参考資料

*デザイン・ガイド: **TIDM-1000***

C2000™ MCU使用の、Vienna整流器をベースとした三相力率補正(PFC)のリファレンス・デザイン

概要

Vienna整流器電源トポロジは、オフボードの電気自動車(EV)充電器やテレコム整流器など、大電力の三相力率補正アプリケーションに使用されます。整流器の制御設計は複雑になる可能性があります。このデザイン・ガイドでは、C2000™マイクロコントローラ (MCU) を使用して電力段を制御する方法について解説します。また、HTTP GUI ページとイーサネットのサポートにより、Vienna 整流器を監視・制御することもできます (F2838x のみ)。このデザインで利用できるハードウェアとソフトウェアは、開発期間の短縮に役立ちます。

リソース

TIDM-1000	デザイン・フォルダ
TMS320F28379D	プロダクト・フォルダ
TMS320F280049C	プロダクト・フォルダ
TMS320F28388D	プロダクト・フォルダ
UCC21520DW	プロダクト・フォルダ
AMC1301	プロダクト・フォルダ

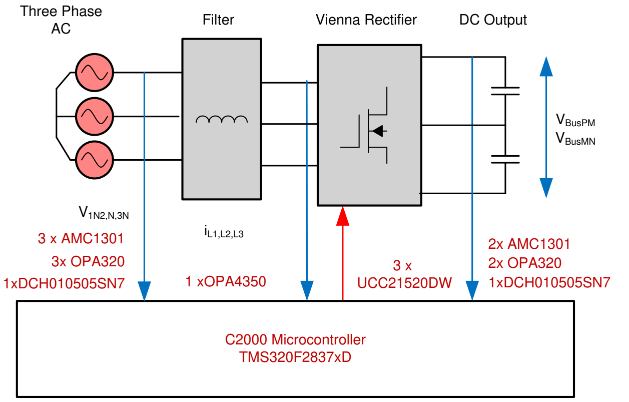
Search Our E2E™ support forums

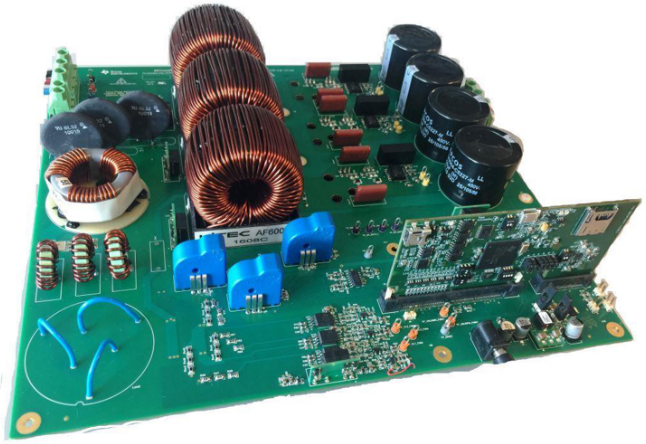
特長

- 三相入力208VL-L 60Hz、出力600V DC (公称値)、1.2KW
- 三相入力400VL-L 50Hz、出力700V DC (公称値)、2.4KW
- 50kHzのパルス幅変調(PWM)スイッチング
- 98%を上回るピーク効率
- 全負荷および低ライン時に2%未満の全高調波歪み(THD)
- powerSUITEのサポートにより、ユーザー要件に合わせて設計を簡単に適合可能
- ソフトウェア周波数応答アナライザ(SFRA)および補償デザイナーにより、制御ループを簡単にチューニング
- ドライバ・ライブラリを使用して F2838x、F2837x、F28004x のソフトウェアをサポート。制御ループは、C28x と CLA のどちらでも同じソースコードで実行可能
- HTTP GUI ページとイーサネットのサポートにより、Vienna 整流器を監視・制御 (F2838x のみ)

アプリケーション

- EVのオフボード充電器
- テレコム整流器
- ドライブ、溶接、その他産業用途





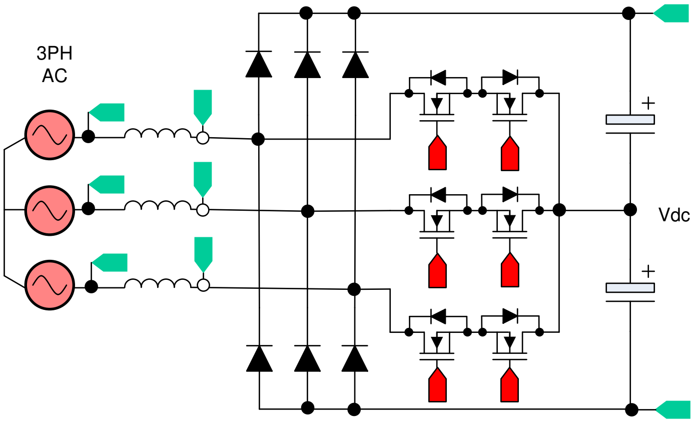
TIDUCJ0 翻訳版 — 最新の英語版資料 http://www-s.ti.com/sc/techlit/TIDUCJ0

1 System Description

Three-phase power is used by equipment operating at high power in industrial applications. To improve grid power quality and reduce the harmonic currents drawn, power factor correction is needed as many of the forward loads are DC. For example, in an offboard, fast EV charger, operating at 20 KW, the input is a three-phase AC connection from the grid and the output is DC to the battery.

Though many topologies exist for active three-phase power factor conversion, a Vienna rectifier is popular due to its operation in continuous conduction mode (CCM), inherent multilevel switching (three level), and reduced voltage stress on the power devices. Traditionally, hysteresis-based controllers have been used for Vienna rectifiers. Only recently have sine triangle-based PWM been shown to work for Vienna Rectifier control. This control can be quite challenging to design. Several variants of Vienna rectifiers exist, 図 1 shows the variant of the Vienna rectifier chosen in this design along with the key voltages and currents being sensed.

図 1. Vienna Rectifier Variant Implemented



A Y-connection Vienna rectifier is implemented in this design guide. With this design, the purpose is to provide an example of how to control a Vienna rectifier and how to tune the different loops using the C2000 MCU.

1.1 Key System Level Specifications

The three-phase vienna rectifier key power specification are given in 表 1.

表 1. Key System Specifications

PARAMETER	SPECIFICATION
Input voltage (Vin)	• AC 208 Vrms VL-L or 120 Vrms L-N , 60 Hz or • AC 400 Vrms VL-L or 230 Vrms L-N , 50 Hz
Input current (Iin)	4 Amps RMS Max
Output voltage (Vout)	• 600-V DC bus nominal at 208 Vrms or • 700-V DC bus nominal at 400 Vrms
Output current (Iout)	Absolute RMS maximum 5 Amps, pulse maximum 10 Amps
Power rating	• 1.2 KW at three-phase 208 Vrms or • 2.4 KW at three-phase 400 Vrms
Current THD	• <1% at rated load with 208 Vrms • <4% at rated load with 400 Vrms
Efficiency	Peak 98%, average ~97%
Primary filter inductor	3 mH
Output capacitance	180 µF
PWM switching frequency	50 kHz

WARNING

TI intends this EVM to be operated in a *lab environment only and does not consider it to be a finished product* for general consumer use.

TI Intends this EVM to be used only by *qualified engineers and technicians* familiar with risks associated with handling high-voltage electrical and mechanical components, systems, and subsystems.

There are *accessible high voltages present on the board*. The board operates at voltages and currents that may cause shock, fire, or injury if not properly handled or applied. Use the equipment with necessary caution and appropriate safeguards to avoid injuring yourself or damaging property.

CAUTION

Do not leave EVM powered when unattended.

High voltage! There are ***accessible high voltages present on the board***. Electric shock is possible. The board operates at voltages and currents that may cause shock, fire, or injury if not properly handled. Use the equipment with necessary caution and appropriate safeguards to avoid injuring yourself or damaging property. For safety, use of isolated test equipment with overvoltage and overcurrent protection is highly recommended.

TI considers it the user's responsibility to confirm that the voltages and isolation requirements are identified and understood before energizing the board or simulation. ***When energized, do not touch the EVM or components connected to the EVM***.

Hot surface! Contact may cause burns. Do not touch!

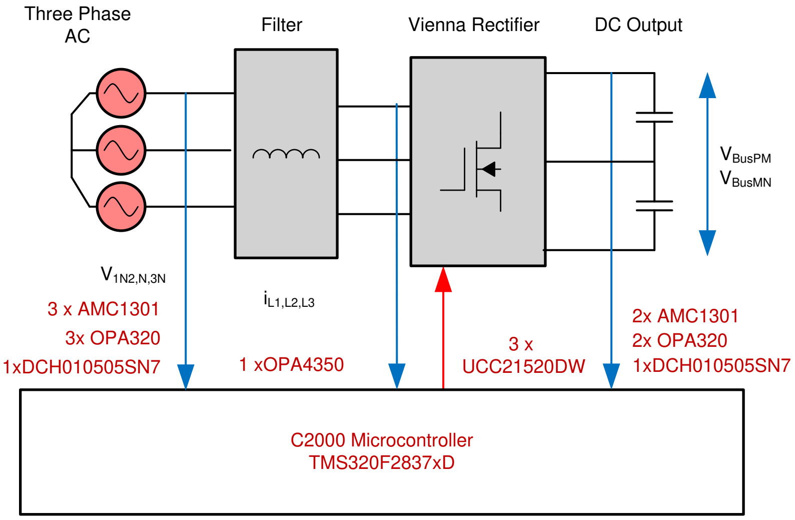
Some components may reach high temperatures >55°C when the board is powered on. The user must not touch the board at any point during operation or immediately after operating, as high temperatures may be present.

2 System Overview

2.1 Block Diagram

図 2 shows the block diagram of the Vienna rectifier chosen in this design along with the key voltages and currents being sensed.

