

Table 3-4 shows the mapping with F2837xD on the TIDA-01606 hardware.

Table 3-4. ADC Loading Architecture F28379D

	ADC-A	ADC-B	ADC-C	ADC-D
SOC0	IINV-A → ADCIN-14, CMPSS4	TEMP_A → ADC-B0	IINV-B → ADC-C4, CMPSS5	IINV-C → ADC-D2, CMPSS8
SOC1	VGRID-A → ADC-A2,	TEMP_B → ADC-B1	VGRID-B → ADC-C2	VGRID-C → ADC-D0
SOC2	VINV-A → ADC-A4	TEMP_A → ADC-B2	VINV-B → ADC-C3	VINV-C → ADC-D1
SOC3	VGRID-A → ADC-A2,	TEMP_AMB → ADC-B3	VGRID-B → ADC-C2	VBUS → ADC-D5
SOC4	VGRID-A → ADC-A2,		VGRID-B → ADC-C2	VGRID-C → ADC-D0
SOC5	VGRID-A → ADC-A2,		VGRID-B → ADC-C2	VBUS → ADC-D5
SOC6				VGRID-C → ADC-D0
SOC7				VBUS → ADC-D5
SOC8				VGRID-C → ADC-D0
SOC9				VBUS → ADC-D5

Table 3-5 shows the mapping with F280039C on the TIDA-01606 hardware.

Table 3-5. ADC Loading Architecture F280039C

	ADC-A	ADC-B	ADC-C
SOC0	IINV-B→ ADCIN-A12, CMPSS2	IINV-C → ADCIN-B14, CMPSS3	IINV-A →ADCIN-C0, CMPSS1
SOC1	VGRID-A → ADC-A2,	V_REF → ADC-B8	VGRID-C → ADC-C1
SOC2	VINV-A → ADC-A5	VGRID-B→ ADC-B0	VINV-B → ADC-C3
SOC3	VINV-C → ADC-A8	VGRID-B → ADC-B0	VBUS → ADC-C14
SOC4	VMID → ADC-A3	VGRID-B → ADC-B0	VGRID-C → ADC-C1
SOC5	VGRID-A → ADC-A2,	VGRID-B → ADC-B0	VBUS → ADC-C14
SOC6	VMID → ADC-A3,	TEMP_A → ADC-B3	VGRID-C → ADC-C1
SOC7	VGRID-A → ADC-A2	TEMP_B→ ADC-B2	VBUS → ADC-C14
SOC8	VMID → ADC-A3,	TEMP_C → ADC-B12	VGRID-C → ADC-C1
SOC9	VGRID-A → ADC-A2	TEMP_AMB → ADC-B4	VBUS → ADC-C14
SOC10	VMID → ADC-A3		

Note

The ADC current reading is not used for the closed loop operation due to layout noise, instead SDFM bases sensing is employed to close the loop. Hence the grid current is used to close the current loop and the diagrams shall be interpreted accordingly for this change.

3.2 Testing and Results

All the labs in this section can be run with both control cards. However, when using the TMS320F280039C control card, consider the following notes:

- The GUI interface cannot be used. To change the parameters and the operating lab, parameters in the `user_settings.h` file must be changed directly.
- SFRA cannot be used.
- Space Vector modulation together with control of the middle point of the DC link capacitors can be used. Notice that the *neutral* of the grid must be disconnected from the grid. By changing the following user settings parameters, the two new control techniques can be operated:
  - `#define TINV_THIRD_HARMONIC_INJECTION_STATUS TINV_THIRD_HARMONIC_INJECTION_ENABLE`

- #define TINV_MIDDLE_POINT_CONTROL_STATUS TINV_MIDDLE_POINT_CONTROL_ENABLE

The following parameters were used for testing the E7 hardware and can be configured either with `main.syscfg` or `tinv_settings.h`. When using E6, keep the `default main.syscfg` values. The E7 parameters are shown in the follow code:

```
// Power Stage Settings
//
#define TINV_PWM_SWITCHING_FREQ_HZ ((float32_t)90*1000)
#define TINV_PWM_DEADBAND_US ((float32_t)0.15)
#define TINV_PWM_PERIOD_TICKS (TINV_PWMSYSCLOCK_FREQ_HZ / TINV_PWM_SWITCHING_FREQ_HZ)
#define TINV_PWM_DEADBAND_TICKS (int16_t)((float32_t)TINV_PWM_DEADBAND_US *    \
                                          (float32_t)TINV_PWMSYSCLOCK_FREQ_HZ *   \
                                          (float32_t)ONE_MICRO_SEC)

#define TINV_AC_FREQ_HZ ((float32_t)60)
#define TINV_VBUS_NOMINAL_VOLTS ((float32_t)800)
#define TINV_LI_INDUCTOR_VALUE  ((float32_t)0.13*0.001)
#define TINV_LG_INDUCTOR_VALUE  ((float32_t)0.01*0.001)
#define TINV_VGRID_MAX_SENSE_VOLTS ((float32_t)512.5)
#define TINV_VINV_MAX_SENSE_VOLTS TINV_VGRID_MAX_SENSE_VOLTS
#define TINV_VBUS_MAX_SENSE_VOLTS ((float32_t)1100)
#define TINV_IINV_MAX_SENSE_AMPS ((float32_t)33)
#define TINV_IINV_TRIP_LIMIT_AMPS ((float32_t)29)
#define TINV_IGRID_MAX_SENSE_AMPS ((float32_t)32)
#define TINV_IGRID_TRIP_LIMIT_AMPS ((float32_t)29)


//
// PI Controller Settings from Compensation Designer
//
#define TINV_GI_PI_KP ((float32_t)0.0996509341)
#define TINV_GI_PI_KI ((float32_t)0.0070057718)



#define TINV_GV_PI_KP ((float32_t) 1.9979056049)
#define TINV_GV_PI_KI ((float32_t) 0.0041887902)
```

Note

Some C2000Ware_DigitalPower_SDK software changes are required to support the new E7 hardware. The first change is the fan GPIO change as previously described in Section 3.1.1.2 (shown here for convenience). The second software change is to invert the voltage sensing polarity in the default SDK software because the latest E7 uses non-inverting voltage sensing for the V_PCC and V_SN.

For the fan GPIO change in `tinv_user_settings.h`:

```
//E6
//#define TINV_FAN_GPIO                         9
//#define TINV_FAN_GPIO_PIN_CONFIG              GPIO_9_GPIO9

//E7
#define TINV_FAN_GPIO                         18
#define TINV_FAN_GPIO_PIN_CONFIG              GPIO_18_GPIO18
```

Note

Make sure to enable the fans when testing at high power using the `TINV_fanSet` function in the CCS watch window during the debug session.

Use the following for the E7 voltage sensing polarity change in `tinv.h`:

```
    // Voltage sensing on the actual board is non-inverted hence a +2.0f needs to be multiplied as
below
    TINV_vInv_A_sensed_pu = ((float32_t)TINV_VINV_A_READ *
                      TINV_ADC_PU_SCALE_FACTOR -
                      TINV_vInv_A_sensedOffset_pu) *  2.0f;

    TINV_vInv_B_sensed_pu = ((float32_t)TINV_VINV_B_READ *
                      TINV_ADC_PU_SCALE_FACTOR -
                      TINV_vInv_B_sensedOffset_pu) * 2.0f;

    TINV_vInv_C_sensed_pu = ((float32_t)TINV_VINV_C_READ *
                      TINV_ADC_PU_SCALE_FACTOR -
                      TINV_vInv_C_sensedOffset_pu) * 2.0f;

    TINV_vGrid_A_sensed_prev_pu = TINV_vGrid_A_sensed_pu;

    TINV_vGrid_A_sensed_pu = ((float32_t)TINV_VGRID_A_READ *
                       TINV_ADC_PU_SCALE_FACTOR -
                       TINV_vGrid_A_sensedOffset_pu ) * 2.0f;

    TINV_vGrid_B_sensed_pu = ((float32_t)TINV_VGRID_B_READ *
                       TINV_ADC_PU_SCALE_FACTOR -
                       TINV_vGrid_B_sensedOffset_pu ) * 2.0f;

    TINV_vGrid_C_sensed_pu = ((float32_t)TINV_VGRID_C_READ *
                       TINV_ADC_PU_SCALE_FACTOR -
                       TINV_vGrid_C_sensedOffset_pu ) * 2.0f;
```

For the E6 voltage sensing, from `tinv.h` (the default SDK code is acceptable):

```
    // Voltage sensing on the actual board is inverted hence a -2.0f needs to be multiplied as below
    TINV_vInv_A_sensed_pu = ((float32_t)TINV_VINV_A_READ *
                      TINV_ADC_PU_SCALE_FACTOR -
                      TINV_vInv_A_sensedOffset_pu) *  -2.0f;

    TINV_vInv_B_sensed_pu = ((float32_t)TINV_VINV_B_READ *
                      TINV_ADC_PU_SCALE_FACTOR -
                      TINV_vInv_B_sensedOffset_pu) * -2.0f;

    TINV_vInv_C_sensed_pu = ((float32_t)TINV_VINV_C_READ *
                      TINV_ADC_PU_SCALE_FACTOR -
                      TINV_vInv_C_sensedOffset_pu) * -2.0f;

    TINV_vGrid_A_sensed_prev_pu = TINV_vGrid_A_sensed_pu;

    TINV_vGrid_A_sensed_pu = ((float32_t)TINV_VGRID_A_READ *
                       TINV_ADC_PU_SCALE_FACTOR -
                       TINV_vGrid_A_sensedOffset_pu ) * -2.0f;

    TINV_vGrid_B_sensed_pu = ((float32_t)TINV_VGRID_B_READ *
                       TINV_ADC_PU_SCALE_FACTOR -
                       TINV_vGrid_B_sensedOffset_pu ) * -2.0f;

    TINV_vGrid_C_sensed_pu = ((float32_t)TINV_VGRID_C_READ *
                       TINV_ADC_PU_SCALE_FACTOR -
                       TINV_vGrid_C_sensedOffset_pu ) * -2.0f;
```

Note

For E7 hardware, to turn on all three phase relays (after precharge) use the `TINV_neutralRelaySet` function in the CCS watch window. Only use `TINV_allRelaySet` for E6 hardware.

3.2.1 Lab 1

This lab is meant to be run on the control card and the docking station.

Set the project to Lab 1 by changing the Lab Number in the <tinv_settings.h> or main.syscfg file, (this is changed with the powerSUITE GUI when using the powerSUITE project). Most users can skip to Lab 2 to begin test flow.

```
#define TINV_LAB 1
```

All the other options can be left at default, for now, in the user_settings.h file

```
#if TINV_LAB == 1
#define TINV_TEST_SETUP TINV_TEST_SETUP_RES_LOAD
#define TINV_PROTECTION TINV_PROTECTION_DISABLED
#define TINV_SFRA_TYPE TINV_SFRA_CURRENT
#define TINV_SFRA_AMPLITUDE (float32_t)TINV_SFRA_INJECTION_AMPLITUDE_LEVEL2
#define TINV_POWERFLOW_MODE TINV_INVERTER_MODE
#define TINV_DC_CHECK 0
#define TINV_SPLL_TYPE TINV_SPLL_SRF
#endif
```

Figure 3-4 shows the software diagram when the code is running.

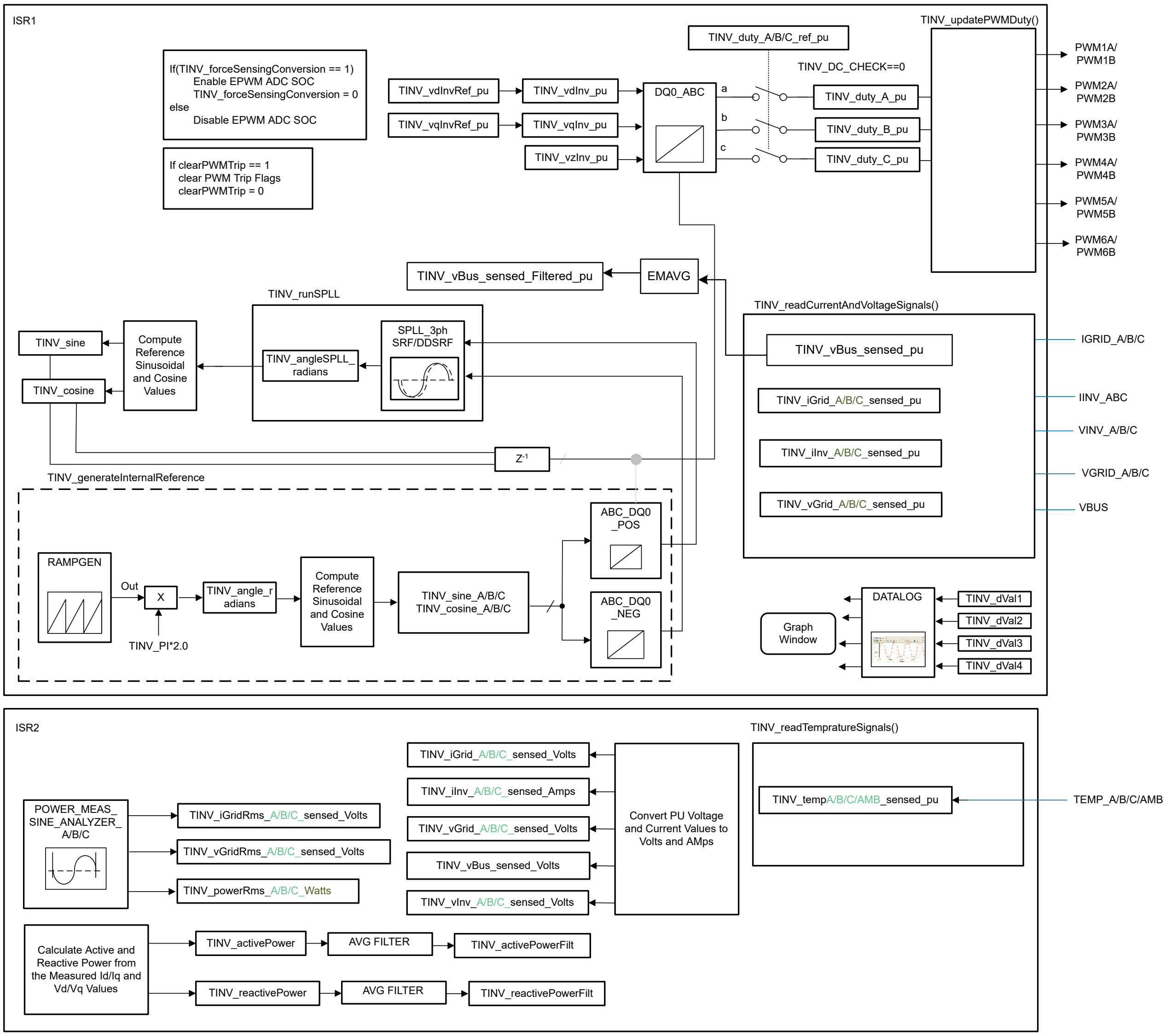


Figure 3-4. Lab 1 Software Diagram

3.2.2 Testing Inverter Operation

Lab 2, Lab 3, and Lab 4 elaborate the steps for running the power stage in the inverter mode. Lab 2 is the inverter mode of operation in open loop. Lab 3 is the inverter mode of operation with closed current loop. Lab 4 is the grid connected inverter mode of operation and this is checked only under hardware-in-the-loop (HIL) platform and not on the hardware. The high voltage (800 VDC) is applied across terminals J13 and J18. 12-V auxiliary power supply is connected to terminal J3. Three-phase star connected resistive load is connected across terminals J14, J16, and J17. J30 is the protective earth terminal which is connected to the high-voltage power source earth.

A check for DC bus overvoltage is added to all *Inverter Labs*, Lab 1 through Lab 5, using a filtered value for the DC bus voltage. The `TINV_filterAndCheckForBusOverVoltage()` function runs from ISR1 and checks for DC bus overvoltage condition. Under overvoltage condition, this function shuts off all PWM outputs and registers the system operating state as *bus overvoltage state*. Filtered DC bus voltage is calculated from instantaneous sensed DC bus voltage using the averaging function `EMAVG`. This is all calculated inside ISR1. The user can set the `TINV_VBUS_OVERVOLT_LIMIT` in `tinv_user_settings.h`:

```
#define TINV_UNDERVOLT_LIMIT
#define TINV_VBUS_OVERVOLT_LIMIT 900
#define TINV_VBUS_CLAMP_MIN_PU 0.1f
#define TINV_GRID_OVER_UNDER_FREQ_LIMIT 3
#define TINV_GRID_OVER_UNDER_VRMS_LIMIT 35
#define TINV_UNIVERSAL_GRID_MAX_VRMS    240
#define TINV_UNIVERSAL_GRID_MIN_VRMS    20
#define TINV_UNIVERSAL_GRID_MAX_FREQ    65
#define TINV_UNIVERSAL_GRID_MIN_FREQ    45
```

The feed-forward and decoupling function is implemented inside ISR1 and added for all Inverter Labs that use a current loop. Therefore, for the inverter mode, this is done (feed-forward and decoupling) in Lab 3 and Lab 4. For this feed-forward and decoupling function, filtered DC bus voltage is compared against a user-defined minimum bus voltage to calculate a clamped filtered DC bus voltage. This is also done inside ISR1. This clamped filtered DC bus voltage and the current controller output are finally used to implement the feed-forward and decoupling function.

For SDFM-based current sensing, overcurrent protection (OCP) is also added for all *Inverter Labs*.

3.2.2.1 Lab 2

In this lab, the power stage is run in an open loop on the hardware or HIL platform. Figure 3-5 shows lab setup of the actual hardware.

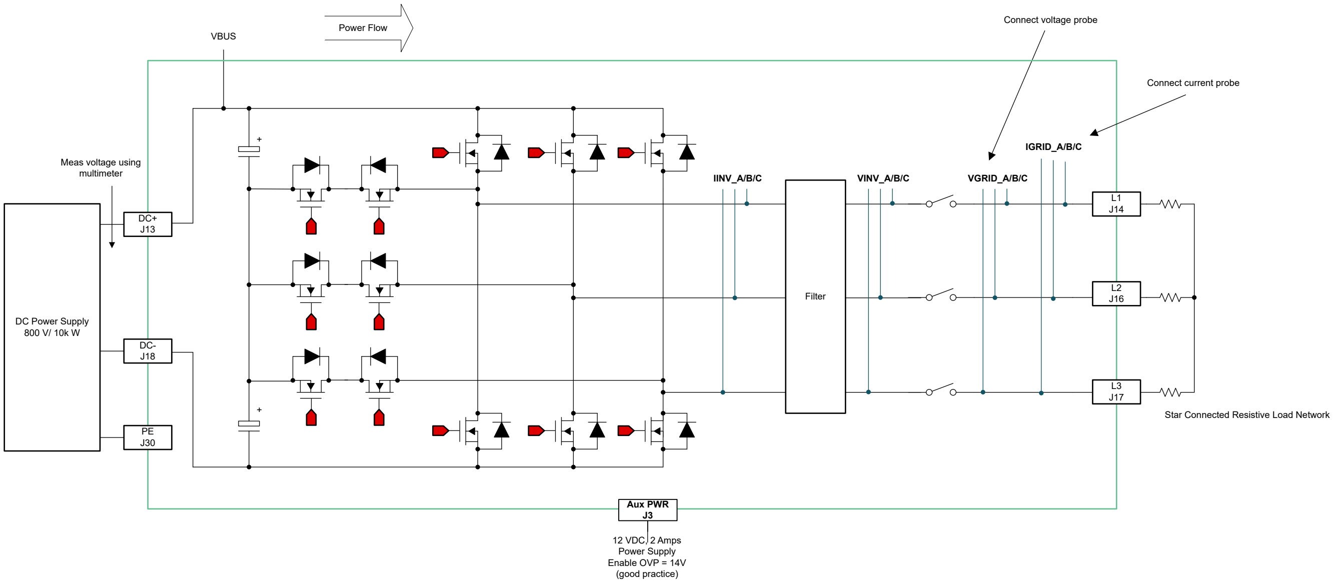
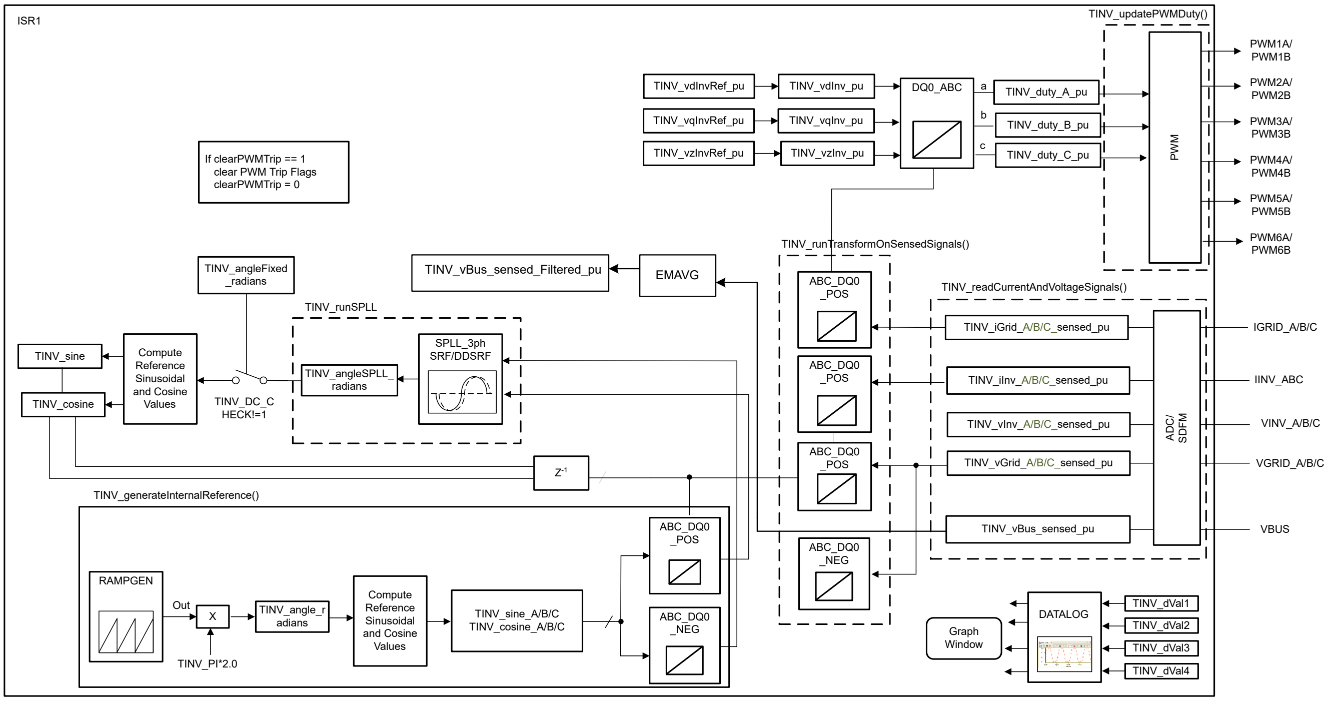

Power Flow
VBUS
Connect voltage probe
Connect current probe
Meas voltage using multimeter
DC Power Supply 800 V/ 10k W
DC+ J13
DC- J18
PE J30
IINV_A/B/C
VINV_A/B/C
VGRID_A/B/C
IGRID_A/B/C
Filter
L1 J14
L2 J16
L3 J17
Star Connected Resistive Load Network
Aux PWR J3
12 VDC, 2 Amps Power Supply Enable OVP = 14V (good practice)

Figure 3-5. Inverter Mode With Resistive Load Lab Setup

Figure 3-6 shows the software diagram.