図 2. Block Diagram



2.2 Highlighted Products

2.2.1 C2000™ MCU F2838x, F2837x and F28004x

The C2000 MCUs are an optimized MCU family for real-time control applications. The fast and high-quality analog-to-digital controller enables accurate measurement of the current and voltage signals, and the integrated comparator subsystem (CMPSS) integrates protection for overcurrent and overvoltage without use of any external devices. The optimized CPU core enables fast execution of control loop. Trigonometric operations are accelerated using the on-chip trigonometric math unit (TMU), which imparts additional speedup in control loop execution. The solution also provides an option to use the control law accelerator (CLA), on the F28004x, F2837x and F2838x. CLA is a co-processor that can be used to alleviate CPU burden and enable running faster loops and/or more functions on the C2000 MCU.

2.2.2 UCC21520

The UCC21520 is an isolated, dual-channel gate driver with a 4-A source and a 6-A sink peak current. The driver is designed to drive power MOSFETs, IGBTs, and SiC MOSFETs up to 5 MHz with a best-in-class propagation delay and pulse-width distortion. The input side is isolated from the two output drivers by a 5.7-kVRMS reinforced isolation barrier, with a minimum of 100-V/ns common-mode transient immunity (CMTI). Internal functional isolation between the two secondary-side drivers allows a working voltage of up to 1500-V DC. A disable pin shuts down both outputs simultaneously when set high and allows normal operation when left floating or grounded. The device accepts VDD supply voltages up to 25 V. A wide input VCCI range from 3 to 18 V makes the driver suitable for interfacing with both analog and digital controllers.

2.2.3 AMC1301

The AMC1301 is a precision isolation amplifier with an output separated from the input circuitry by an isolation barrier that is highly resistant to magnetic interference. The input of the AMC1301 is optimized for direct connection to shunt resistors or other low voltage level signal sources.

2.2.4 OPA320

The OPA320 is a precision, low-power, single-supply op amp optimized for very-low noise. Operated from a voltage range from 1.8 to 5.5 V, the device is well-suited for driving analog-to-digital converters (ADCs). With a typical offset voltage of 40 µV and very-low drift over temperature (1.5 µV/°C typical), it is very well suited for applications like control loop and current sensing in motor control.

3 Control System Design Theory

This section discusses the control system design theory

3.1 PWM Modulation

図 3 shows a simplified, single-phase diagram of the Vienna rectifier. To control this rectifier, the duty cycle is controlled such that it regulates the voltage v_{xiN}directly. That is, if the software variable *Duty* is set to 1, it makes v_{xiN} the largest voltage possible by never turning on Q1 and Q2 switches and letting the inductor connect to the DC bus though the bridge diode. Similarly when *Duty* is set to 0, PWM is modulated such that Q1 and Q2 always conduct making v_{xiN} connect to the midpoint of the DC bus (which is zero), which makes it the lowest possible voltage for the switching cycle.

図 3. Single Phase Diagram of Vienna Rectifier

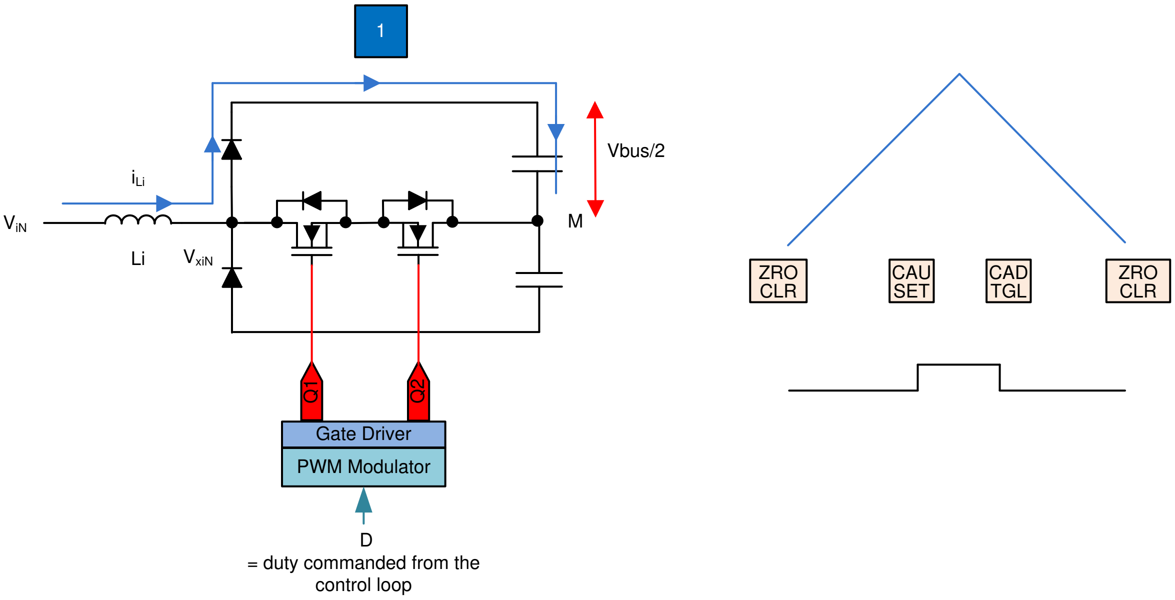
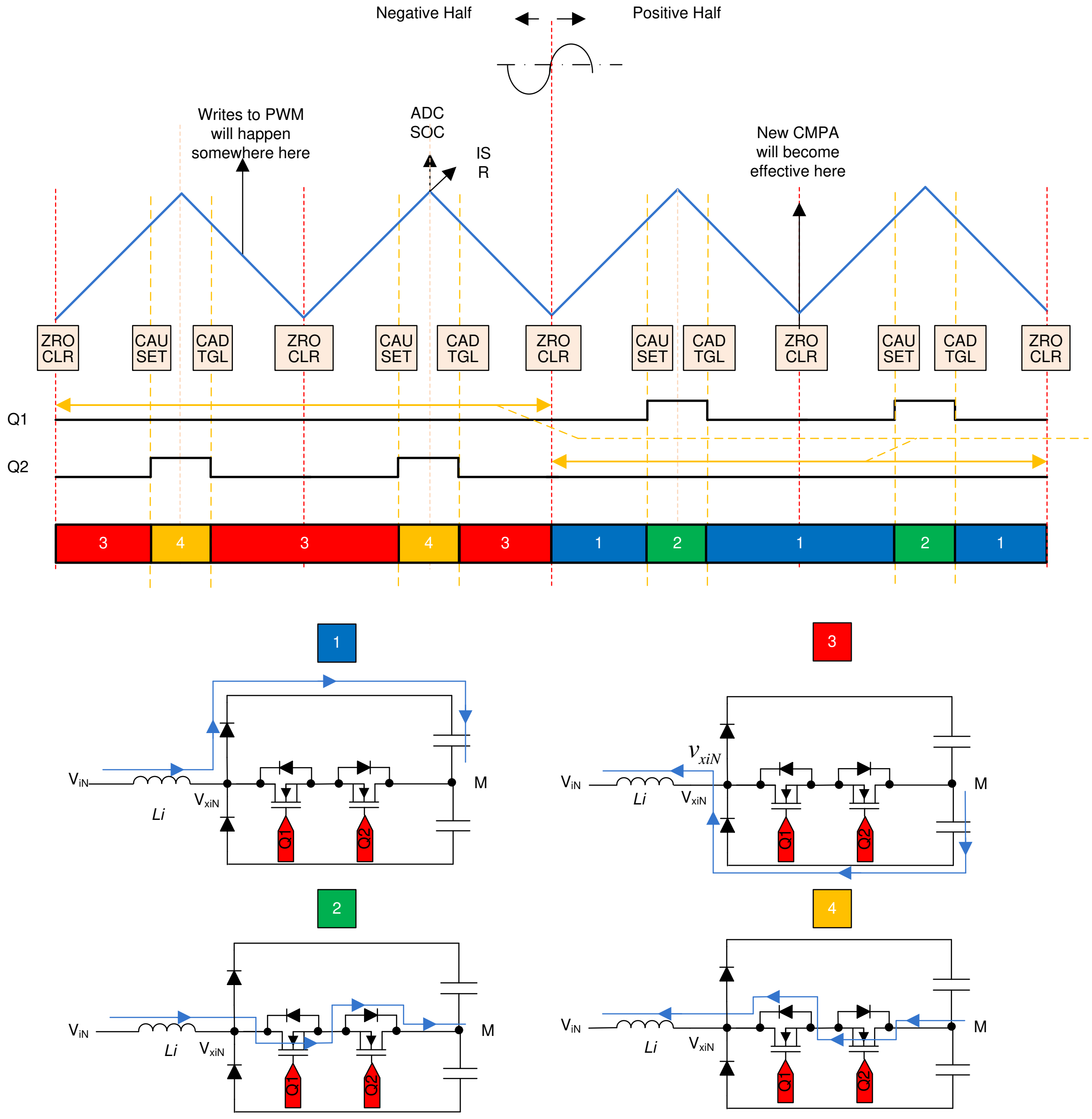


図 4 shows the detailed PWM configuration.

図 4. Vienna Rectifier Detailed PWM Modulation Scheme



Though Q1 and Q2 are shown to be not switching here for understanding, the current flow does not change even when it is switched. SW for TIDM-1007 switches the FETs synchronously with each other.

3.2 Current Loop Model

To understand the current loop model, first look at the inductor current closely. In 図 5 the duty cycle D is provided to the PWM modulator, which is connected to the switches Q1 and Q2. With this in mind, see 式 1:

$$V_{xiN} = D * \frac{V_{bus}}{2} \tag{1}$$

注: When D is set to 1, all the switches are *off*, and when D is 0, all switches are *on*, which connect the inductor to the point to M.