ISR2 not shown, same as Lab 1, see Lab 1 software diagram

Figure 3-6. Lab 2 Software Diagram

See also the hardware test setup section for actual details of the equipment used for configuring the test. Set the project to Lab 2 by changing the Lab Number in the `<tinv_settings.h>` or `main.syscfg` file, (this is changed by powerSUITE GUI when using powerSUITE project).

In the `user-settings.h` file, some additional options are available, but the following are used for the tests documented in this user guide.

```
//
// Option to use SDFM based grid sensing for the current loop
// with this option the inv current from hall sensor is overwritten by the grid current from SDFM
// On Revision 5 of the hardware the only option supported is the SDFM sensing
//
#define TINV_SDFM 1
#define TINV_ADC 2
#define TINV_CURRENT_LOOP_SENSE_OPTION TINV_ADC
....
#if TINV_LAB == 2
#define TINV_TEST_SETUP TINV_TEST_SETUP_RES_LOAD
#define TINV_PROTECTION TINV_PROTECTION_ENABLED
#define TINV_SFRA_TYPE TINV_SFRA_CURRENT
#define TINV_SFRA_AMPLITUDE (float32_t)TINV_SFRA_INJECTION_AMPLITUDE_LEVEL2
#define TINV_POWERFLOW_MODE TINV_INVERTER_MODE
#define TINV_DC_CHECK 0
#define TINV_SPLL_TYPE TINV_SPLL_SRF
#endif
```

In this check, the software is run on the hardware, or the HIL platform, or both.

Set up an appropriate resistive load around 500 Ω (confirm appropriate power rating for your test) for the star connected load to start with, although the inverter mode can be started at no load as well. Enable the 12 V auxiliary supply. Build and load the code, use the `lab2.js` file to populate the watch variables in the CCS window.

- Once the CCS watch window is launched in debug mode, begin the test with continuous refresh.
- Make sure to enable the fans when testing at high power using the `TINV_fanSet` function in the CCS watch window during the debug session.
- Turn on the relays by writing a 1 to `TINV_neutralRelaySet`. The auxiliary power supply draws close to 600 mA.
- Slowly ramp the DC bus voltage *Vbus* to 800 V.
- Set the `TINV_clearPWMTrip` = 1, to clear the PWM trip signal. Now the switching action begins and sinusoidal voltages start appearing at the output. At this point, the auxiliary power supply draws close to 800 mA. With fans enabled, the total auxiliary supply current is around 1.4 A.
- `TINV_vdInvRef_pu` (default value is 0.835) is the modulation index that can be used to vary the AC output of the inverter in open-loop fashion.
- Verify the sensed voltage and current measurement data in the graph window before proceeding to close the current loop in Lab 3. Figure 3-7 is the graph window for sensed grid side current by using the C2000SDFM module. The scale is shown in per unit (pu).

```
#ifndef __TMS320C28XX_CLA__
TINV_dVal1 = TINV_iGrid_A_sensed_pu;
TINV_dVal2 = TINV_iGrid_B_sensed_pu;
TINV_dVal3 = TINV_iGrid_C_sensed_pu;
TINV_dVal4 = TINV_rgen.out;
DLOG_4CH_run(&TINV_dLog1);
#endif
```

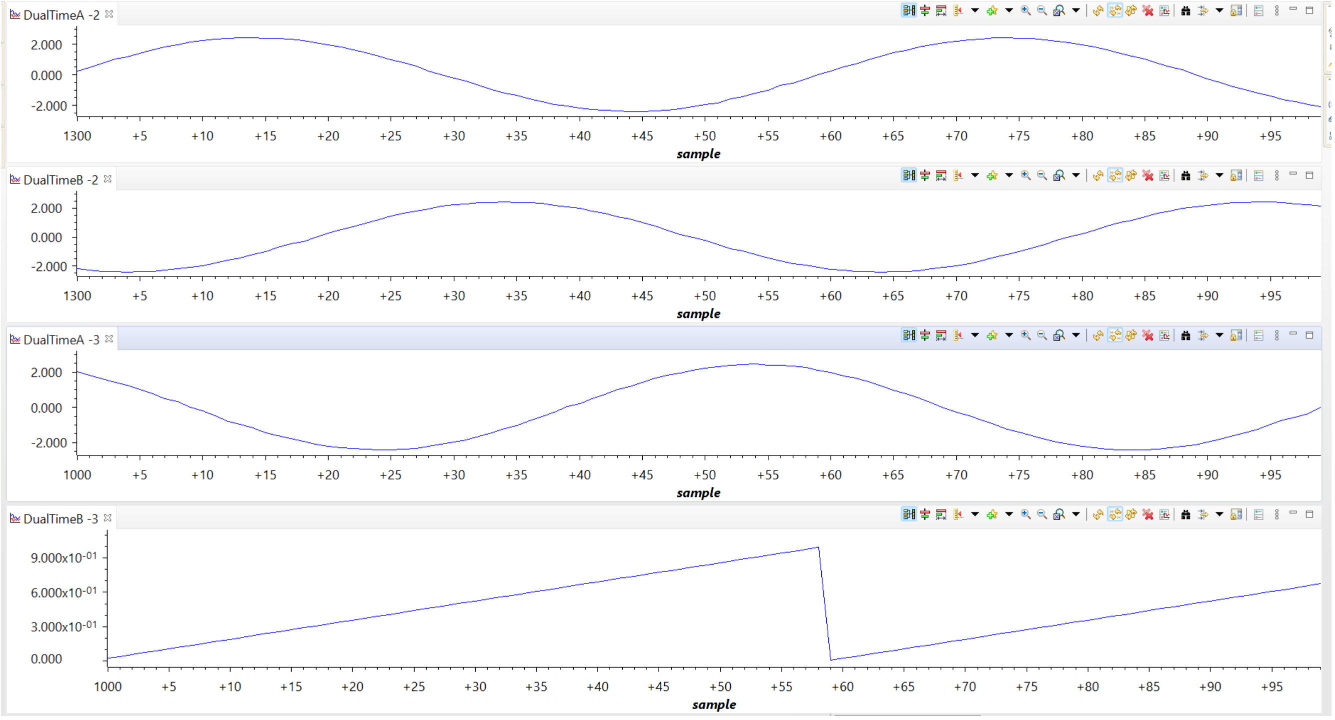


Figure 3-7. Sensed Grid Currents

Figure 3-8 shows the three grid voltages monitored from the CCS graph window. The scale is shown in per unit (pu).

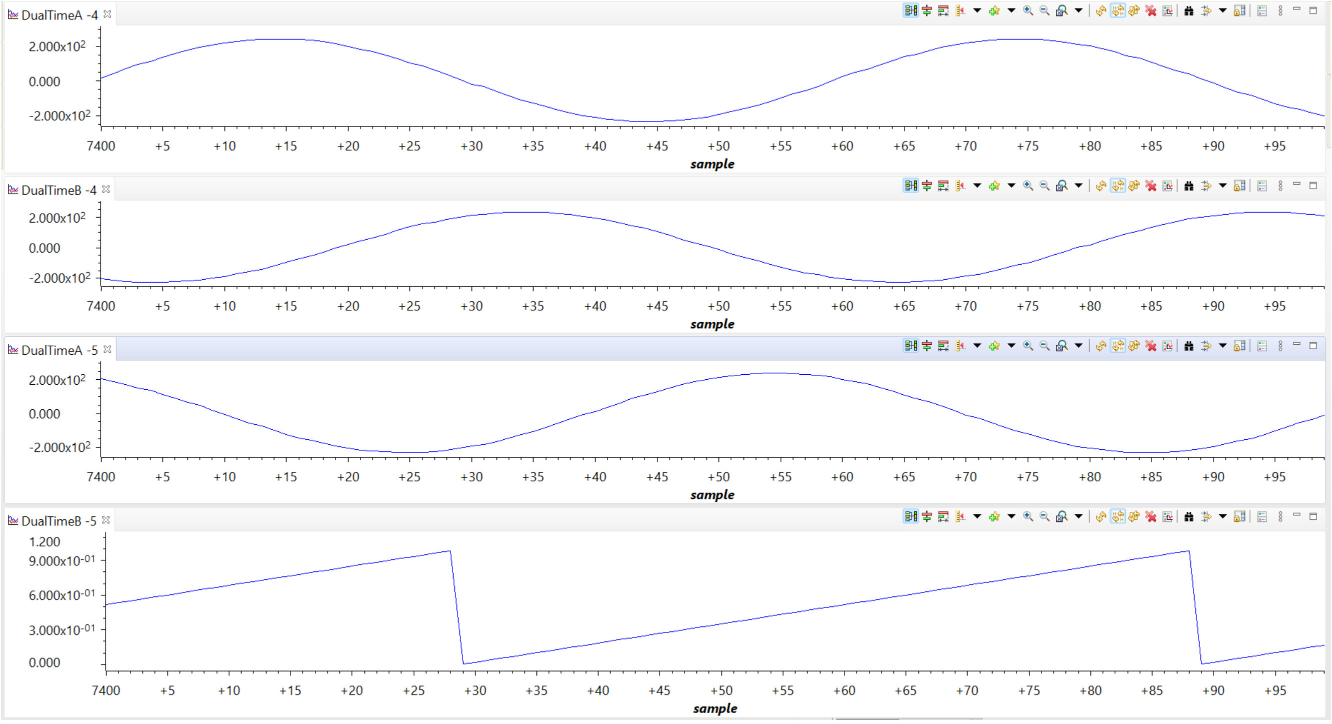
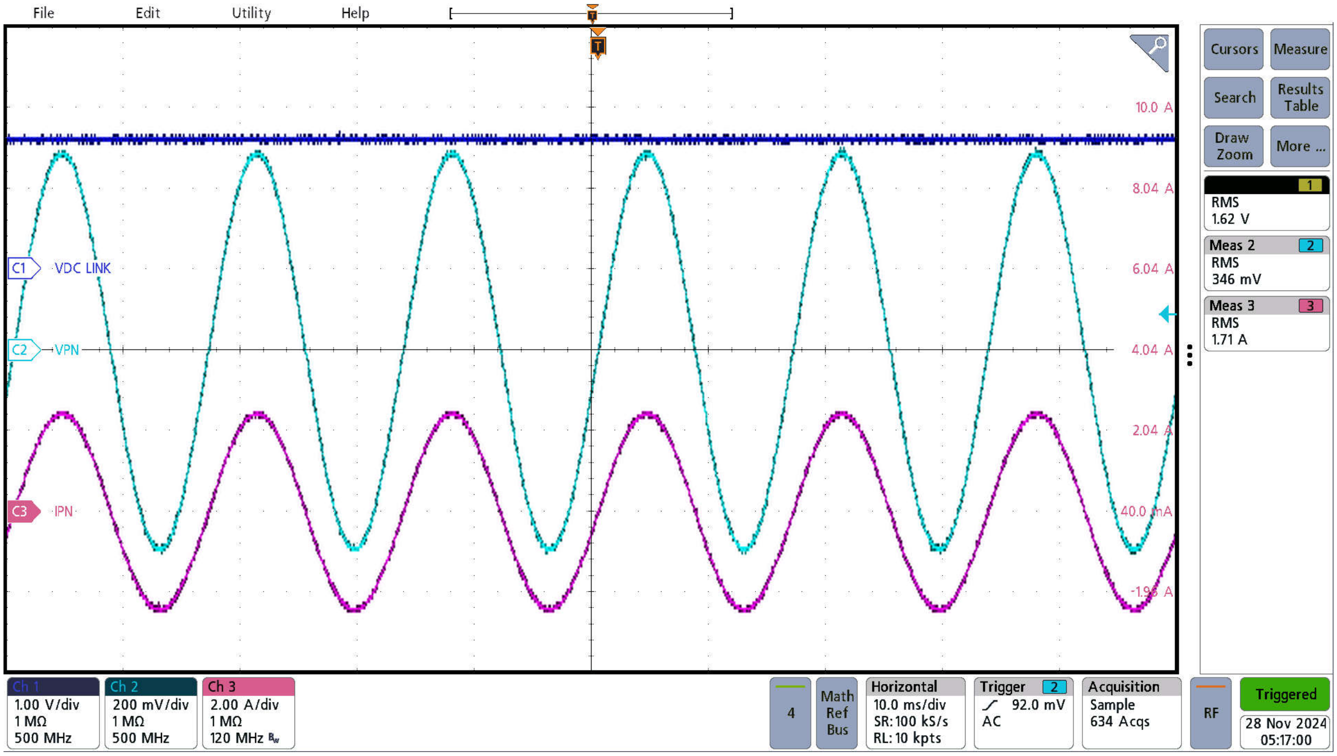


Figure 3-8. Sensed Grid Voltage

Figure 3-9 shows the captured voltage and current waveform of the inverter operating in open loop at 173 VAC and 0.88 kW.



1 RMS 1.62 V

Meas 2 RMS 346 mV

Meas 3 RMS 1.71 A

Ch 1 1.00 V/div 1 MΩ 500 MHz

Ch 2 200 mV/div 1 MΩ 500 MHz

Ch 3 2.00 A/div 1 MΩ 120 MHz

Horizontal 10.0 ms/div SR: 100 kS/s RL: 10 kpts

Trigger 2 92.0 mV AC

Acquisition Sample 634 Acqs

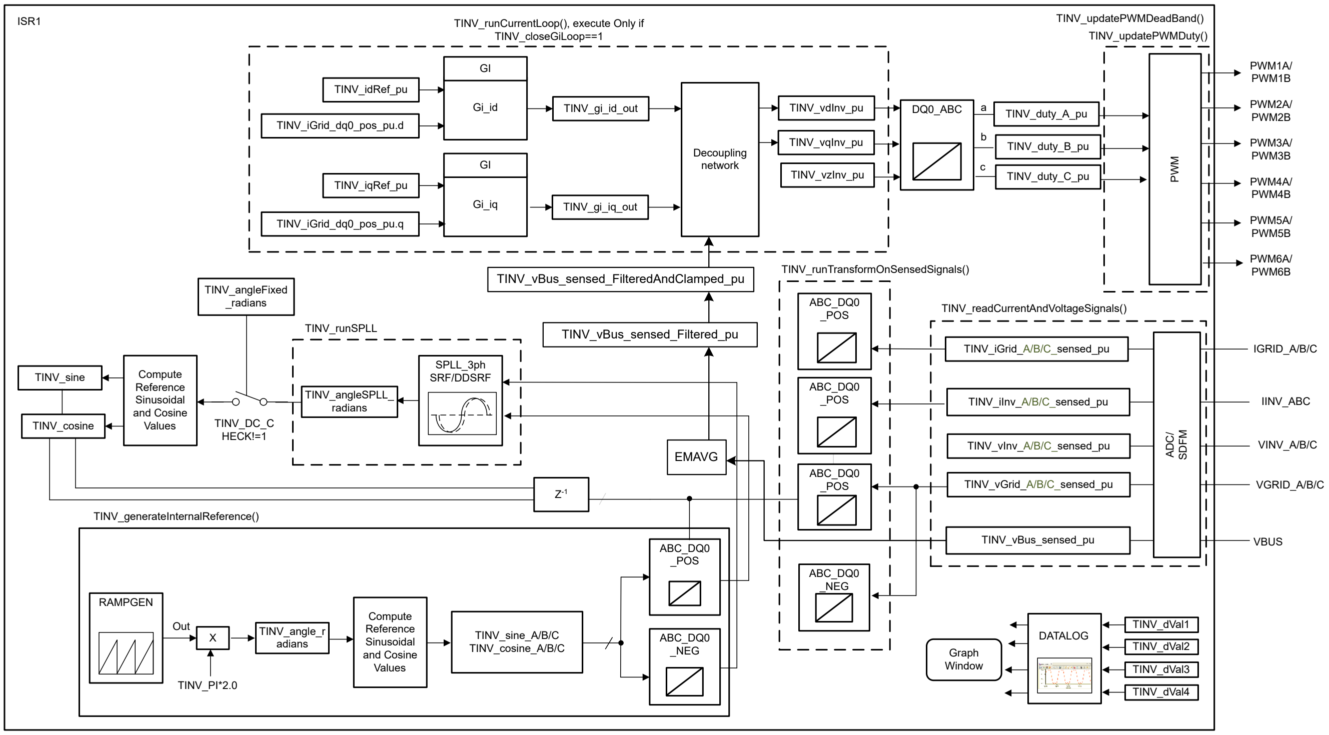
Triggered 28 Nov 2024 05:17:00

Scope signals: Channel 1 - DC link voltage (blue), Channel 2 - VPN AC voltage (turquoise), Channel 3 - IPN AC current (red). The voltage probes are scaled down at 500:1.

Figure 3-9. Open Loop Inverter Voltage and Current Waveform

3.2.2.2 Lab 3

In this lab, the power stage is run in a closed loop on the real hardware or HIL platform. Figure 3-10 shows the software diagram.



ISR2 not shown, same as Lab 1, see Lab 1 software diagram

Figure 3-10. Lab 3 Software Diagram

Set the project to Lab 3 by changing the Lab Number in the <tinv_settings.h> or main.syscfg file, (this is changed by powerSUITE GUI when using powerSUITE project).

In the user settings.h file some additional options are available, but the following are used for the tests documented in this user guide.

```
#if TINV_LAB == 3
#define TINV_TEST_SETUP TINV_TEST_SETUP_RES_LOAD
#define TINV_PROTECTION TINV_PROTECTION_ENABLED
#define TINV_SFRA_TYPE TINV_SFRA_CURRENT
#define TINV_SFRA_AMPLITUDE (float32_t)TINV_SFRA_INJECTION_AMPLITUDE_LEVEL2
#define TINV_POWERFLOW_MODE TINV_INVERTER_MODE
#define TINV_DC_CHECK 0
#define TINV_SPLL_TYPE TINV_SPLL_DDSRF
#endif
```

In this check, the software is run on the hardware, or the HIL platform, or both.

See the hardware test setup section for actual details of the equipment used for configuring the test. At this time, do not supply any high-voltage power to the board.

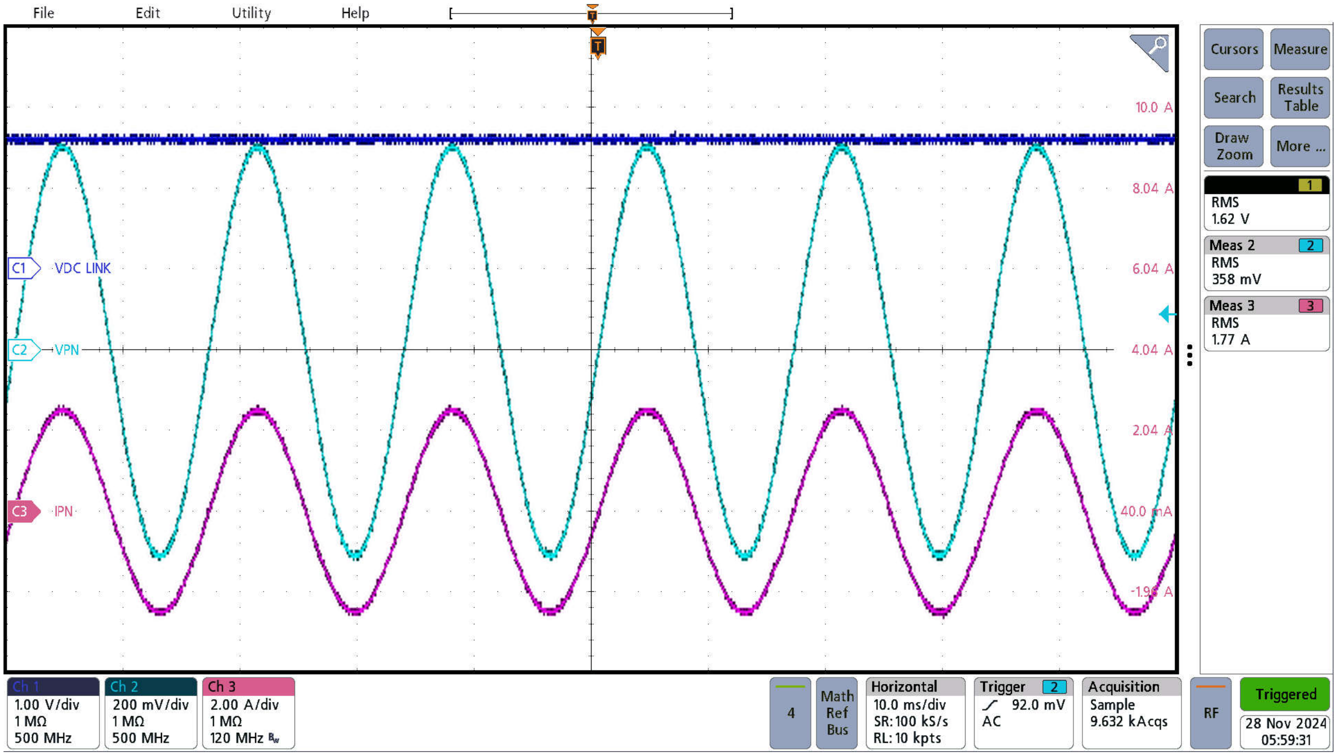
- First launch the main.cfg and select lab3 in the project options. The compensator style (PI compensator) and the tuning loop (current loop) are automatically populated. Now click the run compensation designer icon and the compensation designer tool launches, with the model of the current loop plant with parameters specified on the powerSUITE page.
- The current compensator coefficients used for running the control loop are shown in the following code. The user can modify these coefficients to meet the necessary loop bandwidth and phase margin. The ideal

coefficients with resistive load are slightly different than the one used for grid connection because the grid impedance is very low. The compensator design transfer function and response are as shown in Figure 3-11.

```
#define TINV_GI_PI_KP ((float32_t)0.0996509341)
#define TINV_GI_PI_KI ((float32_t)0.0070057718)
```

- Once satisfied with the proportional and integral gain values, click on *Save COMP*. This saves the compensator values into the project. Close the *Compensation Designer*, and return to the powerSUITE page.
- Build and load the code, use the `lab3.js` file to populate the watch variables in the CCS window.
- Make sure to enable the fans when testing at high power using the `TINV_fanSet` function in the CCS watch window during the debug session.
- Turn on the relay by writing a "1" to `TINV_neutralRelaySet`. The auxiliary power supply draws close to 600 mA.
- Set up an appropriate resistive load around 500 Ω to start with, although the inverter mode can be started at no load as well.
- Slowly ramp the DC bus voltage *Vbus* to 800 V.
- Set the `TINV_clearPWMTrip` = 1, to clear the PWM trip signal. Now the switching action begins and sinusoidal voltages start appearing at the output. At this point, the auxiliary power supply draws close to 800 mA.
- As soon as `TINV_clearPWMTrip` is set, the `TINV_closeGiLoop` variable is enabled and closed current loop action begins.
- `TINV_idRef_pu` is the current command reference and by default this reference is populated to a value of 0.005 pu at start-up. Slowly vary this to increase the output AC voltage and observe measured current tracks the commanded value.
- Verify `TINV_idRef_pu` in the watch window is at low setting (0.005 pu) before proceeding to close the current loop in Lab 3.
- Slowly increase `id_ref` to 0.06 pu at 800-V input voltage to improve output power to 0.9 kW, approximately 300 W per phase. Figure 3-11 shows the power analyzer and scope waveform.

- Figure 3-11 shows the captured voltage and current waveform of inverter operating in closed current loop at 0.9 kW.



Meas 1: RMS 1.62 V
Meas 2: RMS 358 mV
Meas 3: RMS 1.77 A

Ch 1: 1.00 V/div, 1 MΩ, 500 MHz
Ch 2: 200 mV/div, 1 MΩ, 500 MHz
Ch 3: 2.00 A/div, 1 MΩ, 120 MHz

Horizontal: 10.0 ms/div, SR: 100 kS/s, RL: 10 kpts
Trigger: 2, 92.0 mV, AC
Acquisition: Sample, 9.632 kAcqs
28 Nov 2024 05:59:31

Scope signals: Channel 1 - DC link voltage (blue), Channel 2 - AC voltage (turquoise),
Channel 3 - AC current (red) The voltage probes are scaled down at 500:1.

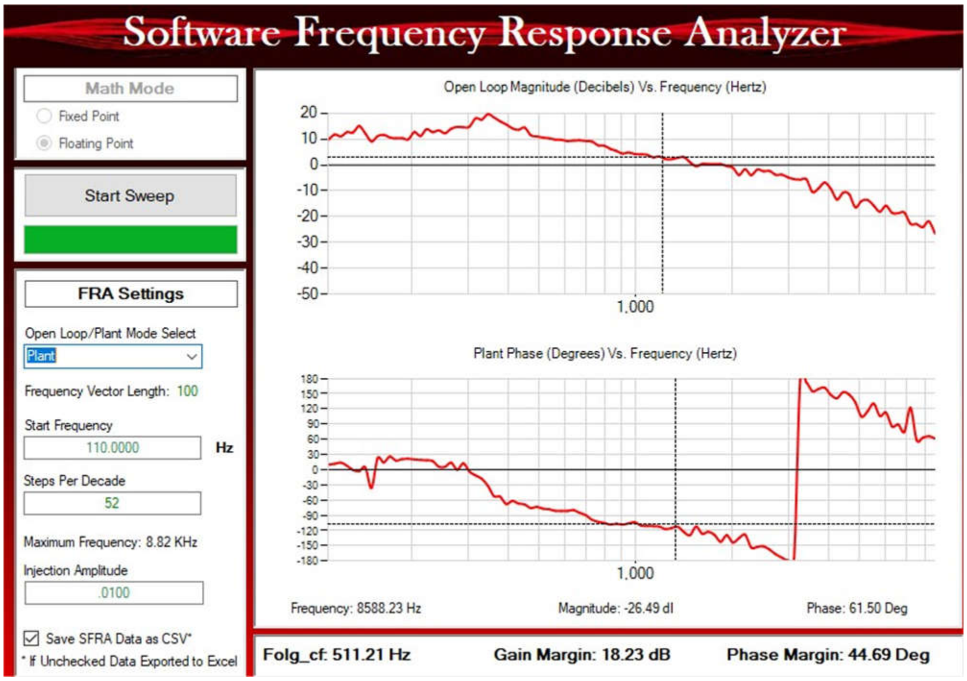
Figure 3-11. Inverter Closed-Loop Operation

- SFRA is integrated in the software of this lab to verify the designed compensator provides enough gain and phase margin by measuring on hardware. To run the SFRA keep the project running, and from the `.cfg` page, click on the SFRA icon. The SFRA GUI pops up.
- Select the options for the device on the SFRA GUI. For example, for F28379D, select floating point. Click on *Setup Connection*. On the pop-up window uncheck the boot on connect option, and select an appropriate COM port. Click *OK*. Return to the SFRA GUI, and click *Connect*.
- The SFRA GUI connects to the device. A SFRA sweep can now be started by clicking Start Sweep. The complete SFRA sweep takes a few minutes to finish. Activity can be monitored by seeing the progress bar on the SFRA GUI and also checking the flashing of blue LED on the back on the control card that indicates UART activity. Once complete, a graph with the open loop plot appears, as in Figure 3-12. This verifies that the designed compensator is indeed stable. The SFRA for plant and open loop with the above coefficients is shown in Figure 3-12 and Figure 3-13, respectively. This action verifies the current compensator design. To bring the system to a safe stop, bring the input DC voltage down to zero.
- The previously stated set of compensation designer coefficients are robust and stable. In case the tracking performance of current against the commanded reference and appears to oscillate, the user can use the following set of coefficients. To change the coefficients, the compensation designer tool must be relaunched from the power suite page.

```
#define TINV_GV_PI_KP ((float32_t) 1.9979056049)
#define TINV_GV_PI_KI ((float32_t) 0.0041887902)
```

- Once satisfied with the proportional and integral gain values, click on *Save COMP*. This saves the compensator values into the project.

- The SFRA response of plant and open loop for the inverter in current mode with the new set of coefficients are shown in Figure 3-12 and Figure 3-13, respectively.


Software Frequency Response Analyzer
Open Loop Magnitude (Decibels) Vs. Frequency (Hertz)
20
10
0
-10
-20
-30
-40
-50
1,000
Plant Phase (Degrees) Vs. Frequency (Hertz)
180
150
120
90
60
30
0
-30
-60
-90
-120
-150
-180
1,000
Start Sweep

Math Mode

○ Fixed Point

● Floating Point

FRA Settings

Open Loop/Plant Mode Select: Plant

Frequency Vector Length: 100

Start Frequency: 110.0000 Hz

Steps Per Decade: 52

Maximum Frequency: 8.82 KHz

Injection Amplitude: .0100

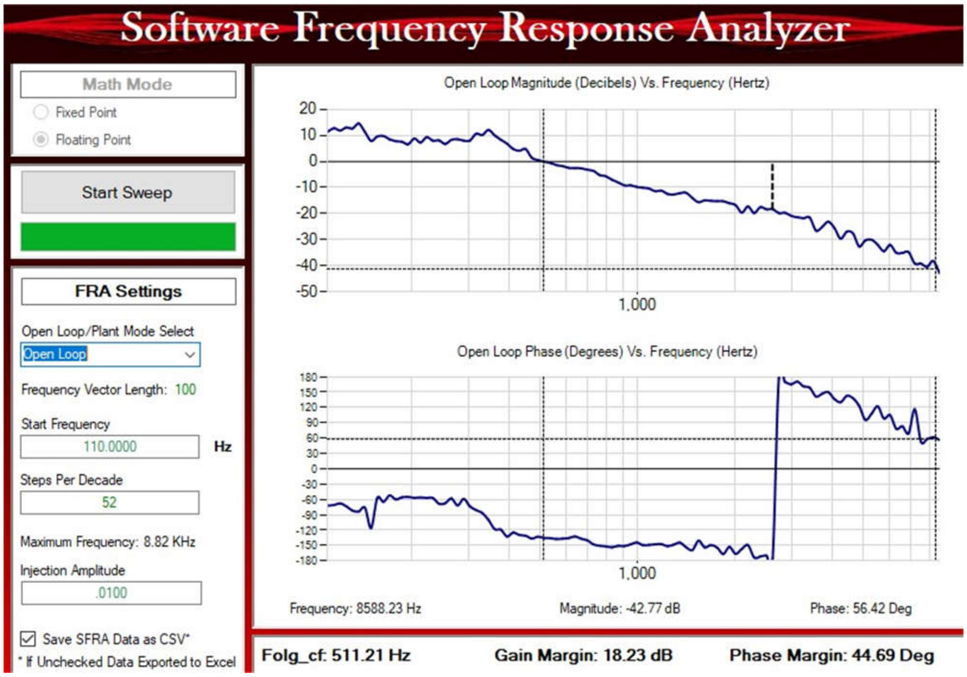
☑ Save SFRA Data as CSV*

* If Unchecked Data Exported to Excel

Frequency: 8588.23 Hz Magnitude: -26.49 dl Phase: 61.50 Deg

Folg_cf: 511.21 Hz Gain Margin: 18.23 dB Phase Margin: 44.69 Deg

Figure 3-12. Inverter SFRA Plant Response for Current Loop


Software Frequency Response Analyzer
Open Loop Magnitude (Decibels) Vs. Frequency (Hertz)
20
10
0
-10
-20
-30
-40
-50
1,000
Open Loop Phase (Degrees) Vs. Frequency (Hertz)
180
150
120
90
60
30
0
-30
-60
-90
-120
-150
-180
1,000
Start Sweep

Math Mode

○ Fixed Point

● Floating Point

FRA Settings

Open Loop/Plant Mode Select: Open Loop

Frequency Vector Length: 100

Start Frequency: 110.0000 Hz

Steps Per Decade: 52

Maximum Frequency: 8.82 KHz

Injection Amplitude: .0100

☑ Save SFRA Data as CSV*

* If Unchecked Data Exported to Excel

Frequency: 8588.23 Hz Magnitude: -42.77 dB Phase: 56.42 Deg

Folg_cf: 511.21 Hz Gain Margin: 18.23 dB Phase Margin: 44.69 Deg

Figure 3-13. Inverter SFRA Loop Response for Current Loop

3.2.2.3 Lab 4

In this lab, the power stage is run in a closed loop with grid connection on HIL platform. Figure 3-14 shows the software diagram.

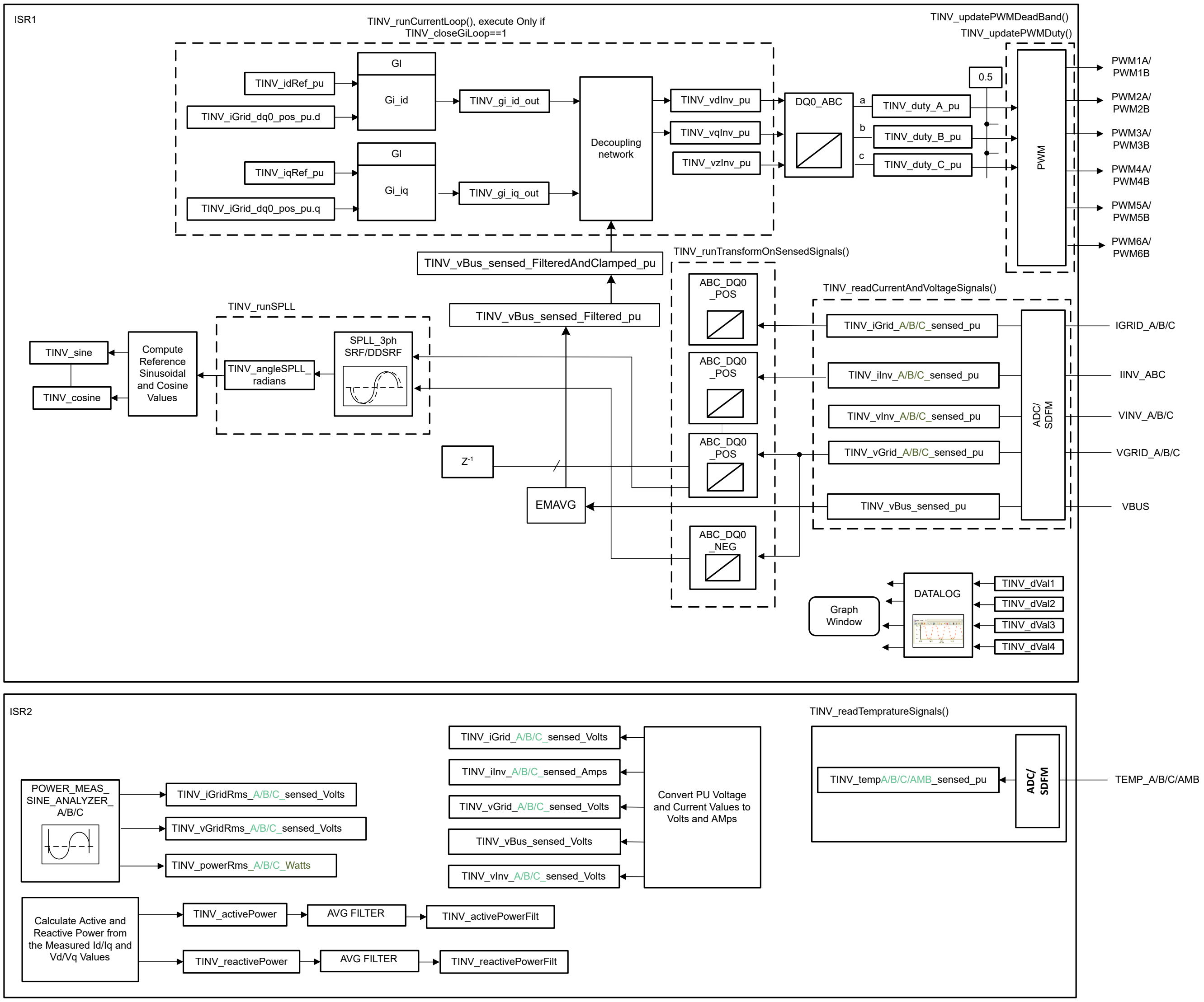

ISR1
TINV_runCurrentLoop(), execute Only if TINV_closeGiLoop==1
TINV_updatePWMDeadBand()
TINV_updatePWMDuty()
GI
Gi_id
TINV_idRef_pu
TINV_iGrid_dq0_pos_pu.d
TINV_gi_id_out
GI
Gi_iq
TINV_iqRef_pu
TINV_iGrid_dq0_pos_pu.q
TINV_gi_iq_out
Decoupling network
TINV_vdInv_pu
TINV_vqInv_pu
TINV_vzInv_pu
DQ0_ABC
a
b
c
0.5
TINV_duty_A_pu
TINV_duty_B_pu
TINV_duty_C_pu
PWM
PWM1A/ PWM1B
PWM2A/ PWM2B
PWM3A/ PWM3B
PWM4A/ PWM4B
PWM5A/ PWM5B
PWM6A/ PWM6B
TINV_vBus_sensed_FilteredAndClamped_pu
TINV_runTransformOnSensedSignals()
TINV_vBus_sensed_Filtered_pu
TINV_runSPLL
TINV_readCurrentAndVoltageSignals()
TINV_sine
TINV_cosine
Compute Reference Sinusoidal and Cosine Values
TINV_angleSPLL_radians
SPLL_3ph SRF/DDSRF
ABC_DQ0 _POS
ABC_DQ0 _POS
ABC_DQ0 _POS
ABC_DQ0 _NEG
TINV_iGrid_A/B/C_sensed_pu
TINV_iInv_A/B/C_sensed_pu
TINV_vInv_A/B/C_sensed_pu
TINV_vGrid_A/B/C_sensed_pu
TINV_vBus_sensed_pu
ADC/ SDFM
IGRID_A/B/C
IINV_ABC
VINV_A/B/C
VGRID_A/B/C
VBUS
Z-1
EMAVG
DATALOG
Graph Window
TINV_dVal1
TINV_dVal2
TINV_dVal3
TINV_dVal4
ISR2
TINV_readTempratureSignals()
POWER_MEAS_ SINE_ANALYZER_ A/B/C
TINV_iGridRms_A/B/C_sensed_Volts
TINV_vGridRms_A/B/C_sensed_Volts
TINV_powerRms_A/B/C_Watts
TINV_iGrid_A/B/C_sensed_Volts
TINV_iInv_A/B/C_sensed_Amps
TINV_vGrid_A/B/C_sensed_Volts
TINV_vBus_sensed_Volts
TINV_vInv_A/B/C_sensed_Volts
Convert PU Voltage and Current Values to Volts and AMps
TINV_tempA/B/C/AMB_sensed_pu
ADC/ SDFM
TEMP_A/B/C/AMB
Calculate Active and Reactive Power from the Measured Id/Iq and Vd/Vq Values
TINV_activePower
AVG FILTER
TINV_activePowerFilt
TINV_reactivePower
AVG FILTER
TINV_reactivePowerFilt

Figure 3-14. Lab 4 Software Diagram

Note

This lab is only verified under HIL setup

Set the project to Lab 4 by changing the Lab Number in the `<tinv_settings.h>` or `main.syscfg` file, (this is changed by powerSUITE GUI when using powerSUITE project).

In the user `settings.h` file some additional options are available, but the following are used for the tests documented in this user guide.

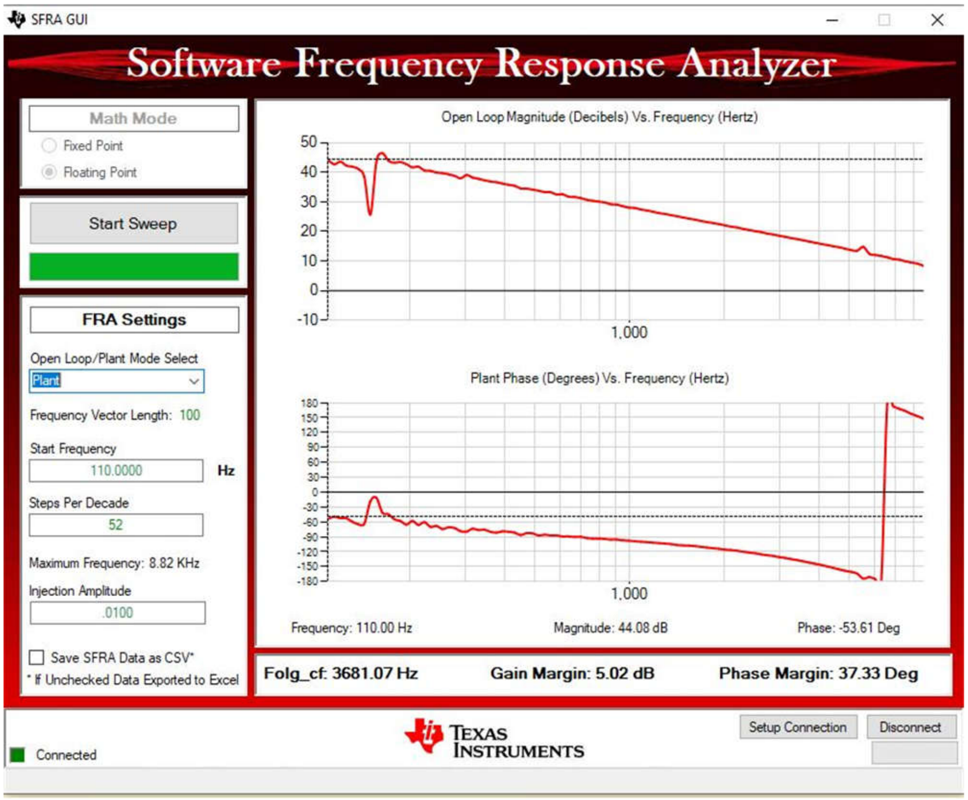
```
#if TINV_LAB == 4
#define TINV_TEST_SETUP TINV_TEST_SETUP_GRID_CONNECTED
#define TINV_PROTECTION TINV_PROTECTION_ENABLED
#define TINV_SFRA_TYPE TINV_SFRA_CURRENT
#define TINV_SFRA_AMPLITUDE (float32_t)TINV_SFRA_INJECTION_AMPLITUDE_LEVEL2
#define TINV_POWERFLOW_MODE TINV_INVERTER_MODE
#define TINV_DC_CHECK 0
#define TINV_SPLL_TYPE TINV_SPLL_SRF
#endif
```

In this check, the software is run on the HIL platform.

Build and load the code, use the `lab4.js` file to populate the watch variables in the CCS window.

- Make sure to enable the fans when testing at high power using the `TINV_fanSet` function in the CCS watch window during the debug session.
- Slowly ramp the DC bus voltage *Vbus* to 800 V
- Enter *"1"* on the `TINV_startPowerStage` variable and ramp the AC voltage of the grid to 230V_{RMS}, L-N; that is, 400 V_{L-L}.
- Set up an appropriate grid connection and turn on the relay by writing a *"1"* to `TINV_neutralRelaySet` as soon as the voltage reaches 230 V_{RMS} as the inrush current limit resistors can get heated and burn out. The current is now supposed to be fed into the grid.
- Slowly increase this `TINV_idRef_pu` variable to be 0.6 pu, at this point the per phase power is approximately 1.9 kW.

Measure the current loop bandwidth using SFRA. Figure 3-15 and Figure 3-16 show plant response and loop response of inverter operating in current mode measured on the HIL platform.



SFRA GUI

Software Frequency Response Analyzer

Math Mode

○ Fixed Point

● Floating Point

Start Sweep

FRA Settings

Open Loop/Plant Mode Select: Plant

Frequency Vector Length: 100

Start Frequency: 110.0000 Hz

Steps Per Decade: 52

Maximum Frequency: 8.82 KHz

Injection Amplitude: .0100

□ Save SFRA Data as CSV*

* If Unchecked Data Exported to Excel

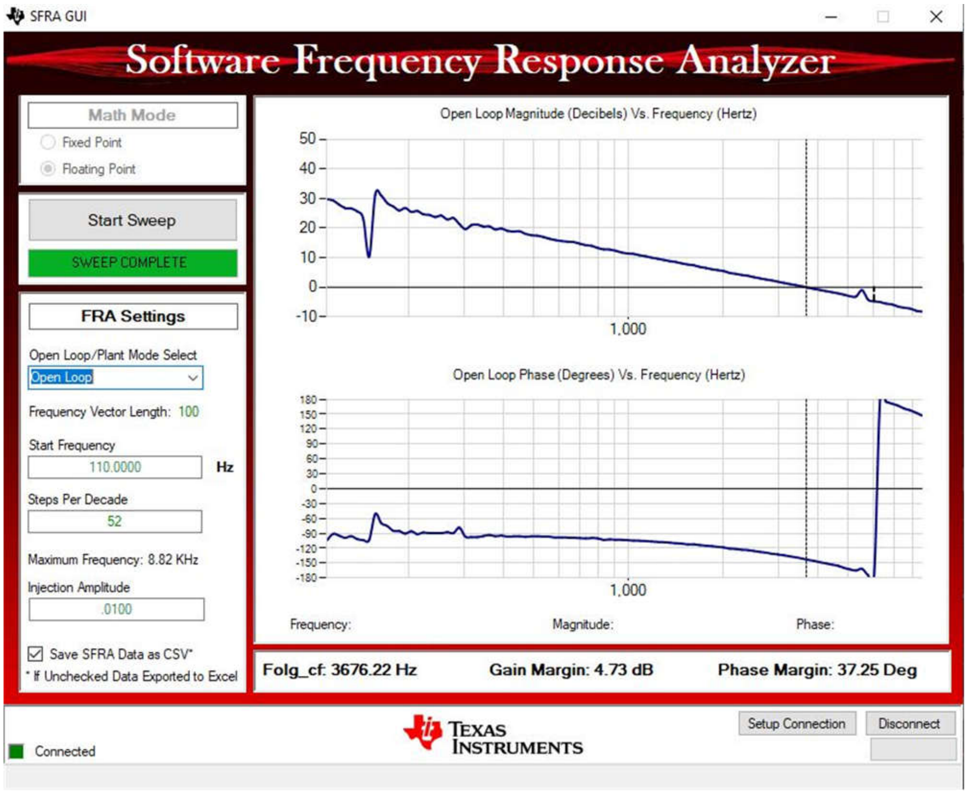
Frequency: 110.00 Hz Magnitude: 44.08 dB Phase: -53.61 Deg

Folg_cf: 3681.07 Hz Gain Margin: 5.02 dB Phase Margin: 37.33 Deg

🟩 Connected

Open Loop Magnitude (Decibels) Vs. Frequency (Hertz)
50
40
30
20
10
0
-10
1,000
Plant Phase (Degrees) Vs. Frequency (Hertz)
180
150
120
90
60
30
0
-30
-60
-90
-120
-150
-180
1,000
Setup Connection
Disconnect

Figure 3-15. Inverter SFRA Plant Response Measured on HIL



○ Fixed Point

● Floating Point

Open Loop/Plant Mode Select: Open Loop

Frequency Vector Length: 100

Start Frequency: 110.0000 Hz

Steps Per Decade: 52

Maximum Frequency: 8.82 KHz

Injection Amplitude: .0100

☑ Save SFRA Data as CSV*

* If Unchecked Data Exported to Excel

Folg_cf: 3676.22 Hz Gain Margin: 4.73 dB Phase Margin: 37.25 Deg

🟩 Connected

Figure 3-16. Inverter SFRA Loop Response Measured on HIL

3.2.3 Testing PFC Operation

Lab 5, *Lab 6*, and *Lab 7* elaborate the steps for running the power stage in the PFC mode. *Lab 5* is the PFC mode of operation in open loop. Lab 6 is the PFC mode of operation with closed current loop. *Lab 7* is the PFC mode of operation in closed voltage and current loop and this is checked both on the HIL platform and actual hardware.