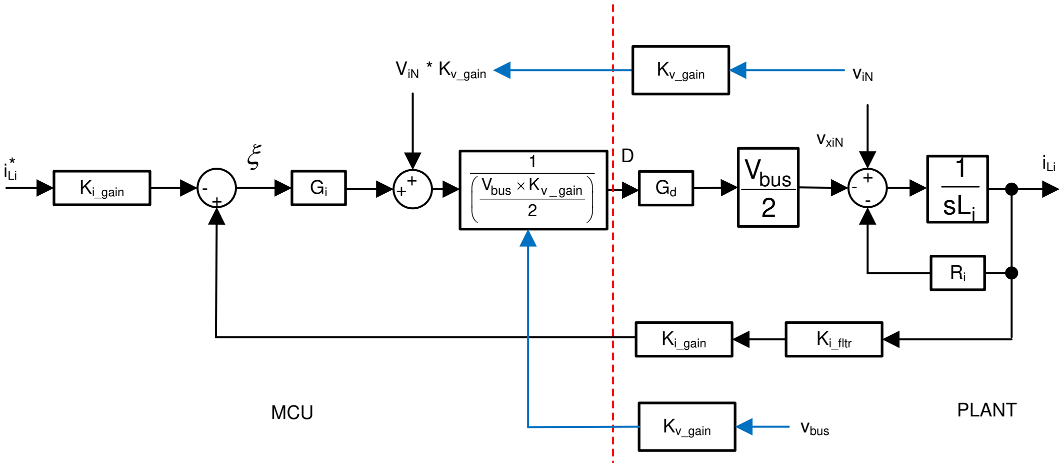
To modulate the current through the inductor, the voltage v_{xiN} is regulated using the duty cycle control of Q1 and Q2 switches. Assuming the direction of current is positive in the direction from the AC line into the rectifier and using the DC bus feedforward, the input AC voltage feedforward along with the assumption that the grid is fairly stiff. The current loop can be simplified as shown in 図 5, and the current loop plant model can be written as in 式 2.

$$H_{p\_i} = \frac{i_{Li}^*}{D} = \frac{1}{K_{v\_gain}} * K_{i\_gain} * K_{i\_fltr} * G_d * \frac{1}{Z_i} \tag{2}$$

Where:

- K_{v_gain} is the inverse of the maximum voltage sensed that is $1/V_{max\_sense}$ for the bus and the ac input. It is assumed the AC voltage max sense and the DC Bus voltage max sense are equal.
- K_{i_gain} is the inverse of the maximum AC current sensed.
- $K_{i_fltr}$is the response of the RC filter connected from the current sensor to the ADC pin.
- G_d is the digital delay associated with the PWM update and digital control.
- $i_{Li}{}^*$ is the current command, i_{Li} is the actual inductor current.
- $V_{bus}/2$ is the voltage across one of the output bus capacitor.
- Z_i is the impedance of the inductor which includes indictance L_i and resistance R_i .
- H_{p_i} is the current loop plant as seen by the digital controller G_i .
- v_{iN} is the instantaneous AC voltage at the input.

図 5. Current Loop Control Model



注: The negative sign on the reference is in place because the current loop is thought to be regulating the voltage v_{xiN}. To increase the current, v_{xiN} must be reduced and, thus, the opposite sign for reference and feedback in 図 5. This current loop model is used to tune the current compensator. A simple proportional controller is used for the current loop. The gain of the proportional gain is adjusted to ensure the system is stable.

3.3 DC Bus Regulation Loop

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

Small signal model of the DC bus regulation loop is developed by linearizing 式 3 around the operating point.

$$i_{DC} v_{bus} = 3 n v_{Nrms} i_{Nrms} \Rightarrow \hat{i}_{DC} = 3n \frac{\bar{V}_{Nrms}}{\bar{V}_{bus}} i_{Li} \tag{3}$$

For resistive load the bus voltage and current relate, as shown in 式 4.

$$\hat{V}_{bus} = \frac{R_L}{1 + sR_L C_o} \hat{i}_{DC} \tag{4}$$

The DC voltage regulation loop control model can be drawn, as shown in 図 6. An additional Vbus feedforward is applied to make the control loop independent of the bus voltage, and, thus, the plant model for the bus control can be written as shown in 式 5.

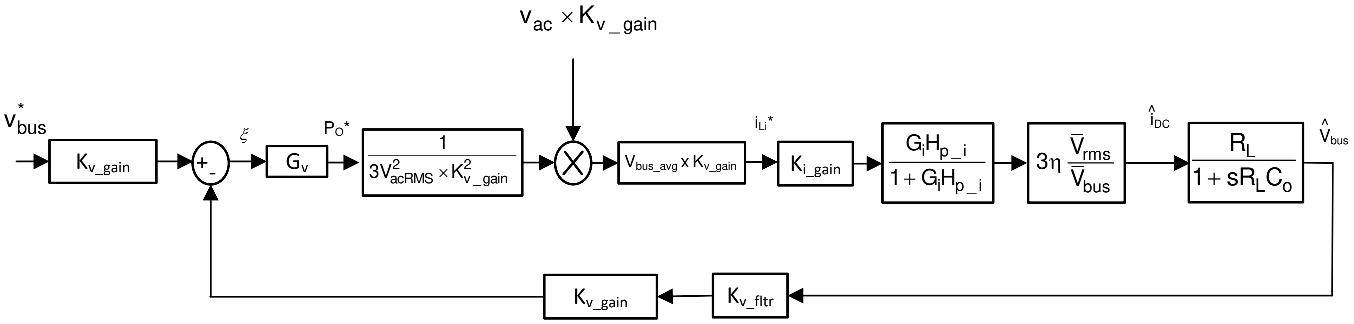
$$H_{p\_bus} = H_{load} * N * K_{i\_gain} * K_{v\_gain} * K_{v\_flt} \tag{5}$$

Where:

- H_{p_bus} is the voltage loop plant as seen by the digital controller G_v.
- Output of the G_v is the power reference $P_o{}^*$
- $v_{bus}{}^*$ is the voltage command/reference, v_{bus} is the actual bus voltage.
- C_o is the output capacitor, R_L is the load resistance.

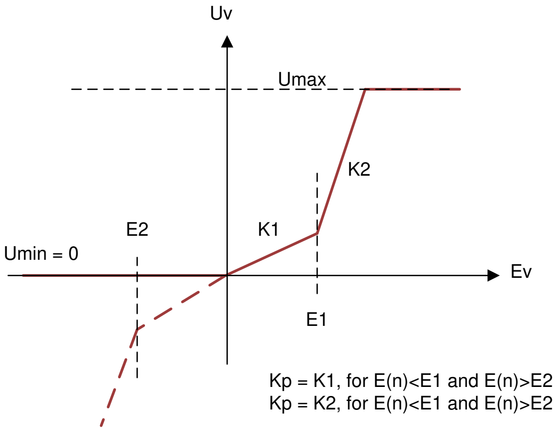
Using 図 6, a proportional integrator (PI) compensator is designed for the voltage loop. The bandwidth of this loop is kept low as it is in conflict with the THD below steady state.

図 6. DC Voltage Loop Control Model


$v_{ac} \times K_{v\_gain}$
v^*_{bus}
K_{v_gain}
+
-
ξ
G_v
P_O*
$\frac{1}{3V^2_{acRMS} \times K^2_{v\_gain}}$
$i_{Li}*$
$V_{bus\_avg} \times K_{v\_gain}$
K_{i_gain}
$\frac{G_iH_{p\_i}}{1+G_iH_{p\_i}}$
$3\eta \frac{\bar{V}_{rms}}{\bar{V}_{bus}}$
$\hat{i}_{DC}$
$\frac{R_L}{1+sR_LC_o}$
$\hat{V}_{bus}$
K_{v_fltr}
K_{v_gain}

Additionally, a non-linear PI loop is used to reduce the transients in case of step load changes. 図 7 shows the structure of the non-linear PI loop implemented on this design

図 7. Non-Linear PI Loop for Voltage Controller


Uv
Umax
K2
E2
K1
Umin = 0
Ev
E1
Kp = K1, for E(n)<E1 and E(n)>E2
Kp = K2, for E(n)<E1 and E(n)>E2

3.4 DC Voltage Balance Controller

A split capacitor is used for the output voltage bus in the Vienna rectifier. The voltages across these capacitors may not naturally stay balanced hence a DC balance controller loop is added. This loop modulates an offset, which is added to the duty cycle, thus, modulating the current through the midpoint to balance the voltages on the split capacitor.

A simple proportional gain is used for the DC bus balance controller, with the output of the balance loop given as in 式 6.

$$G_{s\_out} = (V_{bus\_PM} - V_{bus\_MN}) * G_{s\_gain\_Kp} \quad (6)$$

4 Hardware Design Theory

Below are details on selection of the inductor, output capacitor, and the sensing schemes used in the design. Refer to the excel sheet in the SW install package for more details on each one.

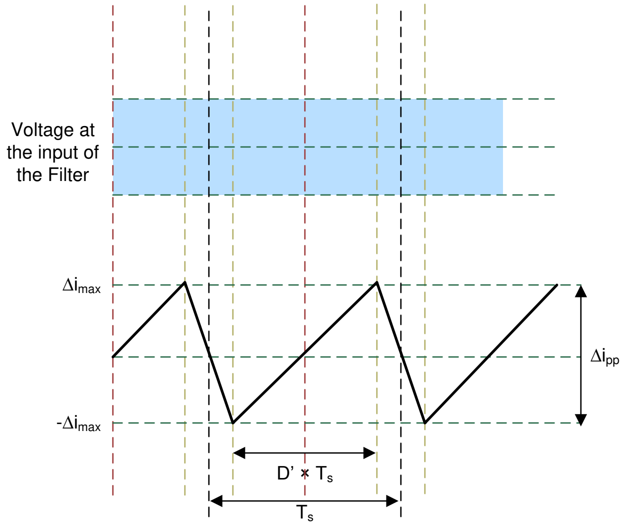
C2000Ware_DigitalPower_SDK_<version>\solutions\tidm_1000\hardware\

calculations.xlsx

4.1 Inductor Design

Input inductor (Li) filters out the switching frequency harmonics. Inductor design, amongst other factors, depends on calculation of the current ripple and choosing a material for the core that can tolerate the calculated current ripple. 図 8 shows one switching cycle waveform of the inverter output voltage vi with respect to the inductor current.