A check for DC bus overvoltage is added to all *PFC Labs*, *Lab 5* through *Lab 7*, using a filtered value of the DC bus voltage. The `TINV_filterAndCheckForBusOverVoltage()` function runs from ISR1 and checks for DC bus overvoltage condition. Under the overvoltage condition this function shuts off all PWM outputs and registers the system operating state as *bus overvoltage state*. Filtered DC bus voltage is calculated from instantaneous sensed DC bus voltage using the averaging function EMAVG. This is all calculated inside ISR1.

The feed-forward and decoupling function is implemented inside ISR1 and added for all *PFC Labs* that use a current loop. Therefore, for the PFC mode, this is done in *Lab 6* and *Lab 7*. For this feed-forward and decoupling function filtered DC bus voltage is compared against a user-defined minimum bus voltage to calculate a clamped filtered DC bus voltage. This is also done inside ISR1. This clamped filtered DC bus voltage and the current controller output are finally used to implement the feed-forward and decoupling function.

For SDFM-based current sensing, overcurrent protection (OCP) is also added for all PFC labs.

Figure 3-17 shows the hardware setup, the DC terminals J13 and J18 are connected to an e-load. A 12-V auxiliary power supply is connected to terminal J3. Three-phase AC source is connected across terminals J14, J16, and J17 (A, B, and C). J30 is the PE terminal which is connected to the source PE. See the hardware test setup section for actual details of the equipment used for configuring the test.

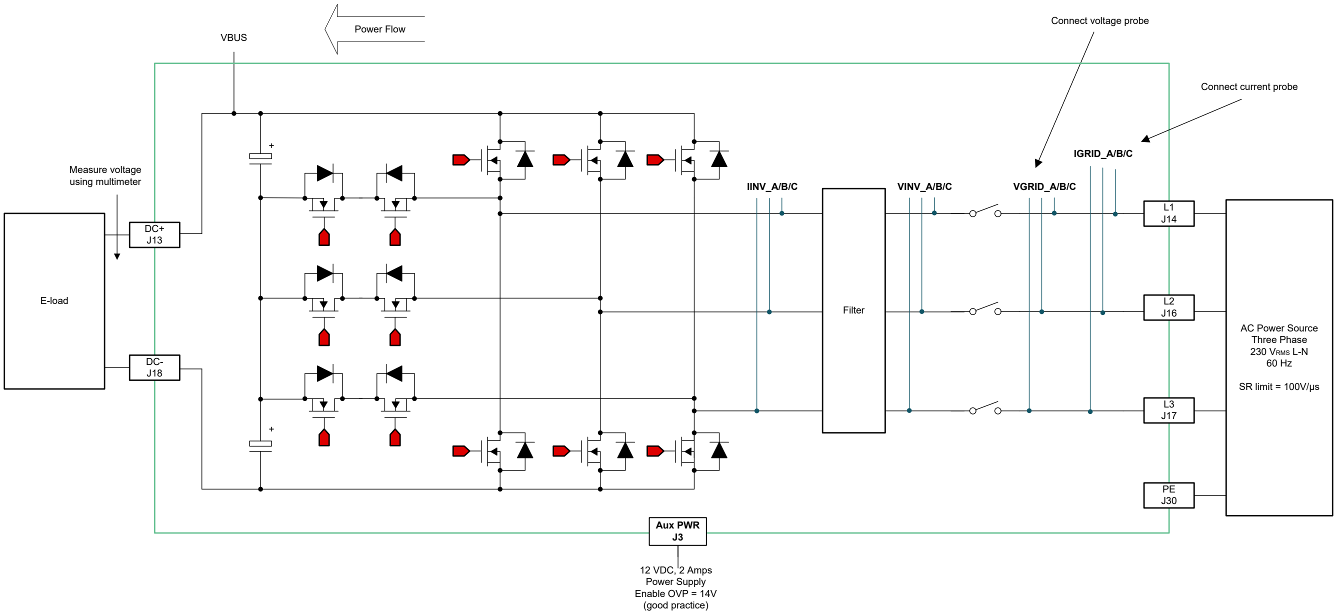


Figure 3-17. PFC Mode Test Setup

3.2.3.1 Lab 5

This is the first PFC lab. In this lab sensing is checked and no switching action occurs until `clearPWMTrip` is set to *1*.

The hardware setup for the PFC mode is explained in Figure 3-17. TI recommends starting the PFC at a low voltage like 30 V_{RMS} and connecting a 2-kΩ resistor.

Set the project to Lab 5 by changing the Lab Number in the `<tinv_settings.h>` or `main.syscfg` file, (this is changed with the powerSUITE GUI when using the powerSUITE project).

Under this condition, the converter operates as a rectifier and rectified current can be observed being drawn without any power factor correction. Software Phase-Lock Loop (SPLL) locking can also be safely verified in this build.

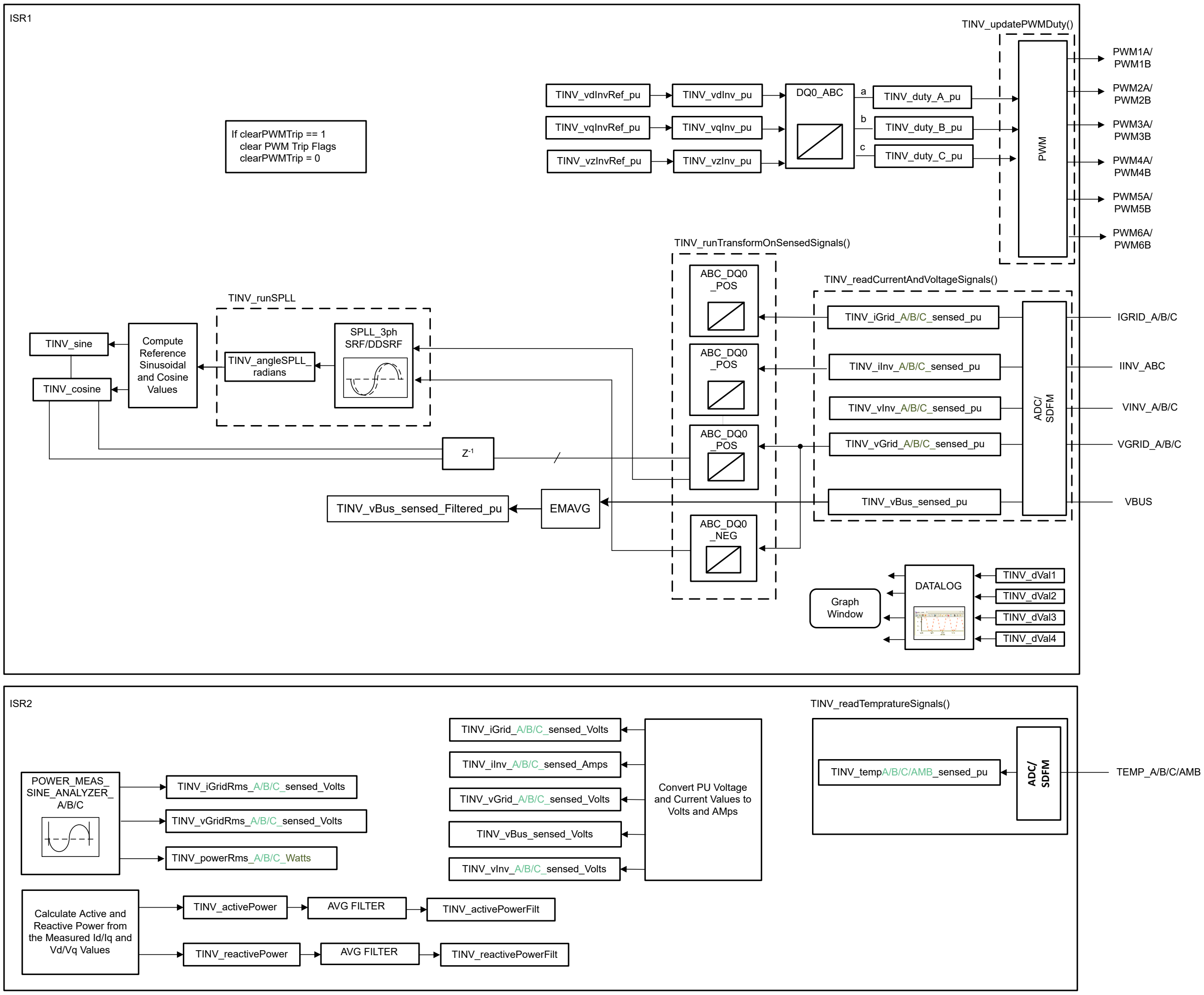


Figure 3-18. Lab 5 Software Diagram

Hence, the following variables are put on the data logger:

```
TINV_dVal1 = TINV_vGrid_A_sensed_pu;
TINV_dVal2 = TINV_angleSPLL_radians / (float32_t)(2.0f * TINV_PI);
TINV_dVal3 = TINV_vGrid_A_sensed_pu;
TINV_dVal4 = TINV_iInv_A_sensed_pu;
DLOG_4CH_run(&TINV_dLog1);
```

Make sure the grid frequency is specified correctly, the grid frequency can be changed through the `sysconfig` page for powerSUITE-based projects. If not using a powerSUITE-based project, modify the `tinv_settings.h` file.

```
#define TINV_AC_FREQ_HZ ((float32_t)50)
```

Build and load the code, use the `lab5.js` file to populate the watch variables in the CCS window.

PLL lock can be checked by plotting the buffers. Use the `graph1.graphprop` to see the buffer through *Tools → Graph → Dual Time*.

Cosine transforms are used; therefore, the angle is *0* when Vgrid peaks.

Close the relay by writing a *1* to `TINV_neutralRelaySet`.

Initially, only run this test with 30 V_{RMS} for safety, hence safely ramp the AC supply to 30 V_{RMS} and observe the graph in the CCS debug window to confirm the PLL is locking. Figure 3-19 shows the low-voltage phase-locked loop check from watch window.

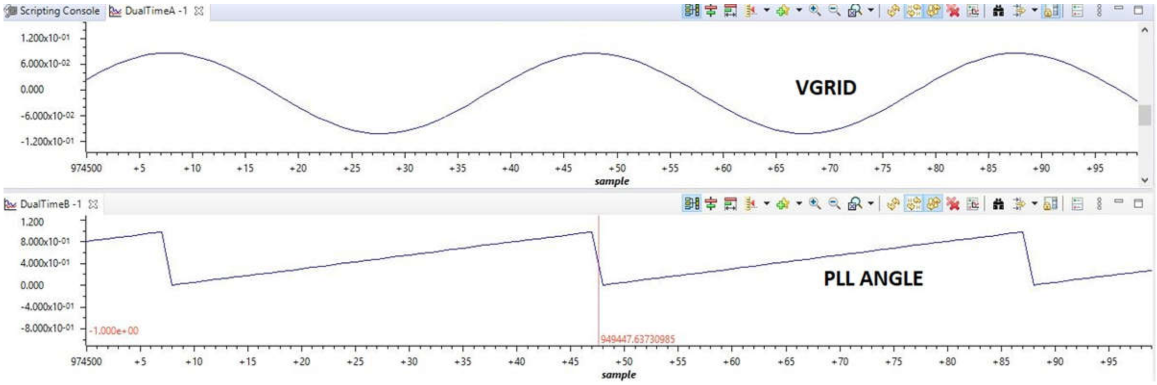


Figure 3-19. PLL - Grid Voltage Synchronization

If the PLL is not locking, issue a `tinv_reset_PLL` command by setting the command to *"1"*, which initiates a task to *zero out* an integrated error in the module and *zero* all the memory elements.

Similarly, the current flowing from the grid across all phases can be checked, using the graph watch window of CCS. Figure 3-20 shows the sensed grid currents from graph `windowCheck` for three phase grid currents observed from the watch window.

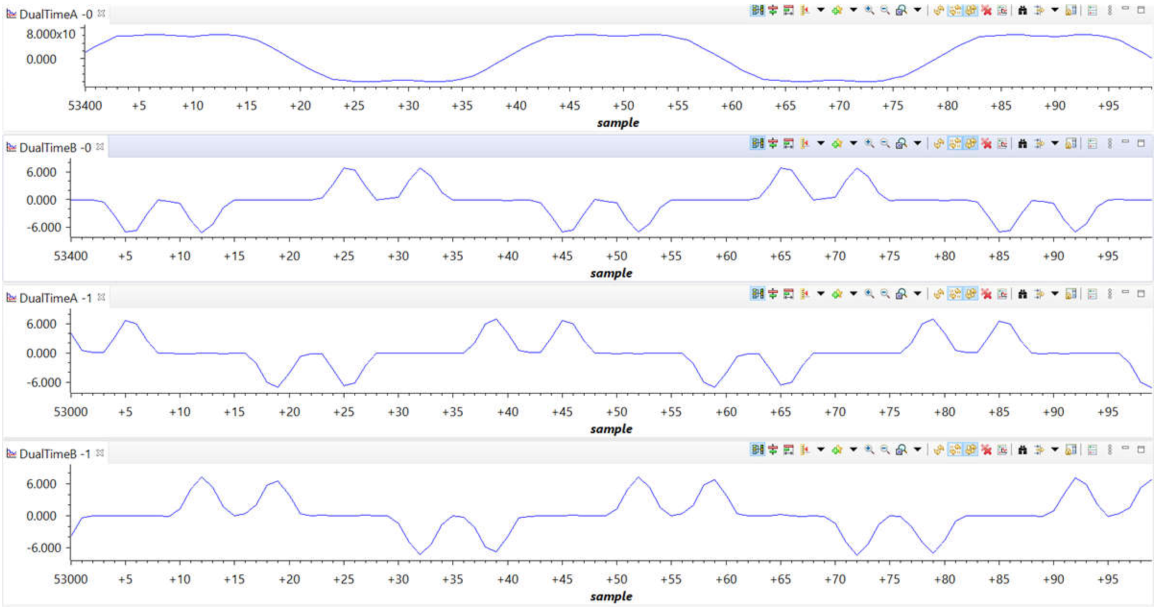


Figure 3-20. Sensed Grid Currents - PFC Mode

To verify boost action in Lab 5, follow the steps according to the sequence provided:

- Turn on the auxiliary power supply, set at 12 V and then debug and run the code.
- Connect an acceptable load to the J13 and J18 terminals. Make sure to use a high load resistance (around 2 kΩ) which otherwise can lead to high inrush currents triggering the overcurrent flag. The e-load can also be used in constant voltage (CV) mode. Set the clamp voltage higher than 2.6 × the input AC phase voltage. For

example, if 30 V_{RMS} AC voltage is used, set the e-load CV voltage to at least 78 V (higher than the unboosted rectified voltage at the DC link).

- Make sure to enable the fans when testing at high power using the `TINV_fanSet` function in the CCS watch window during the debug session.
- Apply 30 V_{RMS} AC voltage to the three phase terminals.
- Immediately turn on the relay by writing a *1* to `TINV_neutralRelaySet`. Voltage starts to appear across the DC terminals.
- Clear the PWM trip by setting `TINV_clearPwmTrip` to *1* to see a slight boost in DC voltage.

Before PFC action begins, a rectified current is drawn due to the load on the Vbus. As soon as `clearPWMTrip` is set to *1*, a slight boost in DC voltage is evident.

Note

There can be a situation in the labs for PFC (Lab 5, Lab 6, and Lab 7) where the converter operates as a rectifier and rectified current is seen being drawn without any power factor correction. But as soon as `TINV_clearPwmTrip` is set to *1*, there is no switching action – the Gate Signals remain off.

This is because there is an overcurrent or DSAT flag (`InvA_overcurrent`, `InvB_overcurrent`,`DSATA`, `DSATB`) which is set in one of the three phases and this happens under three circumstances:

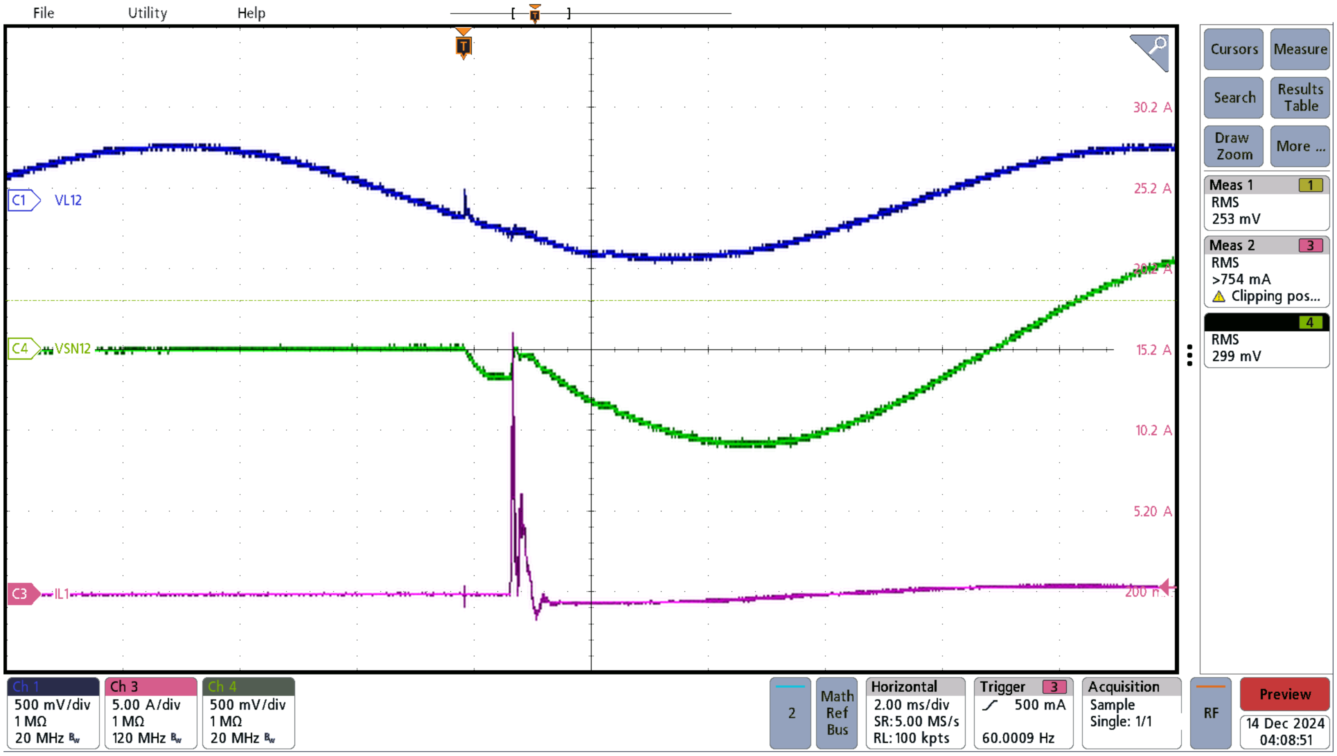
1. On closing the relays, there is an inrush current which creates an overcurrent trip in one of the three phases.
2. When `TINV_clearPwmTrip` is set to *1*, the switching action causes one of the flags to be set.
3. Setting `TINV_StartpowerStage` to *1* for closing the current and voltage loop.

The EPWM TZFLG is set to 0X000C and under this condition no switching occurs. So make sure the load resistance is increased so that the inrush currents do not cause a trip condition and the EPWM TZFLG changes from 0x0004 to 0x0000 and switching occurs.

Once the FLG is set to 0x000C, even if a `TINV_reset_fault_status` is performed to reset the faults and even though the faults are cleared, PWM action is not observed.

The goal is, as soon as the auxiliary power supply is started and the code debugged, all the faults – namely `InvA_overcurrent`, `InvB_overcurrent`, `DSATA`, `DSATB`; and so forth – are supposed to be set to zero so that the controller does not go into a trip state.

If possible, limit the slew rate on the AC source to 100 V/µs. This helps with the inrush current tripping the AC source OCP. This can happen if when the relay is left open for a long time causing the voltage on the EMI filter caps to decay close to 0 V, as shown in Figure 3-21. Later when the relay is closed, a large inrush current is in the reactive load which can trip the AC source OCP feature.



Meas 1 (1)
RMS
253 mV

Meas 2 (3)
RMS
>754 mA
⚠ Clipping pos...

(4)
RMS
299 mV

Ch 1
500 mV/div
1 MΩ
20 MHz

Ch 3
5.00 A/div
1 MΩ
120 MHz

Ch 4
500 mV/div
1 MΩ
20 MHz

Horizontal
2.00 ms/div
SR:5.00 MS/s
RL:100 kpts

Trigger (3)
500 mA
60.0009 Hz

Acquisition
Sample
Single: 1/1

14 Dec 2024
04:08:51

Scope signals: Channel 1 - $V_{L\text{-}L}$ at input (blue), Channel 4 - voltage across EMI cap C68 VSN (light green), Channel 3 - AC input current (red). The voltage probes are scaled down at 500:1.

Figure 3-21. Inrush Current Due to Reactive Load Voltage Decay Following Precharge Period

3.2.3.2 Lab 6

The current loop for the PFC is checked in this build level. Lab 6 is primarily meant for tuning the current loop and optimizing performance. This lab can be safely started at low voltage and low power because starting at higher power without a supervisory voltage loop can boost the voltage due to overcurrent events and cause blown switches. Carefully set the `TINV_idRef_pu` variable so as to avoid overcurrent trips and high voltages at the DC terminals. Also the `TINV_idRef_pu` is defined with a negative sign for PFC mode of operation and with a positive sign for inverter mode of operation.

Figure 3-22 describes the software flow for running Lab 6.

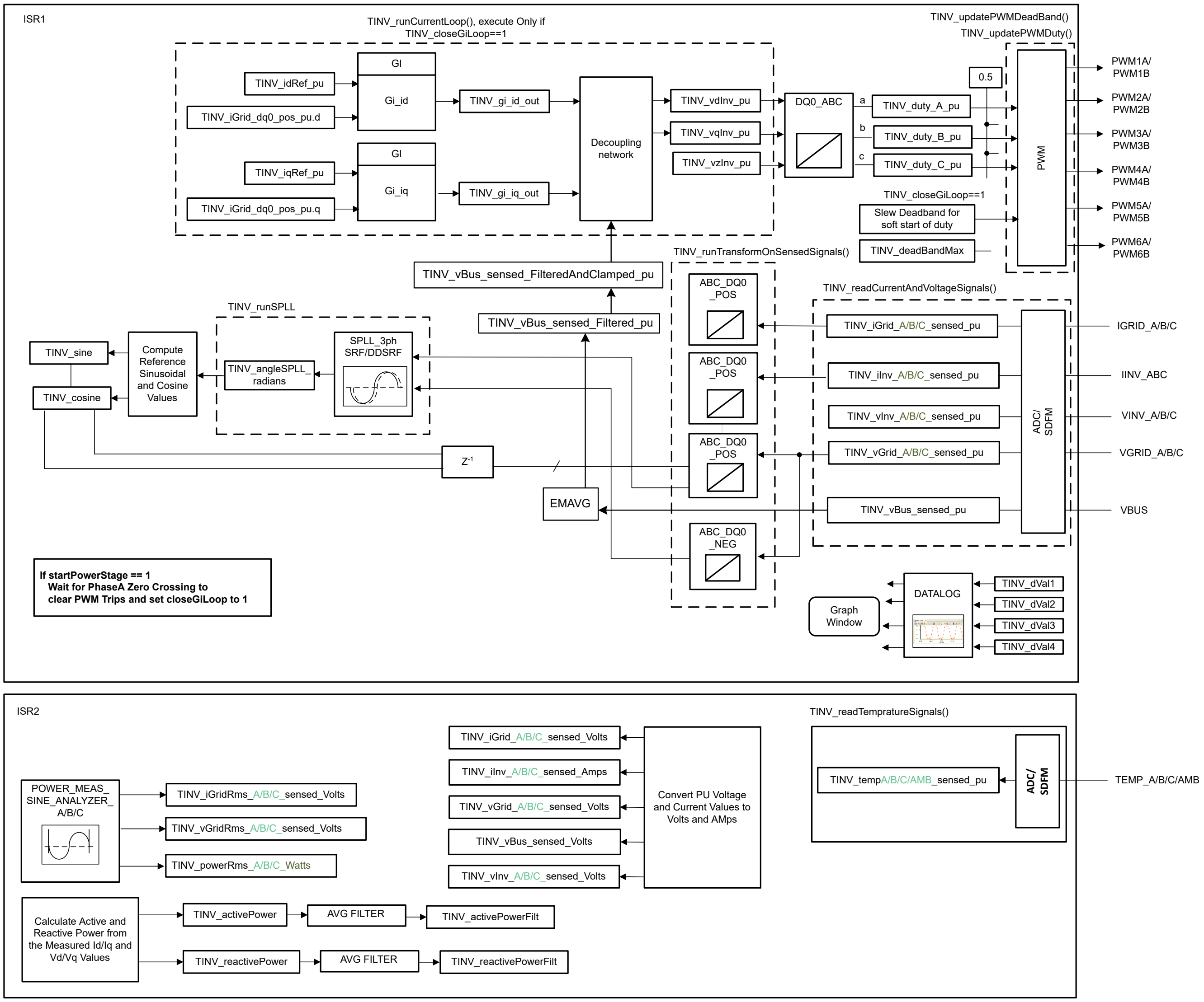

ISR1
TINV_runCurrentLoop(), execute Only if TINV_closeGiLoop==1
TINV_updatePWMDeadBand()
TINV_updatePWMDuty()
TINV_idRef_pu
TINV_iGrid_dq0_pos_pu.d
GI
Gi_id
TINV_gi_id_out
TINV_iqRef_pu
TINV_iGrid_dq0_pos_pu.q
GI
Gi_iq
TINV_gi_iq_out
Decoupling network
TINV_vdInv_pu
TINV_vqInv_pu
TINV_vzInv_pu
DQ0_ABC
a
b
c
TINV_duty_A_pu
TINV_duty_B_pu
TINV_duty_C_pu
0.5
PWM
PWM1A/ PWM1B
PWM2A/ PWM2B
PWM3A/ PWM3B
PWM4A/ PWM4B
PWM5A/ PWM5B
PWM6A/ PWM6B
TINV_closeGiLoop==1
Slew Deadband for soft start of duty
TINV_deadBandMax
TINV_vBus_sensed_FilteredAndClamped_pu
TINV_runTransformOnSensedSignals()
TINV_readCurrentAndVoltageSignals()
TINV_runSPLL
TINV_vBus_sensed_Filtered_pu
TINV_sine
TINV_cosine
Compute Reference Sinusoidal and Cosine Values
TINV_angleSPLL_ radians
SPLL_3ph SRF/DDSRF
ABC_DQ0 _POS
ABC_DQ0 _POS
ABC_DQ0 _POS
ABC_DQ0 _NEG
TINV_iGrid_A/B/C_sensed_pu
TINV_iInv_A/B/C_sensed_pu
TINV_vInv_A/B/C_sensed_pu
TINV_vGrid_A/B/C_sensed_pu
TINV_vBus_sensed_pu
ADC/ SDFM
IGRID_A/B/C
IINV_ABC
VINV_A/B/C
VGRID_A/B/C
VBUS
Z-1
EMAVG
If startPowerStage == 1
Wait for PhaseA Zero Crossing to clear PWM Trips and set closeGiLoop to 1
DATALOG
Graph Window
TINV_dVal1
TINV_dVal2
TINV_dVal3
TINV_dVal4
ISR2
TINV_readTempratureSignals()
TINV_iGrid_A/B/C_sensed_Volts
TINV_iInv_A/B/C_sensed_Amps
TINV_vGrid_A/B/C_sensed_Volts
TINV_vBus_sensed_Volts
TINV_vInv_A/B/C_sensed_Volts
Convert PU Voltage and Current Values to Volts and AMps
TINV_tempA/B/C/AMB_sensed_pu
ADC/ SDFM
TEMP_A/B/C/AMB
POWER_MEAS_ SINE_ANALYZER_ A/B/C
TINV_iGridRms_A/B/C_sensed_Volts
TINV_vGridRms_A/B/C_sensed_Volts
TINV_powerRms_A/B/C_Watts
Calculate Active and Reactive Power from the Measured Id/Iq and Vd/Vq Values
TINV_activePower
AVG FILTER
TINV_activePowerFilt
TINV_reactivePower
AVG FILTER
TINV_reactivePowerFilt

Figure 3-22. Lab 6 Software Diagram

Set the project to Lab 6 by changing the *lab number* in the `<tinv_settings.h>` or `main.syscfg` file, (this is changed by powerSUITE GUI when using powerSUITE project)

In the user `settings.h` file some additional options are available, but the following are used for the tests documented in this design guide.

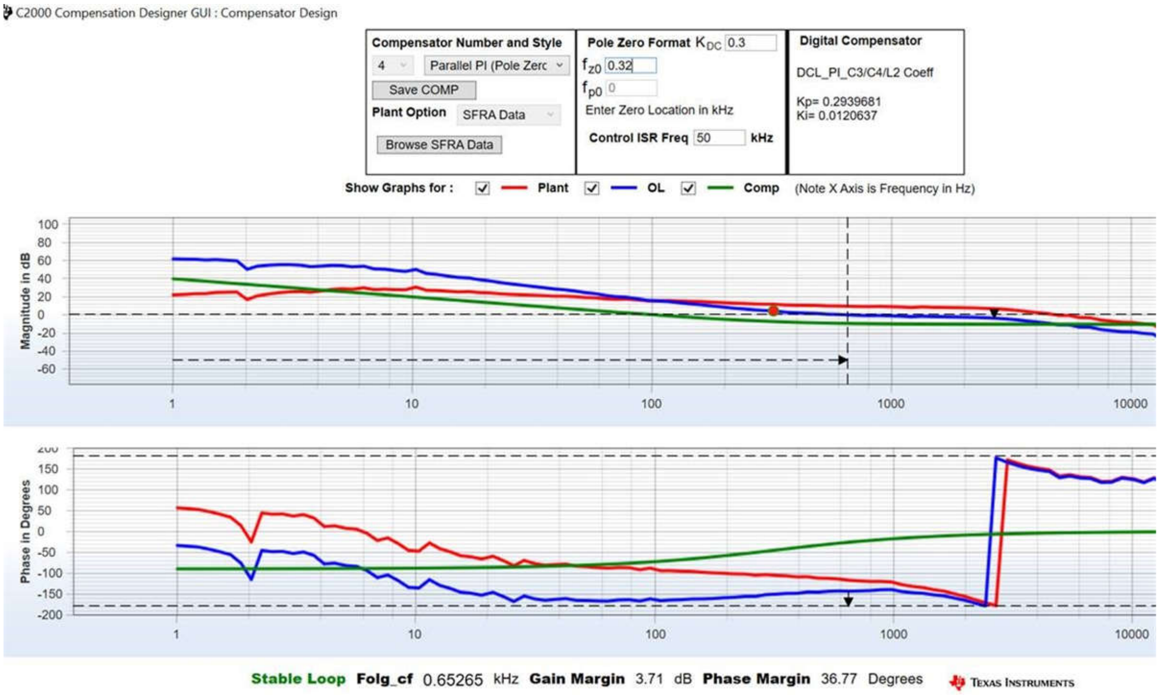
```
#if TINV_LAB == 6
#define TINV_TEST_SETUP TINV_TEST_SETUP_RES_LOAD
#define TINV_PROTECTION TINV_PROTECTION_ENABLED
#define TINV_SFRA_TYPE TINV_SFRA_CURRENT
#define TINV_SFRA_AMPLITUDE (float32_t)TINV_SFRA_INJECTION_AMPLITUDE_LEVEL2
#define TINV_POWERFLOW_MODE TINV_RECTIFIERER_MODE
#define TINV_DC_CHECK 0
#define TINV_SPLL_TYPE TINV_SPLL_SRF
#endif
```

In this check the software is run on the hardware, or the HIL platform, or both.

See the hardware test set up section for actual details of the equipment used for configuring the test. Do not supply any high-voltage power to the board yet.

- First launch the `main.syscfg` and select *Lab 6* in the project options. The compensator style (PI compensator) and the tuning loop (current loop) is automatically populated. Now click the run compensation designer icon and the compensation designer tool launches, with the model of the current loop plant with parameters specified on the powerSUITE page.
- The current compensator coefficients used for running the control loop are shown in the following code. The user can modify these coefficients to meet the necessary loop bandwidth and phase margin. The ideal coefficients with resistive load are slightly different than the one used for grid connection because the grid impedance is very low. The compensator design transfer function and response are shown in Figure 3-23.

```
#define TINV_GV_PI_KP ((float32_t) 1.9979056049)
#define TINV_GV_PI_KI ((float32_t) 0.0041887902)
```



C2000 Compensation Designer GUI : Compensator Design

Compensator Number and Style: 4, Parallel PI (Pole Zerc

Plant Option: SFRA Data

Pole Zero Format K_{DC} 0.3

f_{z0} 0.32

f_{p0} 0

Enter Zero Location in kHz

Control ISR Freq 50 kHz

Digital Compensator

DCL_PI_C3/C4/L2 Coeff

Kp= 0.2939681
Ki= 0.0120637

Show Graphs for : ☑ Plant ☑ OL ☑ Comp (Note X Axis is Frequency in Hz)

Stable Loop Folg_cf 0.65265 kHz Gain Margin 3.71 dB Phase Margin 36.77 Degrees

Figure 3-23. Compensator Design GUI - Current Loop PI Coefficients

- Once satisfied with the proportional and integral gain values, click the *Save COMP* button. This saves the compensator values into the project. Close the Compensation Designer, and return to the powerSUITE page.
- Turn on the auxiliary power supply, set at 12 V. Build and load the code, use the lab6.js file to populate the watch variables in the CCS window.
- Set the e-load CV voltage to 100 V, 20 A limit.
- Set the AC input voltage to 30 V_{RMS} with appropriate current limit.
- Make sure to enable the fans when testing at high power using **TINV_fanSet** function in the CCS watch window during the debug session.
- After turning on the AC power supply, immediately turn on the relay by writing a *1* to `TINV_neutralRelaySet`. Make sure that the relay is turned on immediately (within 2 seconds) after turning on the AC supply.
- Set `TINV_idRef_pu` to –0.015 pu
- To start the PFC mode, enter *"1"* on the `TINV_startPowerStage` variable, the current is now drawn from the grid as a sinusoidal signal (with some harmonics as the current is at low power) and boost the action seen on the vBus. The output voltage boosts from 75 V to around 100 V.
- The current becomes sinusoidal as the load is increased. This verifies start-up of PFC at 30 V_{RMS}.
- Next, redo the PFC tests with 120 V_{RMS} and 230 V_{RMS}inputs.

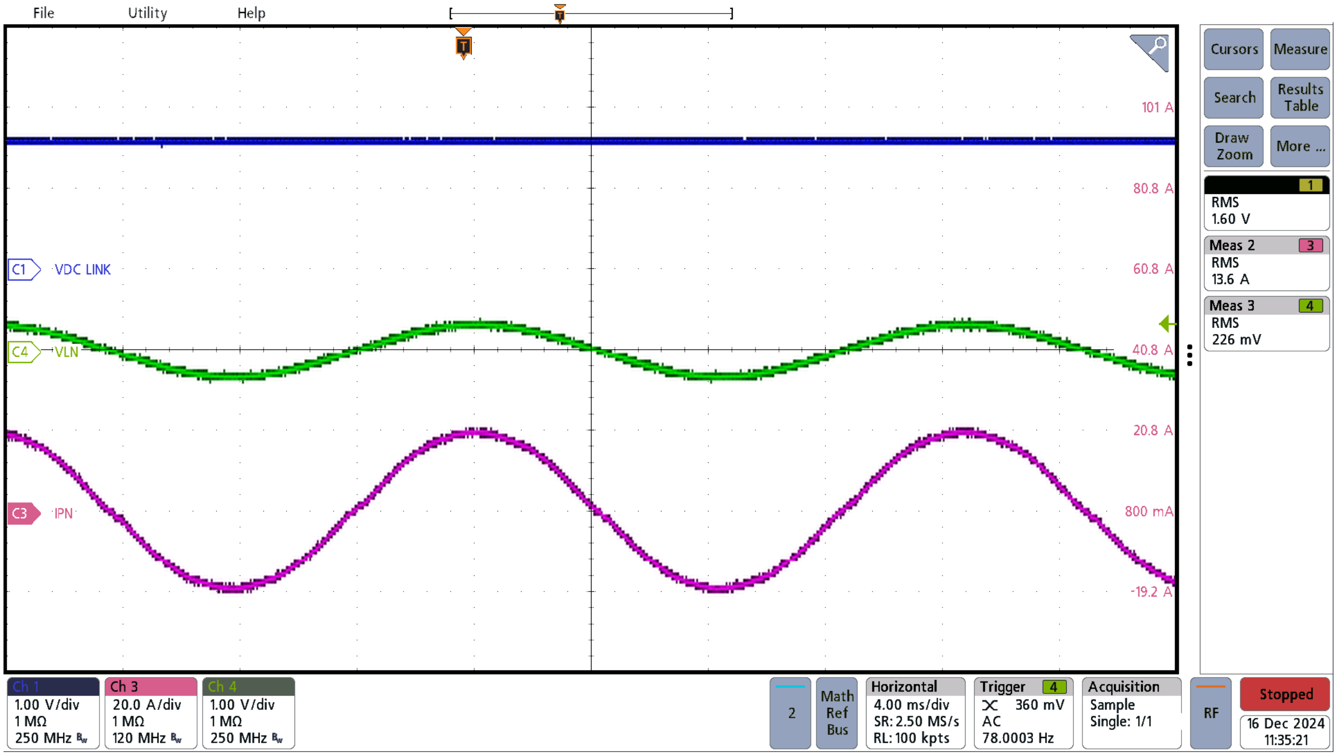
- Set the e-load CV voltage to 800 V, 20 A limit. Start the PFC boost similar to previously done.

Note

Lab 7 introduces a voltage loop for the PFC.

This lab relies on the e-load that regulates the output voltage. `TINV_VBUS_OVERVOLT_LIMIT` must be configured for these tests.

- Figure 3-24 shows an example PFC closed current loop operating at 120 V_{RMS}, 90 kHz PWM.



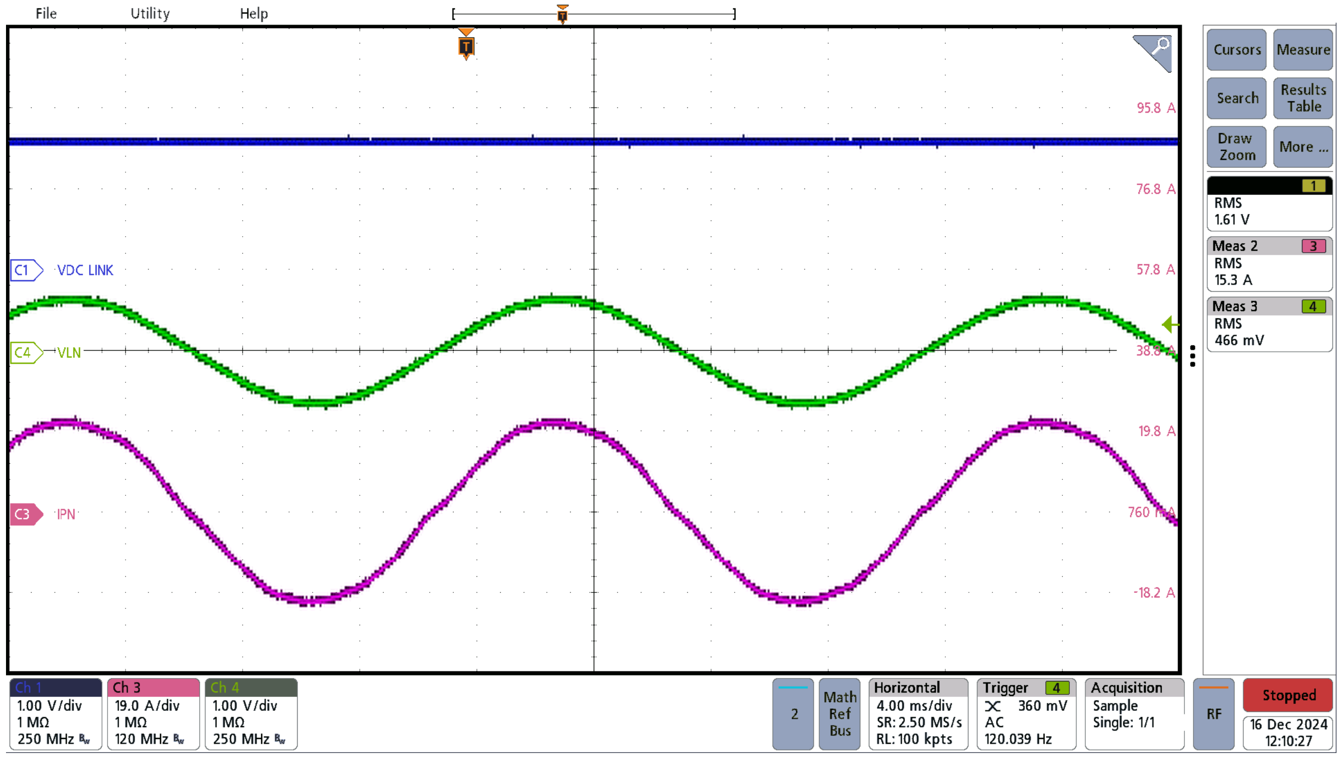
1 RMS 1.60 V
Meas 2 (3) RMS 13.6 A
Meas 3 (4) RMS 226 mV

Ch 1 1.00 V/div 1 MΩ 250 MHz
Ch 3 20.0 A/div 1 MΩ 120 MHz
Ch 4 1.00 V/div 1 MΩ 250 MHz
Horizontal 4.00 ms/div SR: 2.50 MS/s RL: 100 kpts
Trigger (4) 360 mV AC 78.0003 Hz
Acquisition Sample Single: 1/1
Stopped
16 Dec 2024 11:35:21

Scope signals: Channel 1 - DC voltage (blue), Channel 4- AC voltage (green),
Channel 3 - AC current (red). The voltage probes are scaled down at 500:1.

Figure 3-24. PFC Current Loop Operating at 120 V_{RMS}

- Now repeat the previous steps to verify PFC at 220 V_{RMS}.
- Figure 3-25 shows an example PFC closed current loop operating at 230 V_{RMS}, 90 kHz PWM.



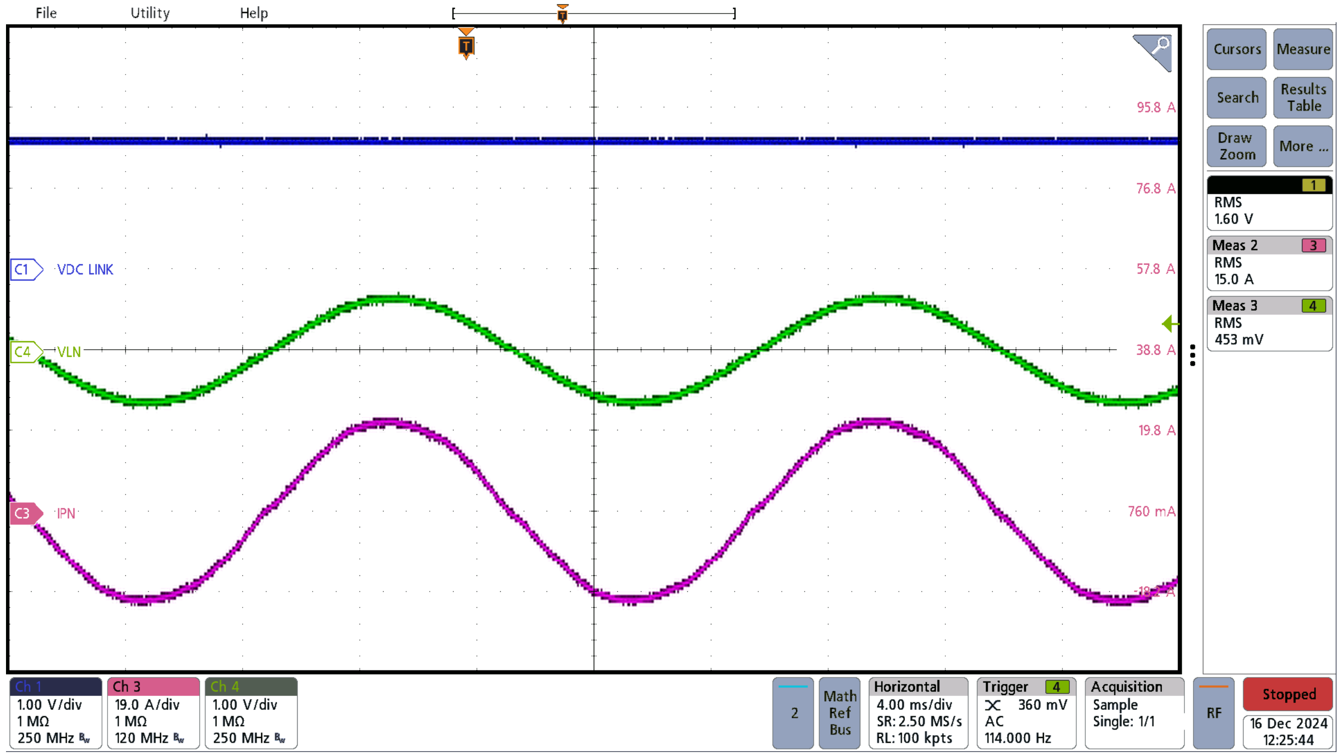
Meas 1: RMS 1.61 V
Meas 2: RMS 15.3 A
Meas 3: RMS 466 mV

Ch 1: 1.00 V/div, 1 MΩ, 250 MHz
Ch 3: 19.0 A/div, 1 MΩ, 120 MHz
Ch 4: 1.00 V/div, 1 MΩ, 250 MHz
Horizontal: 4.00 ms/div, SR: 2.50 MS/s, RL: 100 kpts
Trigger 4: 360 mV, AC, 120.039 Hz
Acquisition: Sample, Single: 1/1
16 Dec 2024 12:10:27

Scope signals: Channel 1 - DC voltage (blue), Channel 4- AC voltage (green), Channel 3 - AC current (red). The voltage probes are scaled down at 500:1.