図 8. Current Ripple Calculation



The voltage across the inductor is given by V = L_i(di/dt). For the Vienna rectifier, see 式 7.

$$\left(\frac{V_{bus}}{2}-V_{in}\right)=L_i*\frac{\Delta i_{pp}}{D'*T_s} \tag{7}$$

where T_s = 1/$F_{sw}$is the switching period and D’ is the duty cycle for which the switches are ON. For control design, D is assumed to be the voltage at the other terminal of the inductor and is related to D’ by D’ = 1 - D. Rearranging the current ripple at any instant in the AC waveform is given as 式 8.

$$\Delta i_{pp}=\frac{D'*T_s*\left(\frac{V_{bus}}{2}-V_{iN}\right)}{L_i} \tag{8}$$

Now assuming modulation index to be m_a the duty cycle can be given as D' = m_a*sin(wt) and assuming that v_{iN}=D'*(V_{bus}/2), 式 9 can be derived.

$$\Delta i_{pp}=\frac{\frac{V_{bus}}{2}*T_S*m_a*\sin(\omega t)*(1-m_a\sin(\omega t))}{L_i} \tag{9}$$

From 式 9, it is clear that the peak ripple is a factor where the input AC is in the sinusoidal waveform.

To get the maximum value differentiating the equation with respect to time, use 式 10.

$$\frac{d(\Delta i_{pp})}{dt}=K\left\{\cos(\omega t)(1-m_a\sin(\omega t))-m_a\sin(\omega t)*\cos(\omega t)\right\}=0 \tag{10}$$

Which gives the maximum ripple exists at sin(wt)=1/(2*m_a), Substituting this value, 式 11 is derived.

$$\Delta i_{pp_{max}}=\frac{\frac{V_{bus}}{2}*T_s}{4*L_i}\Rightarrow L_i=\frac{\frac{V_{bus}}{2}}{4*F_{sw}*\Delta i_{pp_{max}}} \tag{11}$$

With these values in mind, an appropriate core can be selected along with an inductor designed to meet this inductance value.

4.2 Bus Capacitor Selection

The bus capacitor is responsible for removing the ripple on the DC voltage that can be caused by the draw of sinusoidal currents. The capacitor value and the DC bus ripple are related by 式 12.

$$C=\left(\frac{1}{3}\right)\frac{P_{ac}}{4*f*\left(V^2-(V-\Delta V)^2\right)} \tag{12}$$

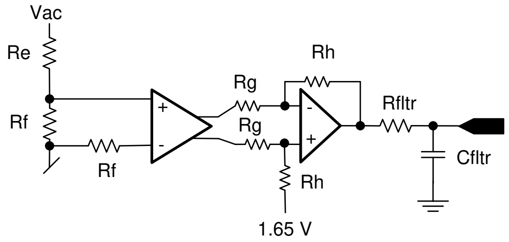
This equation is used to select the minimum DC bus capacitance value.

注: The above outlined calculation may yield to over design of the capacitor. The capacitor sizing is more dependent on the load and the nature of current it draws. For three phase PFC the power ripple is fairly small as the input always has path to the output. Hence, please use the equation above for reference only and be aware that this will yield to overdesign.

4.3 Input AC Voltage Sensing

First a virtual neutral is constructed by using a resistor network connected in Y along with some capacitance for stability. In this design the controller is kept on the cold side, thus, an isolated amplifier AMC1301 is used to process the VL-N’ voltages as shown in 図 9. As the AMC1301 is designed considering low-impedance sources for current sensing application, the input differential resistance plays a non-linear role in the total gain calculation. Therefore, a final calibration during build level one must be done, and the maximum AC voltage range must be adjusted according to the calibration. Alternatively, one can use TINA-TI™ software for simulation.

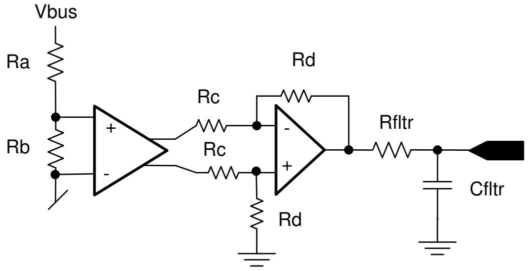
図 9. Input AC Voltage Sensing



4.4 Bus Voltage Sensing

Similarly, the bus voltage, which is split between two capacitors, is sensed using AMC1301 and OPA320 as shown in 図 10.

図 10. Bus Voltage Sensing



As accuracy is of importance in the bus voltage sensing, a further step of calibration with offset and gain adjustment is carried out by doing a regression analysis. A sample of how this is done is shown in the excel sheet in the install package called:

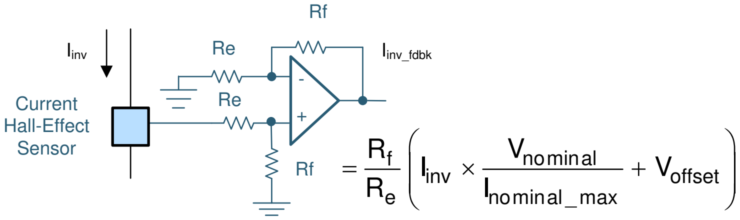
C2000Ware_DigitalPower_SDK_<version>\solutions\tidm_1000\hardware\BusMeasCalibrationRoutine.xlsx

- sheet > calData (this is the raw data that is used to run regression)
- sheet > PM_Reg
- sheet > MN_Reg (these are the results of running the regression that provide the intercept and gain values for the adjustment of the measurement)

4.5 Inductor Current Sensing

A hall effect sensor is used to sense the current through the inductor. The hall effect sensor has an inbuilt offset, and the range is different than what ADC can measure. Therefore, the voltage is scaled to match the ADC range using the circuit shown in 図 11.

図 11. Current Sense Using Hall Effect Sensor

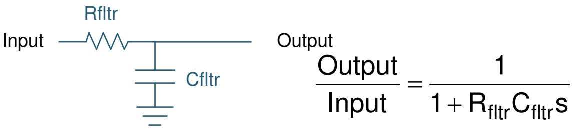


$$= \frac{R_f}{R_e}\left(I_{inv} \times \frac{V_{nominal}}{I_{nominal\_max}} + V_{offset}\right)$$

4.6 Sense Filter

An RC filter is used to filter the signals before connecting to the inverter. A common RC filter is used for all the sensing signals on this design, as shown in 図 12.

図 12. RC Filter

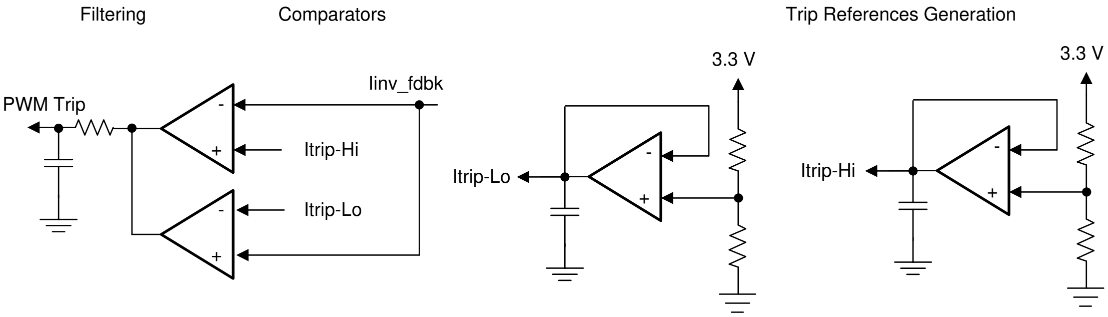


$$\frac{Output}{Input} = \frac{1}{1 + R_{fltr}C_{fltr}s}$$

4.7 Protection (CMPSS)

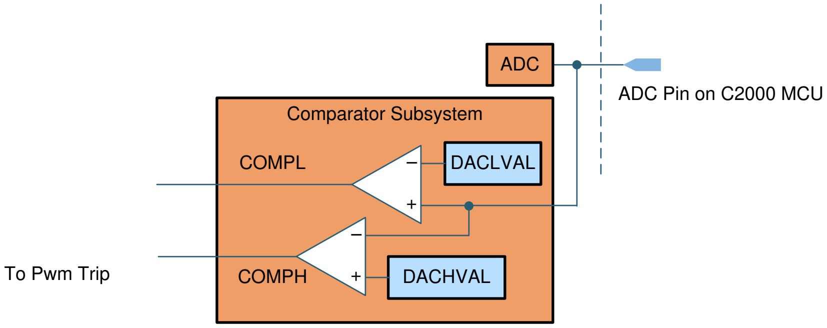
Most power electronics converters need protection from an overcurrent event. For this design multiple comparators are required, and references for the trip must be generated, as shown in 図 13.

図 13. Trip Generation for PWM Using Comparators and Reference Generators



All this circuitry is avoided when using the C2000 MCUs such as TMS320F28379D and TMS320F280049C, which have on-chip windowed comparator as part of the CMPSS that are internally connected to the PWM module and can enable fast tripping of the PWM. This device saves board space and is cost efficient in the end application as extra components can be avoided using on-chip resources, as shown in 図 14.

図 14. Comparator Subsystem (CMPSS) Used for Overcurrent Protection



4.8 *Efficiency Estimates*

An estimate of the expected efficiency is shown in

C2000Ware_DigitalPower_SDK_<version>\solutions\tidm_1000\hardware\calculations.xlsx

- sheet > Efficiency Estimate

5 Getting Started Hardware

This section details the hardware and explains the different sections on the board. If using the firmware of the design alone through powerSUITE, this section may not be valid.