Figure 3-25. PFC Current Loop Operating at 230 V_{RMS}, 90 kHz

- Figure 3-26 shows an example PFC closed current loop operating at 230 V_{RMS}, 70 kHz PWM.



Meas 1: RMS 1.60 V
Meas 2: RMS 15.0 A
Meas 3: RMS 453 mV

Ch 1: 1.00 V/div, 1 MΩ, 250 MHz
Ch 3: 19.0 A/div, 1 MΩ, 120 MHz
Ch 4: 1.00 V/div, 1 MΩ, 250 MHz
Horizontal: 4.00 ms/div, SR: 2.50 MS/s, RL: 100 kpts
Trigger 4: 360 mV, AC, 114.000 Hz
Acquisition: Sample, Single: 1/1
16 Dec 2024 12:25:44

Scope signals Scope signals: Channel 1 - DC voltage (blue), Channel 4- AC voltage (green), Channel 3 - AC current (red). The voltage probes are scaled down at 500:1.

Figure 3-26. PFC Current Loop Operating at 230 V_{RMS}, 70 kHz

- A soft start scheme is implemented at the start-up to make sure overcurrents are reduced at start-up. For this, the duty is restricted to the PWM module by adjusting the dead band set. Figure 3-27 shows the PWM configuration for this setup where the dead band is set to a large value and slowly reduced to the nominal value to limit the current spikes.

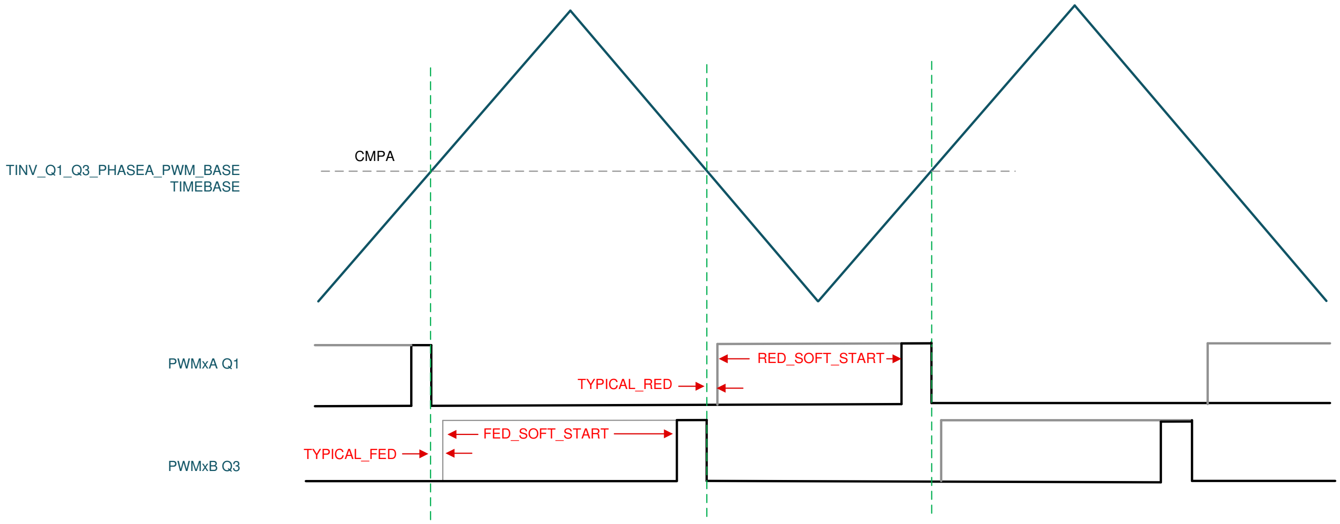
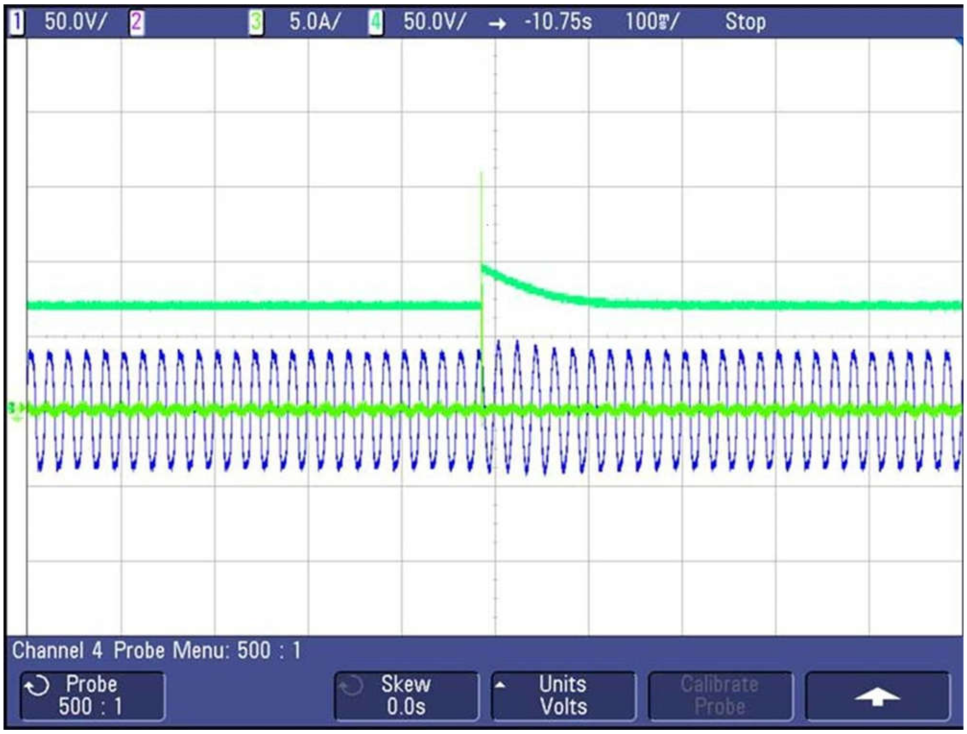


Figure 3-27. Dead Band Soft Start PWM Configuration

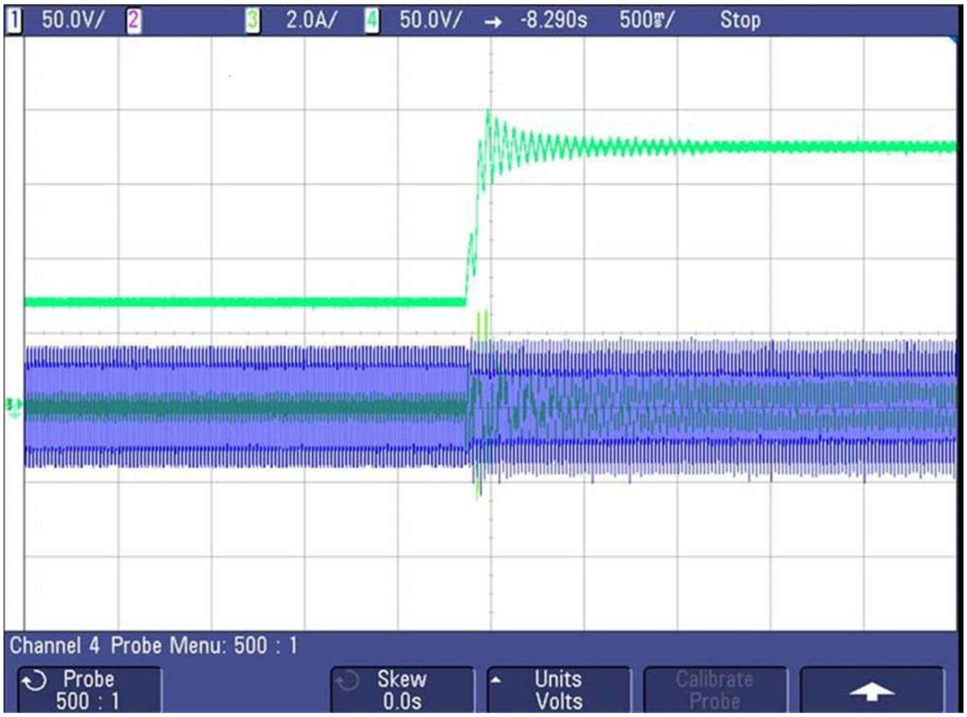
- Figure 3-28 shows the effect of the start-up scheme without dead band implemented at start-up. Without soft start, a huge current spike appears which causes over current trip and the DC bus voltage also collapses. Figure 3-28 shows overcurrent without soft-start implementation.



Scope signals: Channel 1 - AC voltage (blue), Channel 2 - AC current (light green), Channel 3 - DC voltage (dark green)

Figure 3-28. Without Soft Start

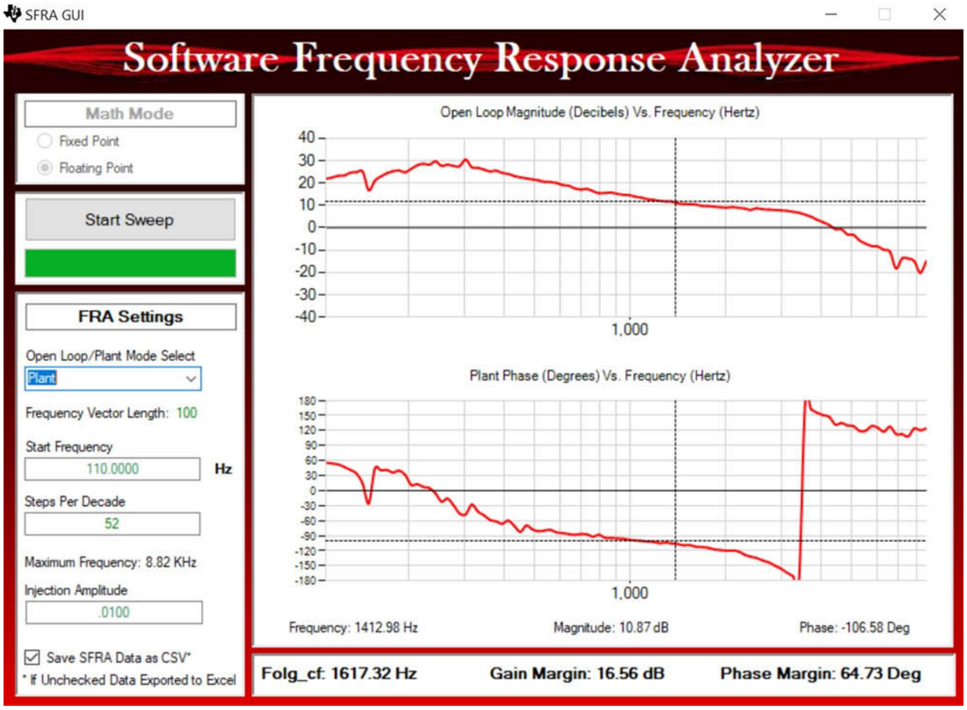
Figure 3-29 shows reduced current spike with soft-start implementation.



Scope signals: Channel 1 - AC voltage (blue), Channel 2 - AC current (light green), Channel 3 - DC voltage (dark green).

Figure 3-29. Soft Start With Adaptive Dead Band

- SFRA is integrated in the software of this lab to verify the designed compensator provides enough gain and phase margin by measuring on hardware. To run the SFRA, keep the project running and from the cfg page, click on the SFRA icon. The SFRA GUI pops up.
- Select the options for the device on the SFRA GUI. For example, for F28377D, select floating point. Click on *Setup Connection*. On the pop-up window uncheck the *boot on connect* option, and select an appropriate COM port. Click the *OK* button. Return to the SFRA GUI, and click *Connect*.
- The SFRA GUI connects to the device. An SFRA sweep can now be started by clicking the *Start Sweep* button. The complete SFRA sweep takes a few minutes to finish. Activity can be monitored by observing the progress bar on the SFRA GUI and also by checking the flashing blue LED on the back of the control card that indicates UART activity. Once complete, a graph with the open loop plot appears, as in Figure 3-30 and Figure 3-31 which corresponds to plant and loop response measured by the SFRA GUI respectively. This verifies that the designed compensator is indeed stable.


SFRA GUI
Software Frequency Response Analyzer
Math Mode
Start Sweep
FRA Settings
Open Loop Magnitude (Decibels) Vs. Frequency (Hertz)
40
30
20
10
0
-10
-20
-30
-40
1,000
Plant Phase (Degrees) Vs. Frequency (Hertz)
180
150
120
90
60
30
0
-30
-60
-90
-120
-150
-180
1,000

○ Fixed Point
● Floating Point

Open Loop/Plant Mode Select: Plant

Frequency Vector Length: 100

Start Frequency: 110.0000 Hz

Steps Per Decade: 52

Maximum Frequency: 8.82 KHz

Injection Amplitude: .0100

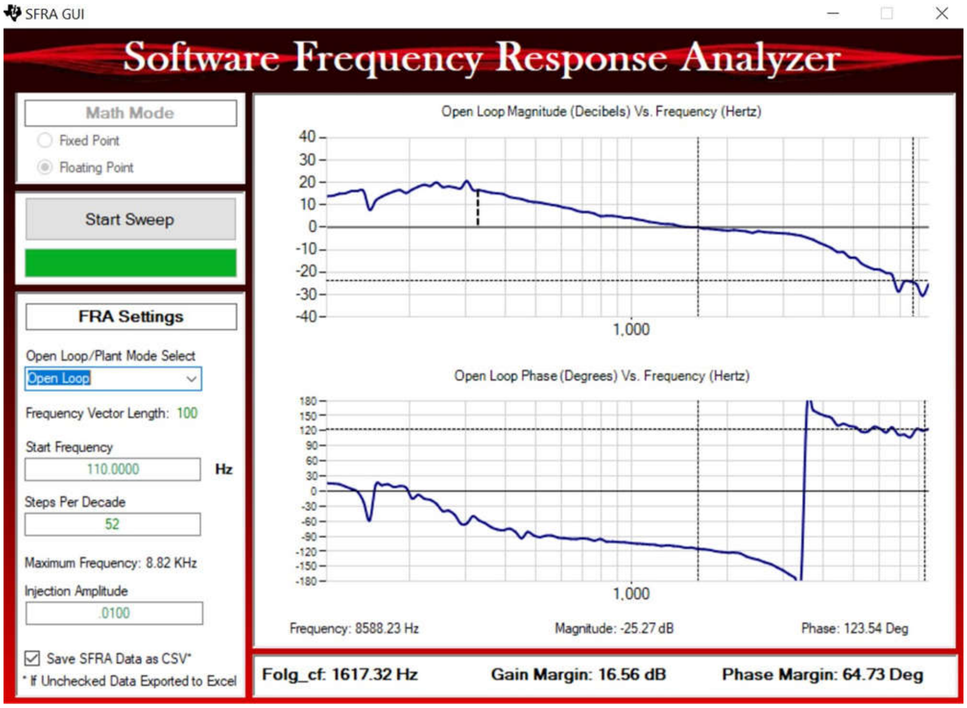
☑ Save SFRA Data as CSV*

* If Unchecked Data Exported to Excel

Frequency: 1412.98 Hz Magnitude: 10.87 dB Phase: -106.58 Deg

Folg_cf: 1617.32 Hz Gain Margin: 16.56 dB Phase Margin: 64.73 Deg

Figure 3-30. PFC SFRA Plant Response for Current Loop


SFRA GUI
Software Frequency Response Analyzer
Math Mode
Start Sweep
FRA Settings
Open Loop Magnitude (Decibels) Vs. Frequency (Hertz)
40
30
20
10
0
-10
-20
-30
-40
1,000
Open Loop Phase (Degrees) Vs. Frequency (Hertz)
180
150
120
90
60
30
0
-30
-60
-90
-120
-150
-180
1,000

○ Fixed Point
● Floating Point

Open Loop/Plant Mode Select: Open Loop

Frequency Vector Length: 100

Start Frequency: 110.0000 Hz

Steps Per Decade: 52

Maximum Frequency: 8.82 KHz

Injection Amplitude: .0100

☑ Save SFRA Data as CSV*

* If Unchecked Data Exported to Excel

Frequency: 8588.23 Hz Magnitude: -25.27 dB Phase: 123.54 Deg

Folg_cf: 1617.32 Hz Gain Margin: 16.56 dB Phase Margin: 64.73 Deg

Figure 3-31. PFC SFRA Loop Response for Current Loop

- The frequency response data is also saved in the project folder under an SFRA data folder and is time stamped with the time of the SFRA run.
- This completes verification of Lab 6.

3.2.3.3 Lab 7

This lab checks the voltage and current loops for the PFC. The variable `TINV_vBusRef_pu` is defined to set the voltage at which the output DC bus voltage is to be regulated.

Figure 3-32 describes the software flow for running Lab 7.

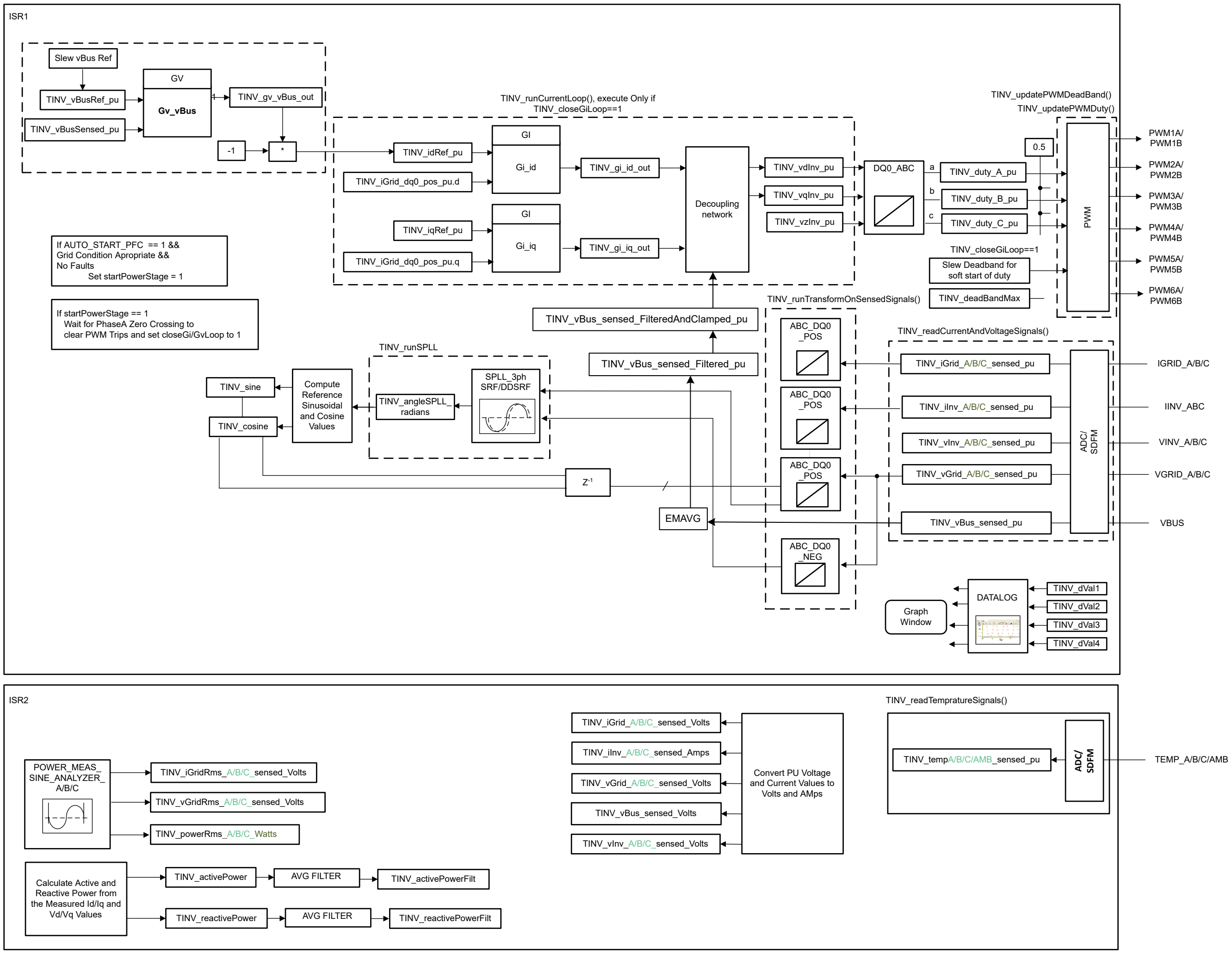


Figure 3-32. Lab 7 Software Diagram

Set the project to Lab 7 by changing the *lab number* in the `<tinv_settings.h>` or `main.syscfg` file, (this is changed by powerSUITE GUI when using the powerSUITE project).

In the user `settings.h` file some additional options are available, but the following code is used for the tests documented in this user guide. These settings can be different than the SDK default settings. Use the verified settings here:

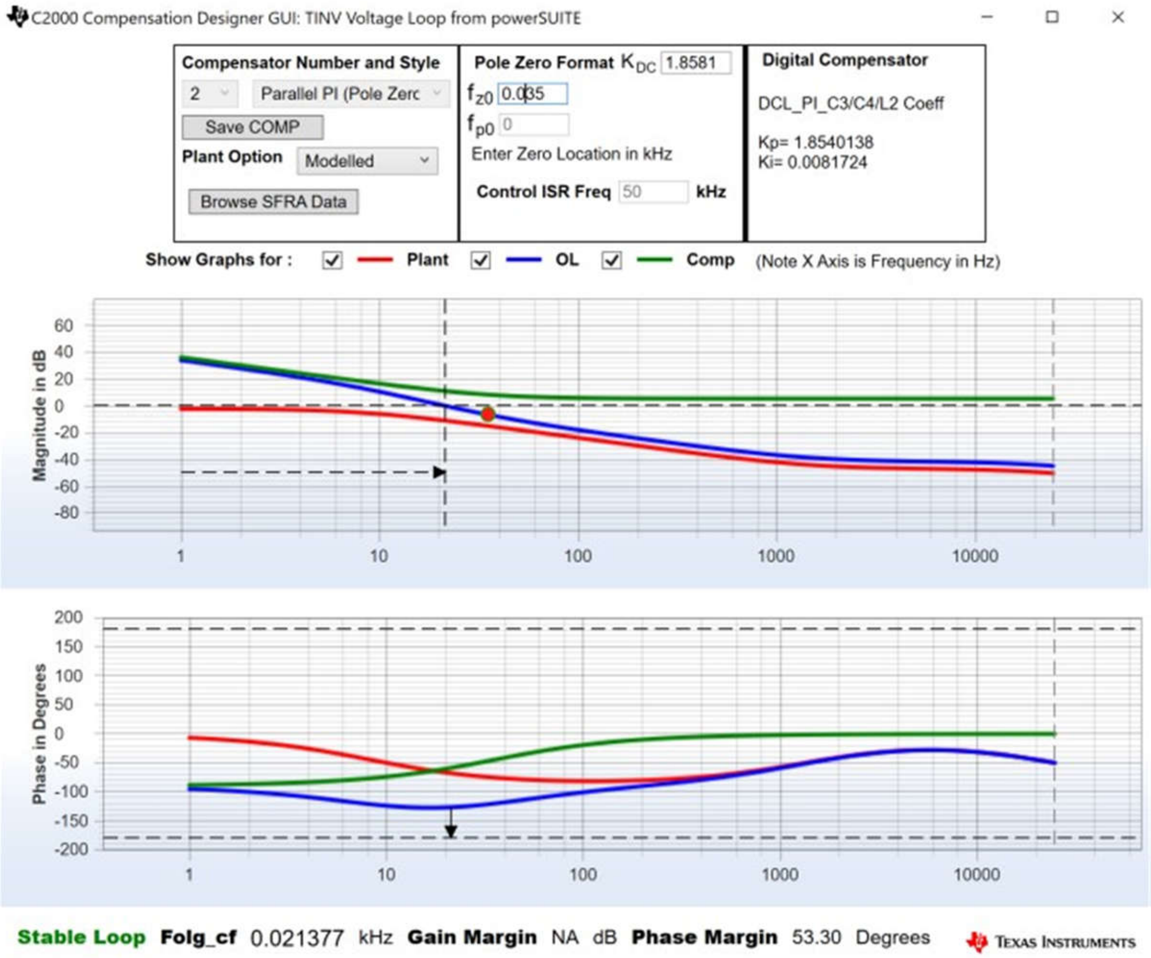
```
#if TINV_LAB == 7
#define TINV_TEST_SETUP TINV_TEST_SETUP_RES_LOAD
#define TINV_PROTECTION TINV_PROTECTION_ENABLED
#define TINV_SFRA_TYPE TINV_SFRA_CURRENT
#define TINV_SFRA_AMPLITUDE (float32_t)TINV_SFRA_INJECTION_AMPLITUDE_LEVEL2
#define TINV_POWERFLOW_MODE TINV_RECTIFIERER_MODE
#define TINV_DC_CHECK 0
#define TINV_SPLL_TYPE TINV_SPLL_SRF
#endif
```

In this check the software is run on the hardware, or the HIL platform, or both.