5.1 *Base Board Settings*

The design follows a HSEC control card concept. Any device for which a HSEC control card is available from the C2000 MCU product family can be potentially used on this design. The key resources used for controlling the power stage on the MCU are listed in 表 2. 図 15 shows the key power stage and connectors on the design guide, and 表 3 lists the key connectors and their functions. To get started:

1. Make sure no power source is connected to the design.
2. Insert the control card in the J8-J10 slot.
3. Insert a jumper at J11 and J12 to connect the bias supply for 5 V and 3.3 V .
4. Connect a 12-V DC, 1-Amps power supply at J4. A few LEDs on the base board will light up indicating power. The LED on the control card will also light up. This light indicates the device is powered up as well.

注: The bias for the MCU is separated from the power stage, which enables safe bring up of the system.

5. To connect JTAG, use a USB cable from the control card and connect it into a host computer.
6. A three-phase power supply is connected to the input J2. The board works as a three-wire system, therefore, the connection of the neutral is not required. A virtual neutral is generated on the board for sensing purpose of VL-N voltages.
7. A resistive load of approximately 500Ω should also be connected to the output at J1. The load is connected across the positive and negative terminal (Vbus). The midpoint (M) is not connected to the load.

8. Current and voltage probes can be connected to observe the input current, input voltage, and output voltages, as shown in 図 16.

図 15. Board Overview

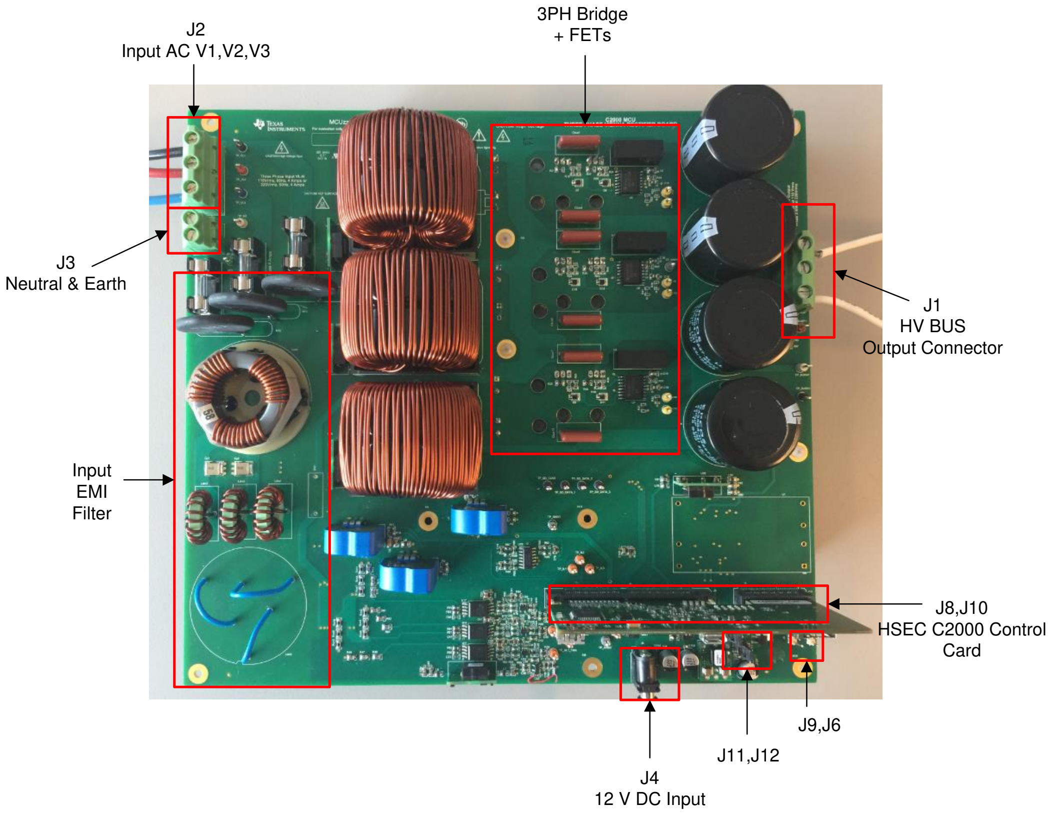


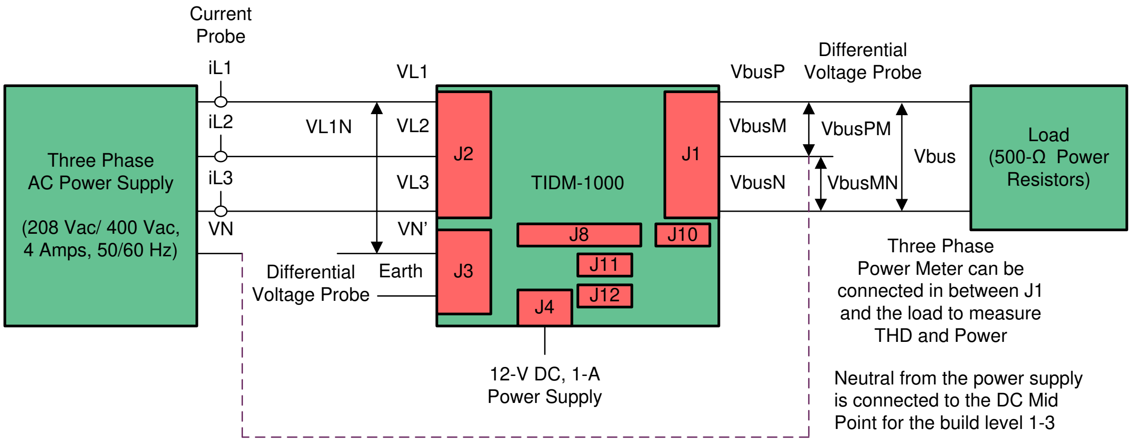
表 2. Key Controller Peripherals Used for Control of Power Stage on Board

SIGNAL NAME	HSEC PIN NUMBER	FUNCTION
PWM-1A	49	PWM: Vienna rectifier PWM Ph1A
PWM-1B	51	PWM: Vienna rectifier PWM Ph1B
PWM-2A	53	PWM: Vienna rectifier PWM Ph2A
PWM-2B	55	PWM: Vienna rectifier PWM Ph2B
PWM-3A	50	PWM: Vienna rectifier PWM Ph3A
PWM-3B	52	PWM: Vienna rectifier PWM Ph3B
IL1	15, 20	ADC with CMPSS: Inductor current measurement Ph1
IL2	21.27	ADC with CMPSS: Inductor current measurement Ph2
IL3	25	ADC with CMPSS: Inductor current measurement Ph3
V1	18	ADC: AC voltage sensing Ph1
V2	28	ADC: AC voltage sensing Ph2
V3	34	ADC: AC voltage sensing Ph3
Vbus_PM	31, 24	ADC: Bus voltage positive terminal to midpoint
Vbus_MN	37, 26	ADC: Bus voltage midpoint to negative terminal
Profiling GPIO1	58	GPIO: Used to profile code (optional)
Profiling GPIO2	60	GPIO: Used to profile code (optional)

表 3. Key Connectors and Function

CONNECTOR NAME	FUNCTION
J1	Output high voltage bus
J2	Input three phase voltage
J3	Neutral and Earth connection
J4	Input bias supply, 12-V DC 1 Amps
J5	DAC output for debugging
J6	Ground
J7	Profiling GPIO connector
J8,J10	HSEC control card connector slot
J9	Remote enable connector
J11	5-V jumper, used to disconnect the 5-V supply
J12	3-V jumper, used to disconnect the 12-V jumper

図 16. Hardware Setup to Run Software



5.2 Control Card Settings

Certain setting 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 F28388D control card. Refer to the information sheet located inside C2000Ware at

<sdk_install_path>\c2000ware\boards\controlCARDs\TMDSCNCD28388D:

1. S1:A on the control card must be set on both ends to **ON** position to enable JTAG connection to the device and the UART connection for SFRA GUI. If this switch is **OFF**, one cannot use the isolated JTAG built in on the control card nor can SFRA GUI communicate to the device.
2. J1:A is the connector for the USB cable that is used to communicate to the device from a host PC on which Code Composer Studio (CCS) runs.
3. Based on the TMDSCNCD28388D_user_guide inside C2000ware, below debug mode using CCS and the onCard xds100v2 emulator, S2 is suggested to set in this way to put the C2000 device into WaitMode which can reduce the risk of connectivity issues: Position 1: OFF, Position 2: ON.
4. 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 F28388D control card, this means S3 and S4 are moved to the end with "." that is, to the left, which puts 3.3-V VDDA as the reference for the ADC. Refer to the information sheet for more information.

Certain setting 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. Refer to 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 JTAG connection to the device and the UART connection for SFRA GUI. If this switch is **OFF (down)**, one cannot use the isolated JTAG built in on the control card nor can SFRA GUI 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 Code Composer Studio (CCS) runs.
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, which puts 3.3-V VDDA as the reference for the ADC. Refer to the information sheet for more information.

Certain settings on the device control card are required to communicate over JTAG and use the isolated UART port. The user must also provide a correct ADC reference voltage. The following settings are required for revision A of the TMS320F280049C control card (refer to the info sheet located at *<sdk_install_path>\c2000ware\boards\controlCARDs\TMDSCNCD280049C*.

1. Set both ends of S1:A on the control card to **ON** (up) position to enable JTAG connection to the device and UART connection for SFRA GUI. If this switch is **OFF** (down), the user cannot use the isolated JTAG built in on the control card, nor can the SFRA GUI communicate with the device.
2. Connect the USB cable to J1:A to communicate with the device from a host PC on which CCS runs.
3. For the control loop, use the internal reference of the TMS320F28004x and move the S8 switch to the left (that is, pointing to VREFHI).
4. For the best performance of this reference design, remove the capacitor connected between the isolated grounds on the control card, C26:A.
5. GPIO24 through GPIO27 are muxed on the TMS320F280049C control card. Put all the switches on SW5 to **OFF** (up) and all the switches on SW6 to **ON** (up).