See the hardware test set up section for actual details of the equipment used for configuring the test. At this time, do not supply any high-voltage power to the board.

- First launch the `main.cfg` and select Lab 7 in the project options. The compensator style (PI compensator) and the tuning loop (current loop) are automatically populated. Now click the *run compensation designer* icon and the compensation designer tool launches, with the model of the current loop plant with parameters specified on the powerSUITE page.
- Figure 3-33 shows the current compensator coefficients used for running the control loop. The user can modify these coefficients to meet the necessary loop bandwidth and phase margin. The ideal coefficients with resistive load are slightly different than the one used for grid connection because the grid impedance is very low. Figure 3-33 shows the compensator design transfer function and response.

```
#define TINV_GI_PI_KP ((float32_t)1.8540138247))
#define TINV_GI_PI_KI ((float32_t)0.0081723506))
```



C2000 Compensation Designer GUI: TINV Voltage Loop from powerSUITE

Compensator Number and Style: 2, Parallel PI (Pole Zerc

Save COMP

Plant Option: Modelled

Browse SFRA Data

Pole Zero Format K_{DC} 1.8581

f_{z0} 0.035

f_{p0} 0

Enter Zero Location in kHz

Control ISR Freq 50 kHz

Digital Compensator

DCL_PI_C3/C4/L2 Coeff

Kp= 1.8540138
Ki= 0.0081724

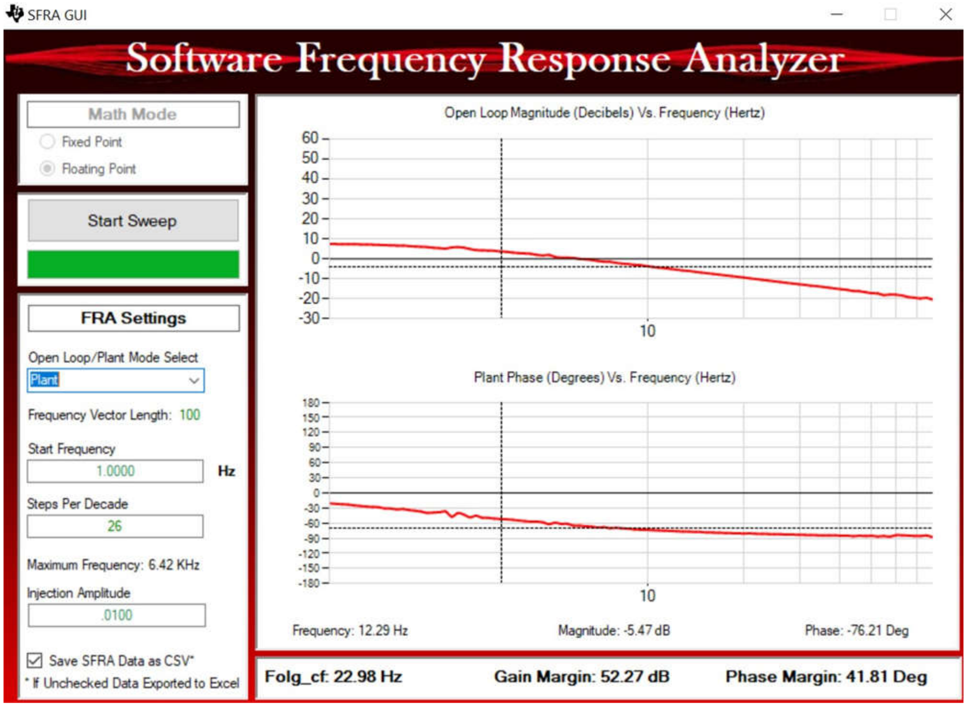
Show Graphs for : ☑ Plant ☑ OL ☑ Comp (Note X Axis is Frequency in Hz)

Stable Loop Folg_cf 0.021377 kHz Gain Margin NA dB Phase Margin 53.30 Degrees

Figure 3-33. Compensator Design GUI - Voltage Loop PI Coefficients

- Once satisfied with the proportional and integral gain values, click the *Save COMP* button. This saves the compensator values into the project. Close the *Compensation Designer*, and return to the powerSUITE page.
- Turn on the auxiliary power supply, set at 12 V. Build and load the code, use the `lab7.js` file to populate the watch variables in the CCS window.
- Set the e-load resistance to a high value around 3 kΩ or in CC mode of 0.25 A.
- Set the AC input voltage to 230 V_{RMS} with an appropriate current limit.
- After turning on the AC power supply, immediately turn on the relay by writing a *1* to `TINV_neutralRelaySet`. Make sure that the relay is turned on immediately (within 2 seconds) after turning on the AC supply.
- Now set `TINV_vBusRef_pu` to 0.727 pu. This corresponds to bus voltage of 800 V.

- Make sure to enable the fans when testing at high power using the `TINV_fanSet` function in the CCS watch window during the debug session.
- To start the PFC mode, enter *"1"* on the `TINV_startPowerStage` variable, the current is now drawn from the grid as a sinusoidal signal (with some harmonics as the current is at low power) and boost the action seen on the vBus. The output voltage boosts from 550 V to around 800 V drawing around 200 W power from AC supply as shown in Figure 3-34. This transition happens in around 150 ms.
- The current becomes sinusoidal as the load is increased. This verifies start-up of PFC at 230 V_{RMS} and is shown in Figure 3-34.
- In case any overcurrent trip is observed which causes the PWMs to switch off, see the notes in Lab 5 to debug this condition.
- The converter efficiency results and transient tests are shown in the Test Results section.
- SFRA is integrated in the software of this lab to verify the designed compensator provides enough gain and phase margin by measuring on hardware. To run the SFRA keep the project running, and from the `.cfg` page, click on the SFRA icon. The SFRA GUI pops up.
- Select the options for the device on the SFRA GUI. For example, for F28379D select floating point. Click on *Setup Connection*. On the pop-up window uncheck the *boot on connect* option, and select an appropriate COM port. Click the *OK* button. Return to the SFRA GUI, and click *Connect*.
- The SFRA GUI connects to the device. An SFRA sweep can now be started by clicking the *Start Sweep* button. The complete SFRA sweep takes a few minutes to finish. Activity can be monitored by observing the progress bar on the SFRA GUI and also checking the flashing blue LED on the back of the control card that indicates UART activity. Once complete, a graph with the open loop plot appears, as in Figure 3-34. Figure 3-34 shows measured plant response by SFRA GUI and Figure 3-35 shows measured loop response by SFRA GUI. This verifies that the designed compensator is indeed stable.


SFRA GUI
Software Frequency Response Analyzer
Math Mode
Start Sweep
FRA Settings
Open Loop Magnitude (Decibels) Vs. Frequency (Hertz)
60
50
40
30
20
10
0
-10
-20
-30
10
Plant Phase (Degrees) Vs. Frequency (Hertz)
180
150
120
90
60
30
0
-30
-60
-90
-120
-150
-180
10

○ Fixed Point
● Floating Point

Open Loop/Plant Mode Select: Plant

Frequency Vector Length: 100

Start Frequency: 1.0000 Hz

Steps Per Decade: 26

Maximum Frequency: 6.42 KHz

Injection Amplitude: .0100

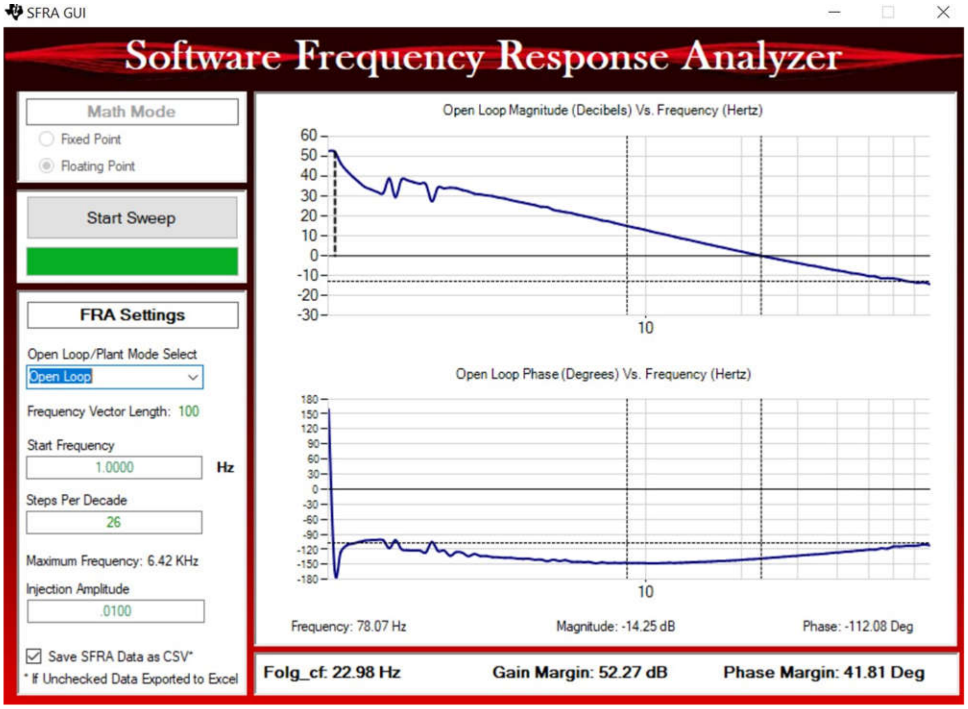
☑ Save SFRA Data as CSV*

* If Unchecked Data Exported to Excel

Frequency: 12.29 Hz Magnitude: -5.47 dB Phase: -76.21 Deg

Folg_cf: 22.98 Hz Gain Margin: 52.27 dB Phase Margin: 41.81 Deg

Figure 3-34. PFC SFRA Plant Response for Voltage Loop


SFRA GUI
Software Frequency Response Analyzer
Math Mode
Start Sweep
FRA Settings
Open Loop Magnitude (Decibels) Vs. Frequency (Hertz)
60
50
40
30
20
10
0
-10
-20
-30
10
Open Loop Phase (Degrees) Vs. Frequency (Hertz)
180
150
120
90
60
30
0
-30
-60
-90
-120
-150
-180
10

○ Fixed Point

● Floating Point

Open Loop/Plant Mode Select: Open Loop

Frequency Vector Length: 100

Start Frequency: 1.0000 Hz

Steps Per Decade: 26

Maximum Frequency: 6.42 KHz

Injection Amplitude: .0100

☑ Save SFRA Data as CSV*

* If Unchecked Data Exported to Excel

Frequency: 78.07 Hz Magnitude: -14.25 dB Phase: -112.08 Deg

Folg_cf: 22.98 Hz Gain Margin: 52.27 dB Phase Margin: 41.81 Deg

Figure 3-35. PFC SFRA Loop Response for Voltage Loop

- The frequency response data is also saved in the project folder under an SFRA data folder and is time stamped with the time of the SFRA run. Also, note the measured gain and phase margin are close to the modeled values as shown in the voltage loop compensator design as previously elaborated.
- This action verifies the voltage loop compensator design. To bring the system to a safe stop, bring the input AC voltage down to zero.

3.2.4 Test Setup for Efficiency

Figure 3-36 illustrates the equipment required to test the efficiency of this reference design.

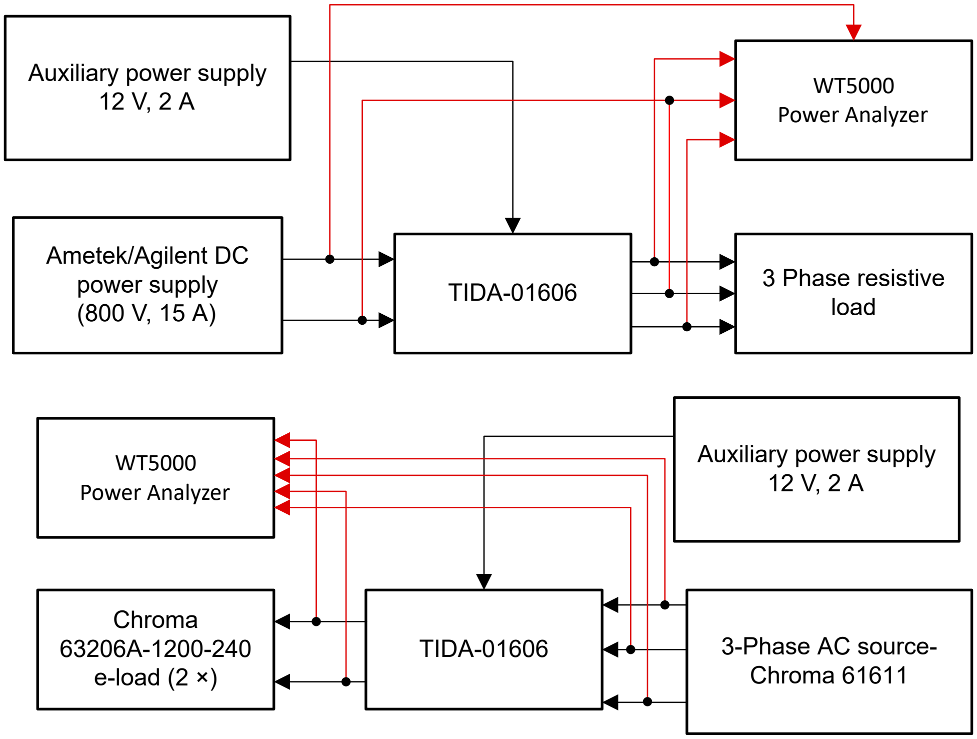


Figure 3-36. Test Setup for Efficiency

- Chroma 616xx series three-phase AC power supply to test PFC mode and an AMETEK, Agilent DC power supply of 800 V to support inverter mode (TIDA-01606) of operation
- 110-kW Simplex PowerStar (or any three-phase resistive) load bank is used as a configurable load to test the design at various set points
- Tektronix WT5000 Precision Power Analyzer
- External BK precision bench power supply is used to provide a 12-V input to power the DUT

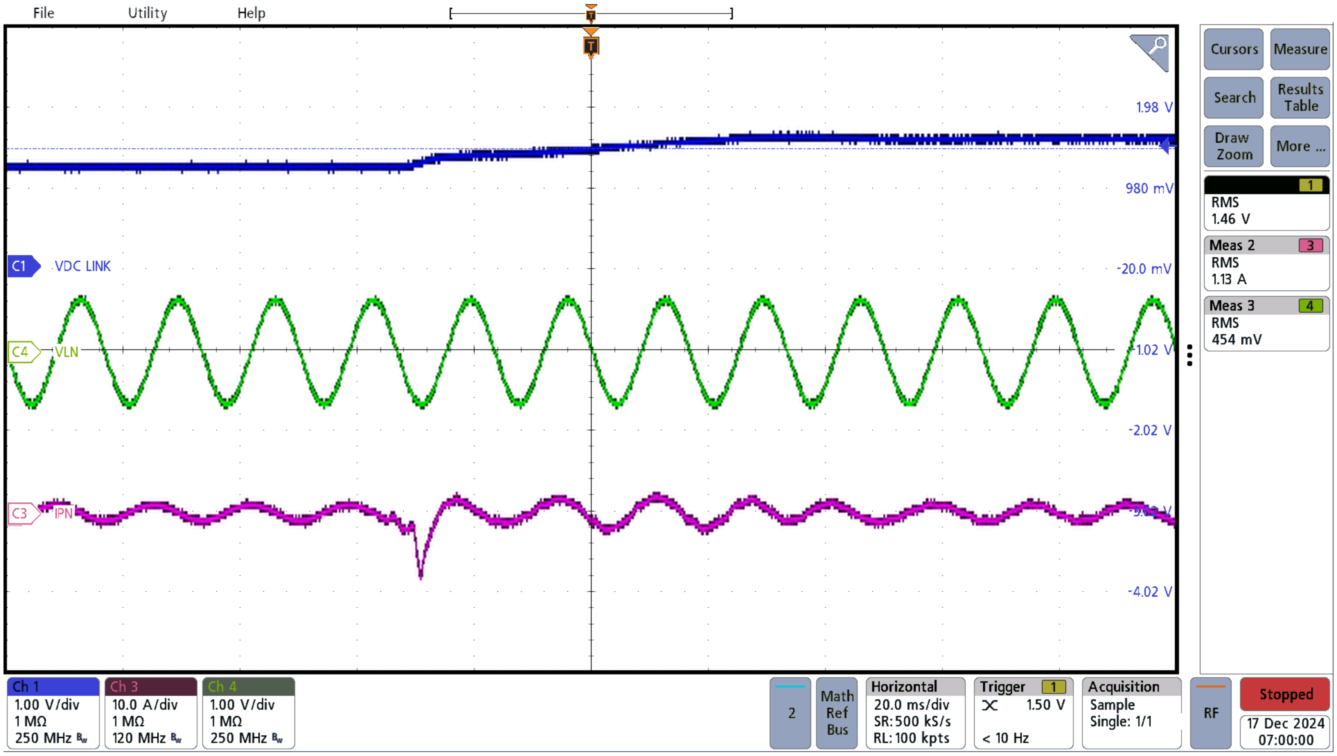
3.2.5 Test Results

The following sections cover the results for the inverter mode and PFC mode of operation.

3.2.5.1 PFC Mode

3.2.5.1.1 PFC Start-Up – 230 V_{RMS}, 400 V_{L-L} AC Voltage

Figure 3-37 shows the start-up sequence of the power stage with an input of three phase 400 V_{L-L} and output bus regulated at 800 V, and around 750-W load. The boost action from 550 V to 800 V happens in around 70 ms. Figure 3-37 shows the PFC start-up performance.



1 RMS 1.46 V
Meas 2 3 RMS 1.13 A
Meas 3 4 RMS 454 mV

Ch 1 1.00 V/div 1 MΩ 250 MHz
Ch 3 10.0 A/div 1 MΩ 120 MHz
Ch 4 1.00 V/div 1 MΩ 250 MHz
Horizontal 20.0 ms/div SR: 500 kS/s RL: 100 kpts
Trigger 1 1.50 V < 10 Hz
Acquisition Sample Single: 1/1
Stopped
17 Dec 2024 07:00:00

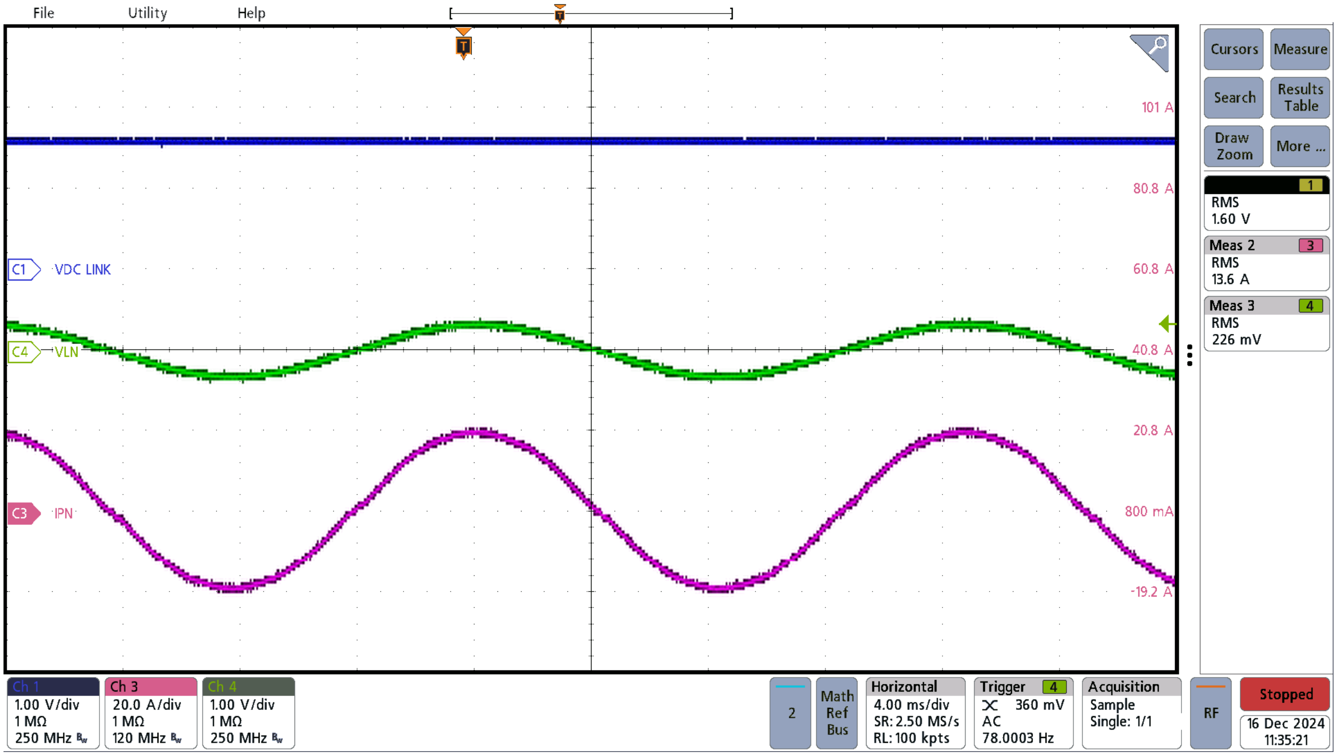
Scope signals: Channel 1 - VDC at output (blue), Channel 4 - AC input phase voltage VLN (green), Channel 3 - AC input current (red). The voltage probes are scaled down at 500:1.

Figure 3-37. PFC 230 V_{RMS} Start-Up

Care was taken when tuning the voltage loop to make sure that there is no overvoltage (above 800 V) during the ramp up of the DC link voltage from 550 V to 800 V at start-up. Starting the PFC at a high load can lead to overcurrent events which, in turn, can trip the PWMs. Hence, start the PFC in the previously mentioned test condition or lower output power. See also *Lab 5* on possible causes for PWM trip and possible precautions the user needs to take to avoid this condition.

3.2.5.1.2 Steady State Results - PFC Mode

Figure 3-38 shows the steady state performance of PFC at 120 V_{RMS} and 4.5kW output power.



Meas 1: RMS 1.60 V
Meas 2: RMS 13.6 A
Meas 3: RMS 226 mV

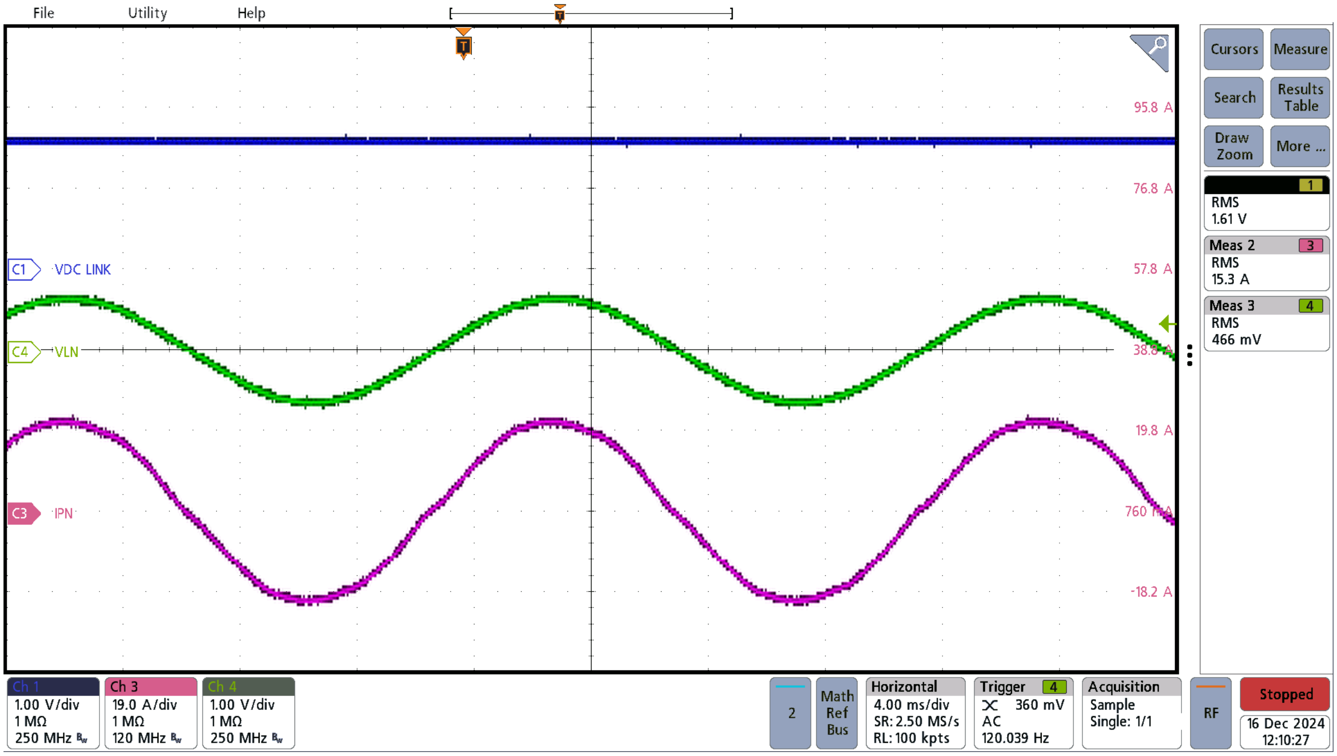
Ch 1: 1.00 V/div, 1 MΩ, 250 MHz
Ch 3: 20.0 A/div, 1 MΩ, 120 MHz
Ch 4: 1.00 V/div, 1 MΩ, 250 MHz

Horizontal: 4.00 ms/div, SR: 2.50 MS/s, RL: 100 kpts
Trigger: 4, 360 mV, AC, 78.0003 Hz
Acquisition: Sample, Single: 1/1
16 Dec 2024 11:35:21

Scope signals: Channel 1 - VDC at output (blue), Channel 4 - AC input phase voltage VLN (green), Channel 3 - AC input current (red). The voltage probes are scaled down at 500:1.

Figure 3-38. 120 V_{RMS} PFC Results at Steady State - 4.5 kW

Figure 3-39 shows the steady state performance of PFC at 230 V_{RMS} input and 4.5-kW output power.



Scope signals: Channel 1 - VDC at output (blue), Channel 4 - AC input phase voltage VLN (green), Channel 3 - AC input current (red). The voltage probes are scaled down at 500:1.

Figure 3-39. 230 V_{RMS} PFC Waveforms at Steady State - 4.5 kW

3.2.5.1.3 Efficiency, THD, and Power Factor Results, 60 Hz – PFC Mode

This section covers the efficiency, ITHD (Current Total Harmonic Distortion), and power factor results for the converter operating in PFC mode at 120 V_{RMS} and 230 V_{RMS}. For the 120 V_{RMS} input, tests were done at 90 kHz PWM for both 400 V and 800 V outputs. For the 230 V_{RMS} input, tests were done at 800 V output for both 70 kHz and 90 kHz PWMs.

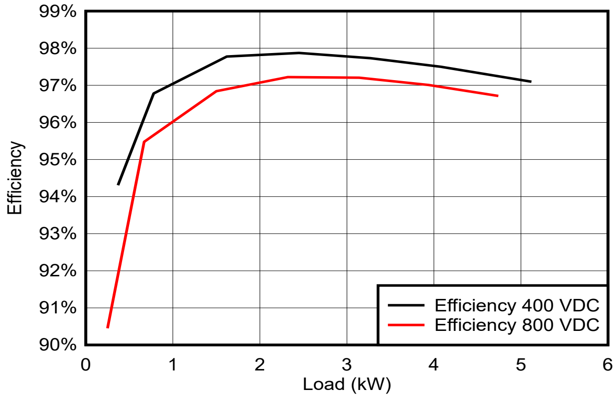


Figure 3-40. Efficiency Results - PFC Mode at 120 V_{RMS}, 90 kHz

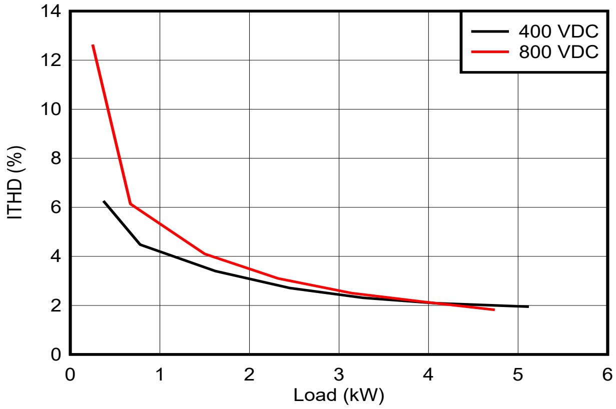


Figure 3-41. THD Results - PFC Mode at 120 V_{RMS}, 90 kHz

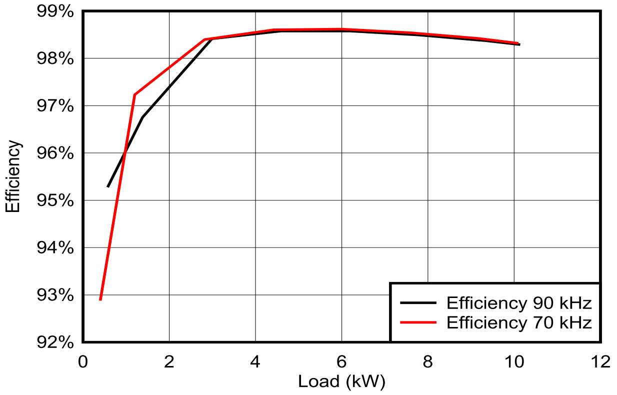


Figure 3-42. Efficiency Results - PFC Mode at 230 V_{RMS}, 800 VDC

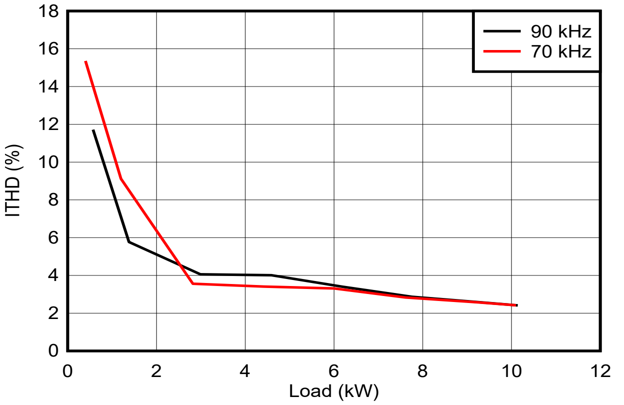


Figure 3-43. THD Results - PFC Mode at 230 V_{RMS}, 800 VDC

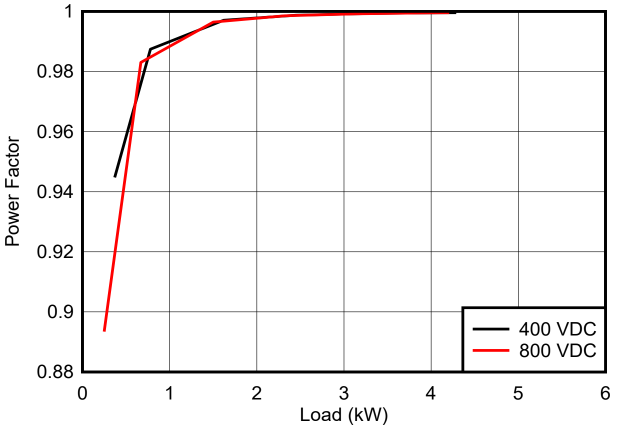


Figure 3-44. Power Factor Results - PFC Mode at 120 V_{RMS}, 90 kHz

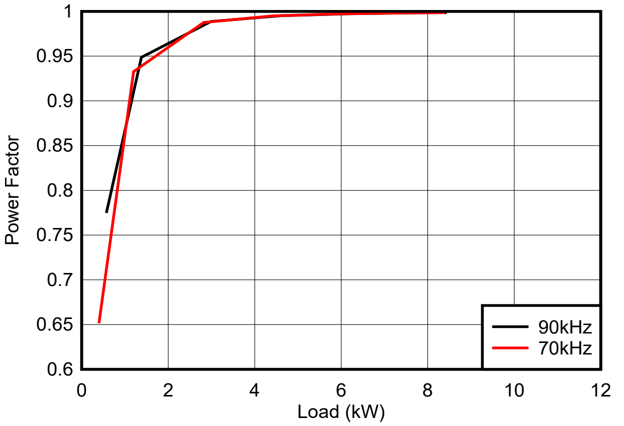
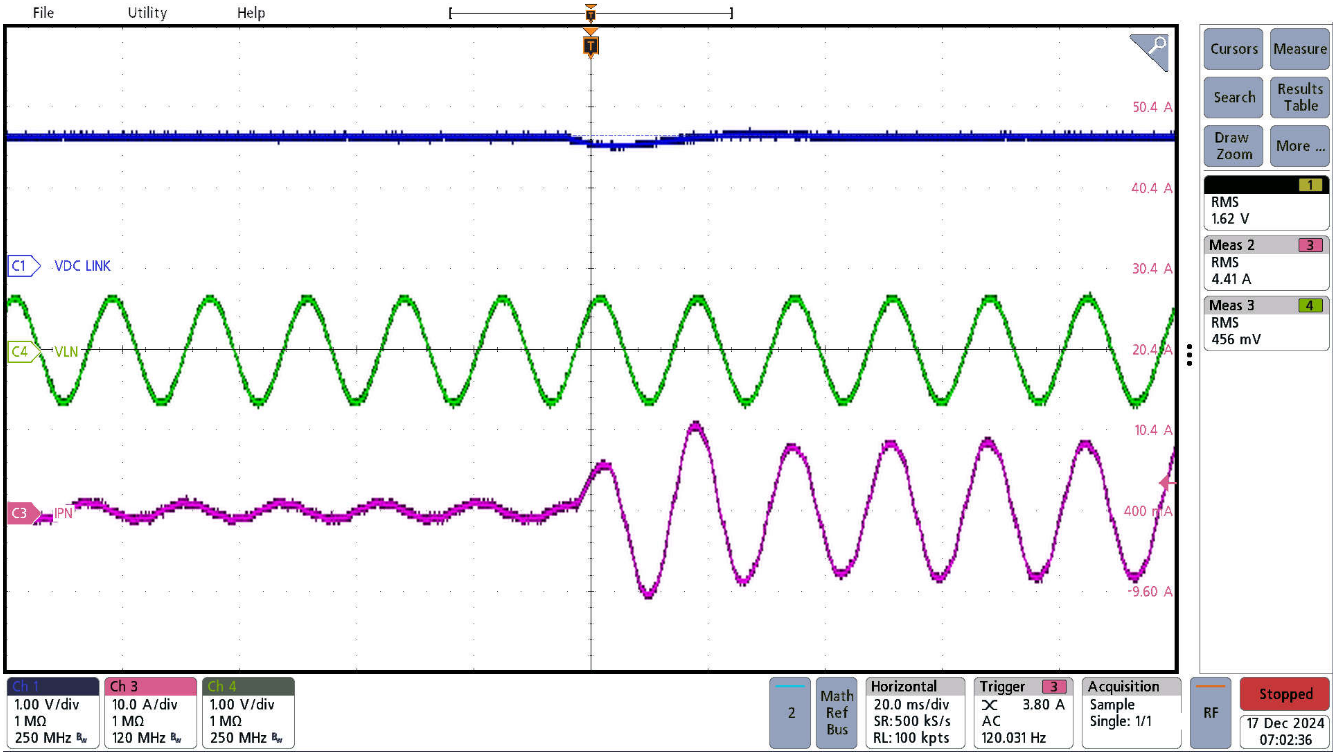


Figure 3-45. Power Factor Results - PFC Mode at 230 V_{RMS}, 800 VDC

3.2.5.1.4 Transient Test With Step Load Change

In this section, the converter is initially operating at 800 V DC link voltage at 230 V_{RMS} AC input voltage and delivering around 0 A. The converter is then subjected to a 5-A step load (4 kW). The next load step is from 5 A to 0 A. The results are shown in Figure 3-46 and Figure 3-47. The peak voltage ripple measured on the DC Link was around 10% of steady state.



1 RMS 1.62 V

Meas 2 3 RMS 4.41 A

Meas 3 4 RMS 456 mV

Ch 1 1.00 V/div 1 MΩ 250 MHz

Ch 3 10.0 A/div 1 MΩ 120 MHz

Ch 4 1.00 V/div 1 MΩ 250 MHz

Horizontal 20.0 ms/div SR:500 kS/s RL:100 kpts

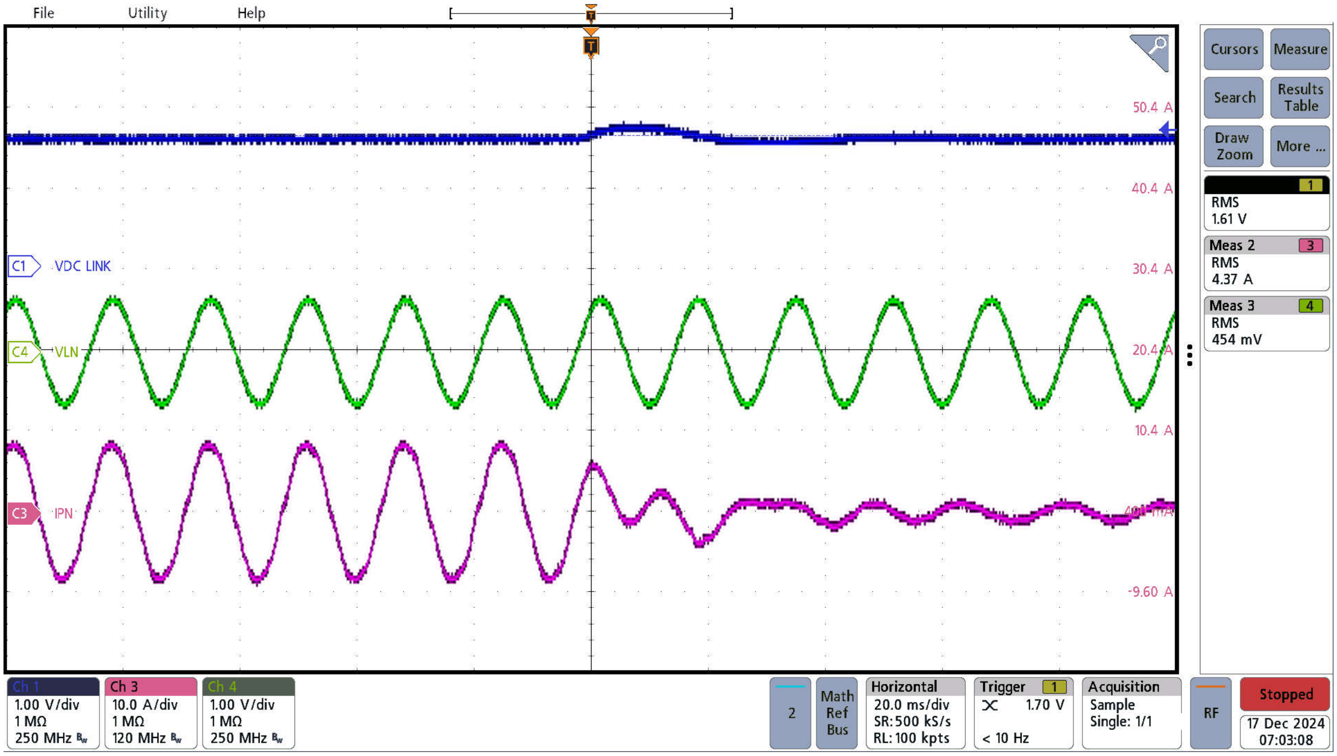
Trigger 3 3.80 A AC 120.031 Hz

Acquisition Sample Single: 1/1

17 Dec 2024 07:02:36

Scope signals: Channel 1 - VDC at output (blue), Channel 4 - AC input phase voltage VLN (green), Channel 3 - AC input current (red). The voltage probes are scaled down at 500:1.

Figure 3-46. Step-Up Load Transient – 0 W → 4 kW



1 RMS 1.61 V

Meas 2 3 RMS 4.37 A

Meas 3 4 RMS 454 mV

Ch 1 1.00 V/div 1 MΩ 250 MHz

Ch 3 10.0 A/div 1 MΩ 120 MHz

Ch 4 1.00 V/div 1 MΩ 250 MHz

Horizontal 20.0 ms/div SR:500 kS/s RL:100 kpts

Trigger 1 1.70 V < 10 Hz

Acquisition Sample Single: 1/1

17 Dec 2024 07:03:08

Scope signals: Channel 1 - VDC at output (blue), Channel 4 - AC input phase voltage VLN (green), Channel 3 - AC input current (red). The voltage probes are scaled down at 500:1.

Figure 3-47. Step Down Load Transient - 4 kW → 0 W

3.2.5.2 Inverter Mode

Preliminary closed-loop inverter mode tests at 800 V, 940 W show an efficiency around 97.5%. Figure 3-48 and Figure 3-49 show the thermal images of the inverter.

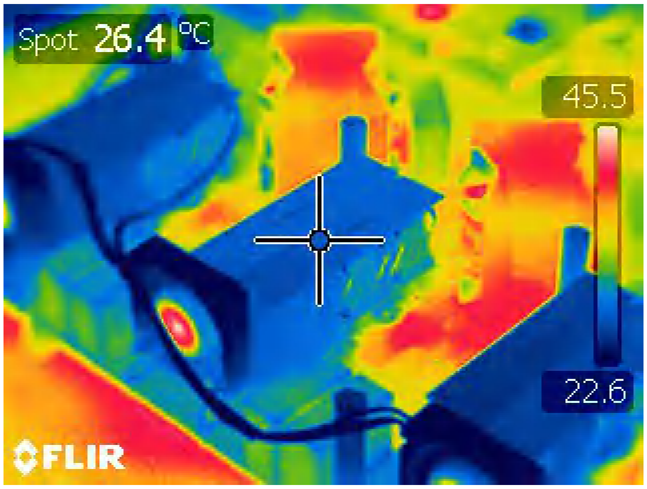


Figure 3-48. Thermal Image of the Bottom Side of the Board, Inverter Running at 900 W

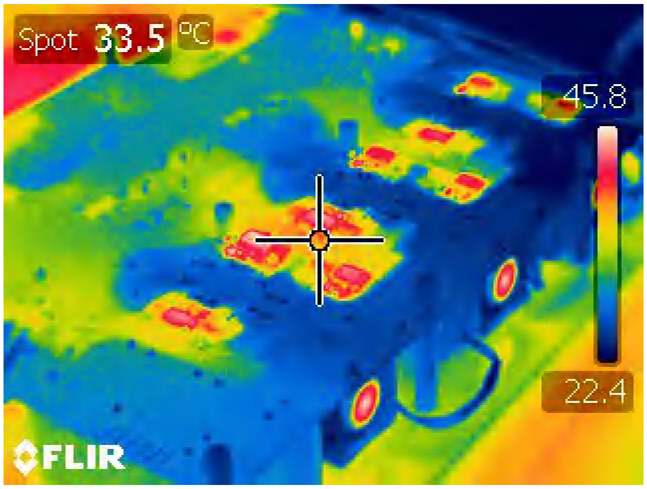


Figure 3-49. Thermal Image of the Top Side of the Board, Inverter Running at 900 W

4 Design Files

4.1 Schematics

To download the schematics, see the design files at TIDA-01606.

4.2 Bill of Materials

To download the bill of materials (BOM), see the design files at TIDA-01606.

4.3 PCB Layout Recommendations

4.3.1 Layout Prints

To download the layer plots, see the design files at TIDA-01606.

4.4 Altium Project

To download the Altium Designer® project files, see the design files at TIDA-01606.

4.5 Gerber Files

To download the Gerber files, see the design files at TIDA-01606.

4.6 Assembly Drawings

To download the assembly drawings, see the design files at TIDA-01606.

5 Trademarks

TI E2E™, Delfino™, TMS320C2000™, and C2000™ are trademarks of Texas Instruments.
Microsoft® and Windows® are registered trademarks of Microsoft Corporation.
Altium Designer® is a registered trademark of Altium LLC or its affiliated companies.
All trademarks are the property of their respective owners.

6 About the Authors

RUFFO RICCARDO received the Ph.D. degree in Electric, Electronics and Communication Engineering from Politecnico di Torino, Turin, Italy, in 2019. He is currently working at Texas Instruments Germany as System Engineer in the area of Grid Infrastructure, Renewable Energy. His main work includes EV charging, inductive wireless power transfer, photovoltaic, renewable energy, and energy storage applications.

KELVIN LE is a systems engineer at Texas Instruments, where he is responsible for developing system designs for the Grid sector with a focus on EV Charging. Kelvin has been with TI since 2015. Kelvin earned his Bachelor of Science in Biomedical Engineering from the University of Central Oklahoma and his Master of Science in Electrical and Computer Engineering from the University of Texas at Austin.

The authors want to thank **MANISH BHARDWAJ**, **BART BASILE**, **HARISH RAMAKRISHNAN**, and **MURALI KRISHNA PACHIPULUSU** for the support given for this reference design.

7 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Revision I (July 2023) to Revision J (February 2025)	Page
• Updates made throughout the document for hardware revision E7	1

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265