6 Getting Started Firmware

The software of this design is available inside C2000Ware_DigitalPower_SDK and is supported inside the powerSUITE framework.

注: The firmware for the solution is supported on both the TMS320F28379D and TMS320F280049C devices.

6.1 *Opening the Project Inside Code Composer Studio™*

To start:

1. Install CCS (version 9.0.1 or above).
2. Open **CCS**.
3. Go to **View** >**CCS App Center**.
4. Below **Code Composer Studio Add-ons**, make sure GUI Composer Runtime in installed. If the GUI is not installed, install GUI Composer Runtime.
5. Install C2000Ware DigitalPower SDK at the *DigitalPower Software Development Kit(SDK)for C2000 Microcontrollers* tools folder.

注: powerSUITE is installed with the DigitalPower SDK in the default install.

6. Close **CCS**, and open a **new workspace**.
7. Close **CCS**, and open a new workspace. CCS will automatically detect powerSUITE. A restart of CCS may be required for the change to be effective.
8. Go to **View** >**Resource Explorer**. Below the TI Resource Explorer, go to **Software > C2000Ware DigitalPower SDK - <version>** .

Open the design guide software as is (opens firmware as it was run on this design and hardware, requires the board to be exactly the same as this design guide, and does not allow modification through the powerSUITE GUI inside the project).

1. Below **C2000Ware DigitalPower SDK** , select **Development Kits** > **Three Phase PFC Vienna Rectifier TIDM-1000**, and click **Run <device> Project**.
2. This will import the project and the development kit or designs page will show up. This page can be used to browse all the information on the design including this user guide, test reports, hardware design files, and so forth.
3. Click **Run <device_name> Project**.
4. This imports the project into the workspace environment, and a syscfg page with a GUI similar to Figure17: powerSUITE Page for Vienne Rectifier Solution 変更 appears.

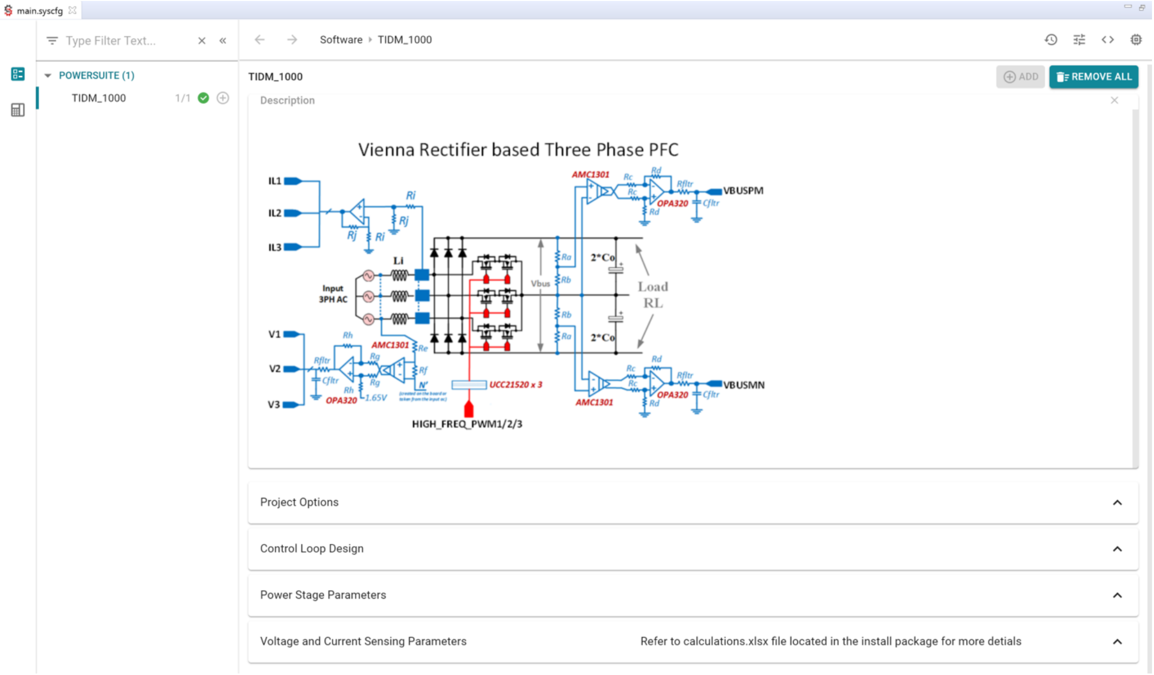
注: As this project is imported from the development kit and design guide page, modifications to the power stage parameters through the GUI will not be allowed.

5. If this GUI page does not appear, refer to the FAQ section below **powerSUITE** in the **C2000Ware DigitalPower SDK** resource explorer.

Open design guide software for adaptation. The user can modify power stage parameters, which are then used to create the model of the power stage in Compensation Designer and can also modify scaling values for voltages and currents.

1. In the resource explorer below C2000Ware DigitalPower SDK powerSUITE, click the **Solution Adapter Tool** ( ).
2. Select **Three Phase PFC** from the list of solutions presented.
3. Select the device this solution needs to run on the next page.
4. When the icon is clicked, a pop-up window displays asking for a location to create the project. You can also save the project inside the workspace itself. When the location is specified, a project is created, and a GUI page appears with modifiable options for the solution 図 17.
5. You can use this GUI to change the parameters for an adapted solution, for example, power rating, inductance, capacitance, sensing circuit parameters, and so forth.
6. If this GUI page does not appear, refer to the FAQ section below **powerSUITE** in the **C2000Ware DigitalPower SDK** resource explorer.

図 17. powerSUITE Page for Vienna Rectifier Solution



Project Options

Control Loop Design

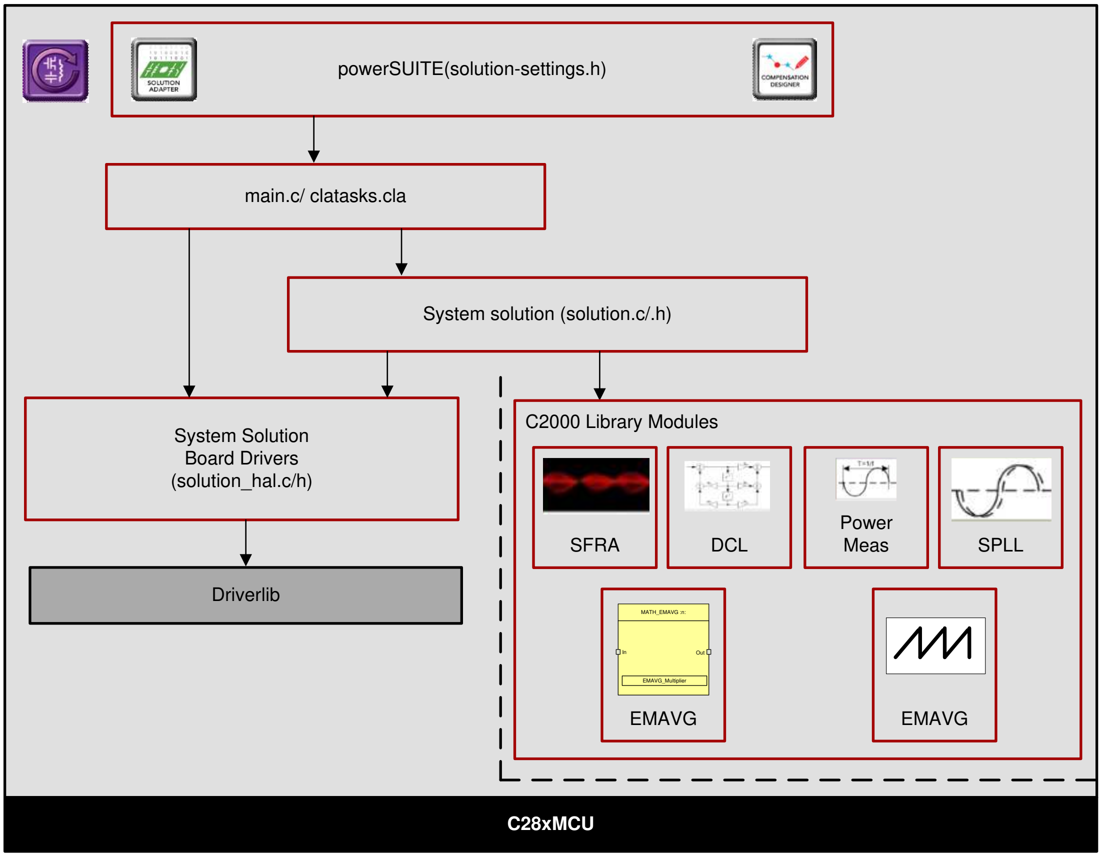
Power Stage Parameters

Voltage and Current Sensing Parameters Refer to calculations.xlsx file located in the install package for more detials

6.2 Project Structure

図 18 shows the general structure of the project.

図 18. Project Structure Overview



注: 図 18 shows the project for F2837x; however, if a different device is chosen from the powerSUITE page, the structure is similar.

Solution specific and device independent files are **<solution>.c/h**. Board-specific and device-specific files are **<solution>_hal.c/h**. The **main.c** file of the project consists of the main function of the C project as well as ISR framework and calls routines in other files. <**solution**>**_settings.h** and <**solution**>_user_**settings.h** consist of all the settings required for the solution. <**solution**>_user_**settings.h**

For this design, <**solution**> is **vienna**.

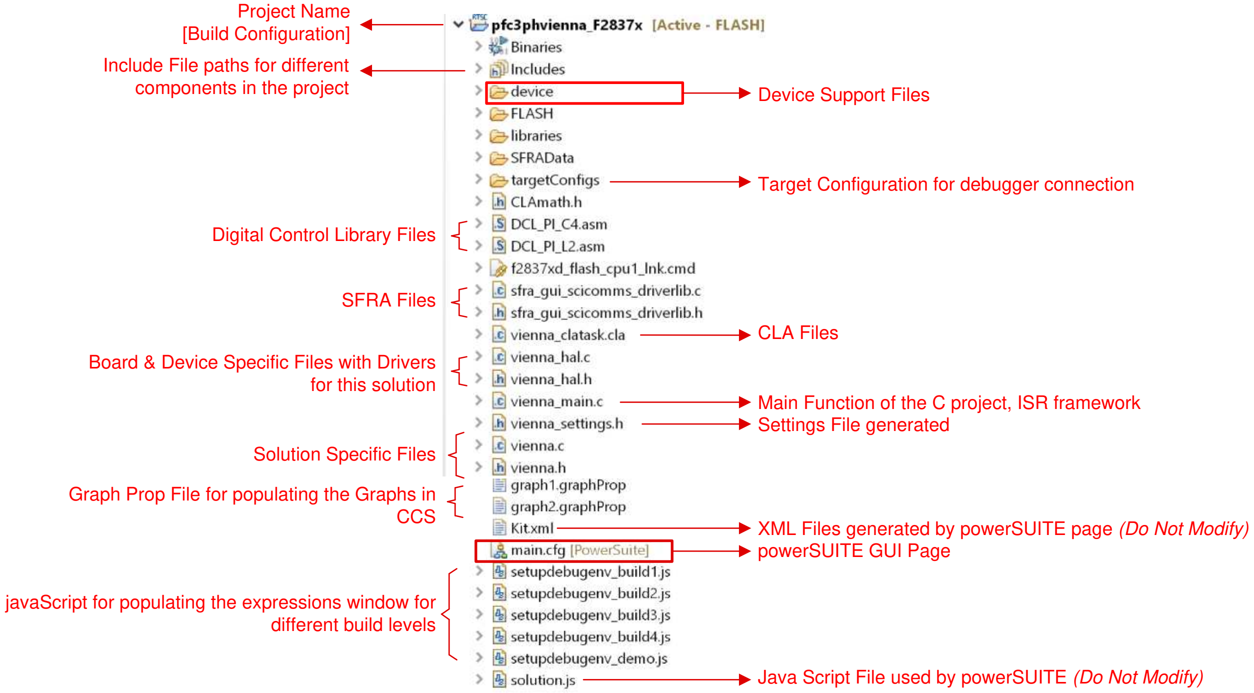
<**solution**>**_clatask.cla** file include the CLA Task definitions.

The powerSUITE page can be opened by clicking the **main.syscfg** file, listed below the project explorer. The powerSUITE page generates the **<solution>_settings.h** file. This file is the only 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 will be overwritten by powerSUITE each time the project is rebuilt. The changes will only happen after rebuilding the entire project

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

When the project is imported, the project explorer appears inside CCS as shown in 図 19.

図 19. Project Explorer View of Solution Project



Project Name [Build Configuration]

Include File paths for different components in the project

Device Support Files

Target Configuration for debugger connection

Digital Control Library Files

SFRA Files

CLA Files

Board & Device Specific Files with Drivers for this solution

Main Function of the C project, ISR framework

Settings File generated

Solution Specific Files

Graph Prop File for populating the Graphs in CCS

XML Files generated by powerSUITE page *(Do Not Modify)*

powerSUITE GUI Page

javaScript for populating the expressions window for different build levels

Java Script File used by powerSUITE *(Do Not Modify)*

The project consists of an interrupt service routine, which is called every PWM cycle, called controlISR(), where the control algorithm is executed. In addition to this, there are background tasks A0-A4 and B0-B4, which are called in a polling fashion and can be used to run slow tasks for which absolute timing accuracy is not required. A slower 10-kHz routine **tenkHzISR()** is called for instrumentation and to run some slower tasks that require timing accuracy.

The software of this design guide is organized in four incremental builds (INCR_BUILD). The incremental build process simplifies the system bring up and design.

- INCR_BUILD 1: Open Loop Check
- INCR_BUILD 2: Closed Current Loop
- INCR_BUILD 3: Closed Voltage and Current Loop
- INCR_BUILD 4: Closed Voltage, Current, and Bus Cap Balance Loop

These build levels are detailed in the section below. If using the design guide hardware, make sure the hardware setup is completed as outlined in 5.

6.3 Using CLA on C2000 MCU to Alleviate CPU Burden

The control law accelerator (CLA) is a co-processor available on the C2000 MCU family of devices. This co-processor enables offloading the control-ISR functions from the main C28x CPU core.

To run the control ISR on the CLA, for solutions supported in powerSUITE, selection is achieved through a drop-down menu on the powerSUITE SYSCFG page. The software structure of the powerSUITE solution is designed such that offloading the task to the CLA is simply a drop-down menu selection. The code is not duplicated and a single source for the solution algorithm is maintained, even when code is run on the CLA or the C28x. This configuration enables flexible debugging of the solution.

The CLA features vary slightly from device to device, for example, on the F2837xD, F2837xS, and F2807x the CLA can support only one task at a given time and there is no nesting capability, which means the task is not interruptible. Hence, realistically only one ISR can be offloaded to the CLA. On the F28004x, the CLA supports a background task from which a regular CLA task can nest. This configuration enables offloading two ISRs on the CLA.

Device-specific information:

- F2837xD: The CLA tasks do not support nesting, hence only one ISR can be offloaded to the CLA. Therefore, only the faster ISR, which on this design runs at 50 kHz, is offloaded to the CLA. The 50-kHz ISR on the C28x CPU takes approximately 20% of CPU bandwidth, and for the instrumentation ISR (10 kHz) approximately 4% of CPU bandwidth is used. Thus, the total CPU use is 24%. With the CLA option, this CPU burden is reduced to 4%.
- F28004x/F2838xD: The CLA supports a background task from which it can nest into a CLA task. This configuration allows offloading two ISR functions to the CLA. Hence, for the F28004x/F2838xD, both the control ISR (50 kHz) and instrumentation ISR (10 kHz) are offloaded to the CLA. On the F28004x, the CPU use is approximately 40% for the 50 kHz loop and 8% for the 10 kHz loop. Thus, the total CPU use is approximately 48%. With the CLA option, the CPU burden is reduced to 0% when both ISRs are offloaded to the CLA.

For more information on the CLA, visit the CLA Hands-On Workshop and the respective device technical reference manuals.

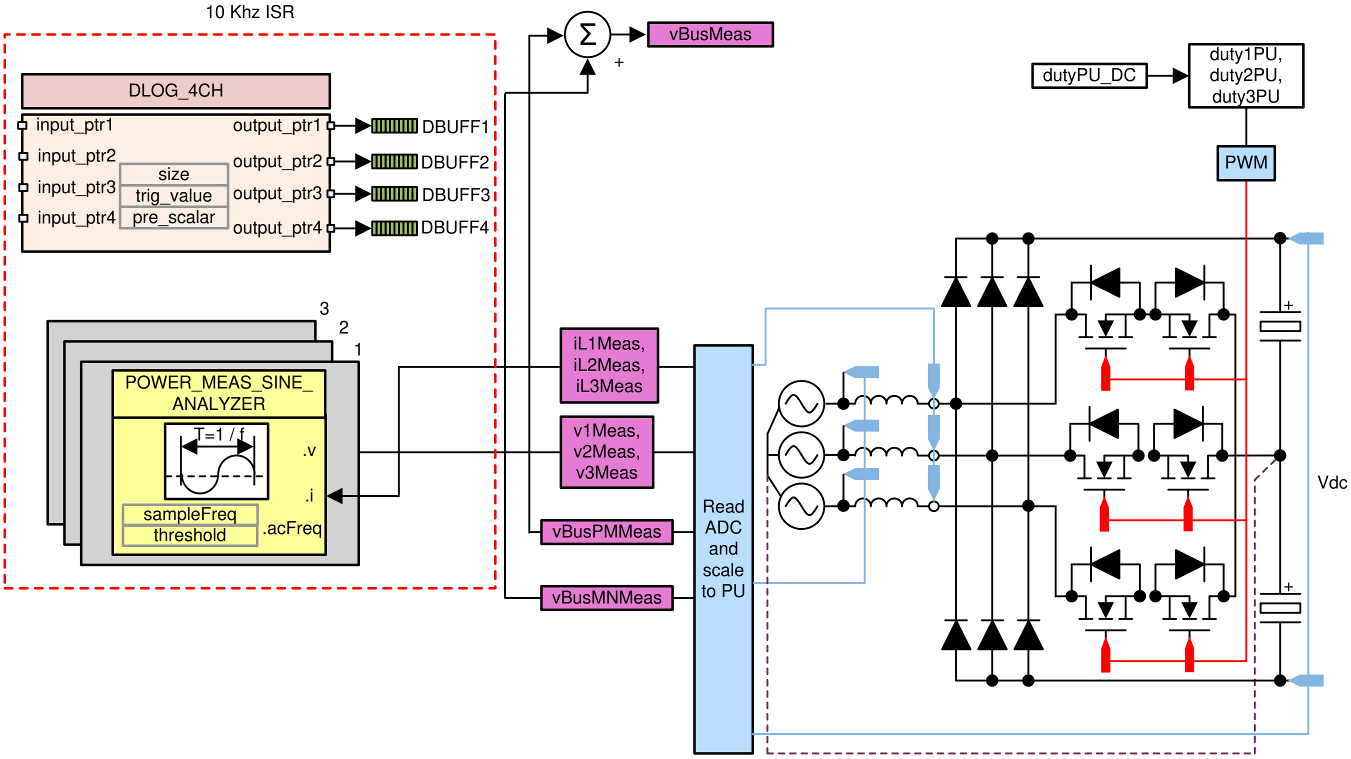
6.4 Running Project

注: The variable name mentioned in the user guide is simplified compared with the real name in the code. For example, VIENNA_guiVbus_Volts *is simplified as guiVbus.*

6.4.1 INCR_BUILD 1: Open Loop

In this build, the board is run in an open-loop fashion with a fixed-duty cycle. The duty cycle is controlled with dutyPU_DC variable. This build verifies the sensing of feedback values from the power stage and also the operation of the PWM gate driver, which ensures there are no hardware issues. Additionally, calibration of input and output voltage sensing can be performed in this build. The software structure for this build is shown in 図 20. Blocks that run in the slower ISR are marked. Other blocks are run in the fast controlISR.

図 20. Build Level 1 Control Software Diagram: Open Loop Project



6.4.1.1 Setting Software Options for BUILD 1

1. Make sure the hardware is setup as outlined in 5.1. Do not supply any high voltage (HV) power yet to the board. For this build, connect the neutral from the three-phase power supply to the board (図 16).
2. powerSUITE Settings - powerSUITE page below *Project Options* section:

   Select *Open Loop* for the build level.

If this is an adapted solution, edit the setting below *ADC Sensing Parameters* with the sensing resistors used in each case. Specify the switching frequency, the dead band, and the power rating. Save the page.

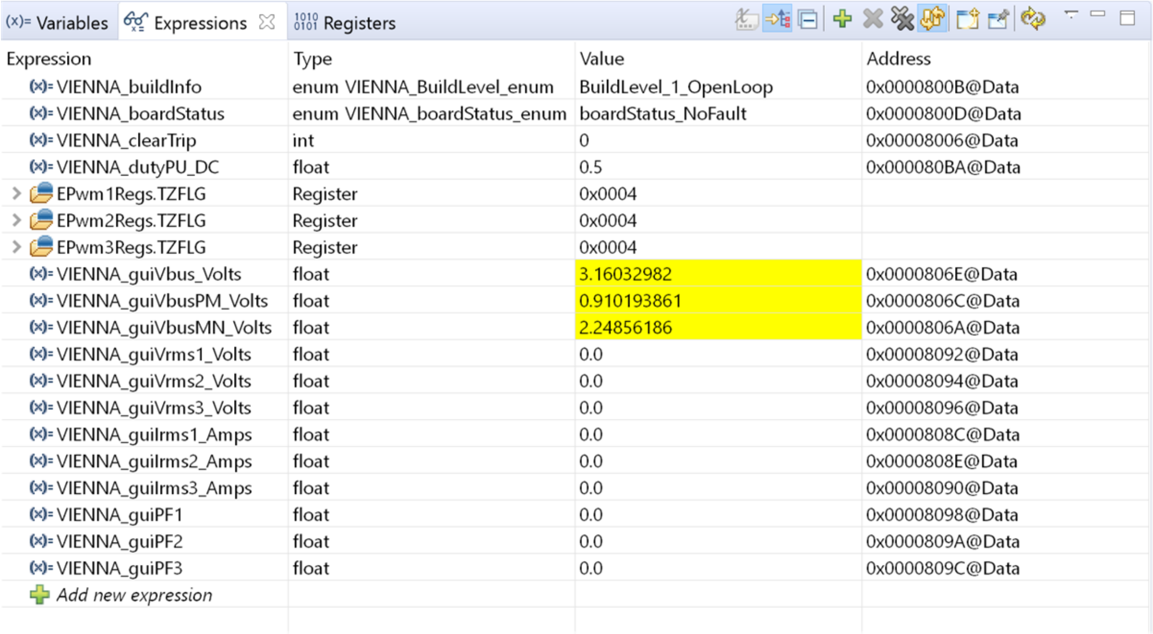
6.4.1.2 Building and Loading Project

1. Right click on the project name, and click **Rebuild Project**.
2. The project will build successfully.
3. In the **Project Explorer** make sure the correct target configuration file is set as Active below tragetconfigs (図 19).
4. Then, click **Run** > **Debug**. This launches a debugging session. In the case of dual CPU devices, a window may appear to select the CPU the debug needs to be performed. In this case, select CPU1.
5. The project then loads on the device, and CCS debug view becomes active. The code halts at the start of the main routine.

6.4.1.3 Setup Debug Environment Windows

1. To add the variables in the watch and expressions window, click **View** > **Scripting Console** to open the scripting console dialog box. On the upper right corner of this console, click Open, then browse to the **setupdebugenv_build1.js** script file located inside the project folder. This script file will populate the watch window with appropriate variables needed to debug the system. Click the Continuous Refresh button on the watch window to enable continuous update of values from the controller. The watch window will appear as shown in 図 21.

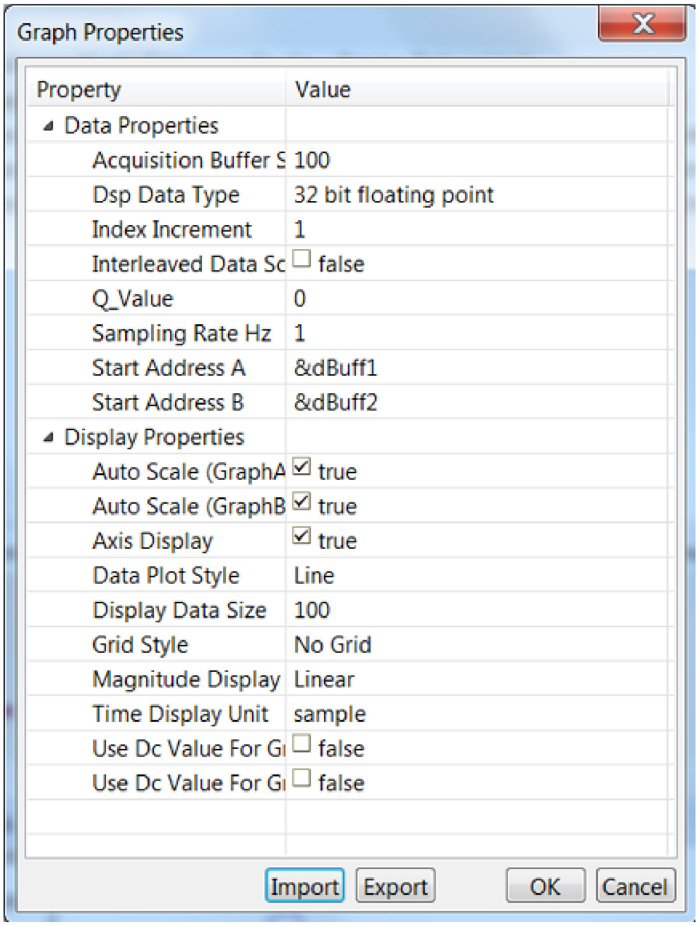
図 21. Build Level 1 Expressions View



Expression	Type	Value	Address
VIENNA_buildInfo	enum VIENNA_BuildLevel_enum	BuildLevel_1_OpenLoop	0x0000800B@Data
VIENNA_boardStatus	enum VIENNA_boardStatus_enum	boardStatus_NoFault	0x0000800D@Data
VIENNA_clearTrip	int	0	0x00008006@Data
VIENNA_dutyPU_DC	float	0.5	0x000080BA@Data
EPwm1Regs.TZFLG	Register	0x0004	
EPwm2Regs.TZFLG	Register	0x0004	
EPwm3Regs.TZFLG	Register	0x0004	
VIENNA_guiVbus_Volts	float	3.16032982	0x0000806E@Data
VIENNA_guiVbusPM_Volts	float	0.910193861	0x0000806C@Data
VIENNA_guiVbusMN_Volts	float	2.24856186	0x0000806A@Data
VIENNA_guiVrms1_Volts	float	0.0	0x00008092@Data
VIENNA_guiVrms2_Volts	float	0.0	0x00008094@Data
VIENNA_guiVrms3_Volts	float	0.0	0x00008096@Data
VIENNA_guiIrms1_Amps	float	0.0	0x0000808C@Data
VIENNA_guiIrms2_Amps	float	0.0	0x0000808E@Data
VIENNA_guiIrms3_Amps	float	0.0	0x00008090@Data
VIENNA_guiPF1	float	0.0	0x00008098@Data
VIENNA_guiPF2	float	0.0	0x0000809A@Data
VIENNA_guiPF3	float	0.0	0x0000809C@Data
Add new expression			

2. The current and voltage measurements can be verified by viewing the data in the graph window. These values are logged in the slower 10-kHz routine. Go to **Tools** > **Graph** > **DualTime**, and click **Import** and point to the **graph1.GraphProp** file inside the project folder. This file will populate the graph properties window. Alternatively, the user can enter the values as shown in the 図 22. When the entries are verified, click **OK**. Two graphs will appear in CCS. Click **Continuous Refresh** on these graphs. A second set of graphs can also be added by importing the **graph2.GraphProp** file.

図 22. Graph Settings



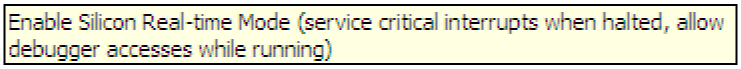
Graph Properties

Property	Value
Data Properties	
Acquisition Buffer S	100
Dsp Data Type	32 bit floating point
Index Increment	1
Interleaved Data Sc	☐ false
Q_Value	0
Sampling Rate Hz	1
Start Address A	&dBuff1
Start Address B	&dBuff2
Display Properties	
Auto Scale (GraphA	☑ true
Auto Scale (GraphB	☑ true
Axis Display	☑ true
Data Plot Style	Line
Display Data Size	100
Grid Style	No Grid
Magnitude Display	Linear
Time Display Unit	sample
Use Dc Value For Gı	☐ false
Use Dc Value For Gı	☐ false

6.4.1.4 Using Real-Time Emulation

Real-time emulation is a special emulation feature that allows windows within CCS to be updated while the MCU is running. This feature allows graphs and watch views to update 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.

1. Enable real-time mode by hovering the mouse on the buttons on the horizontal toolbar and clicking the button.



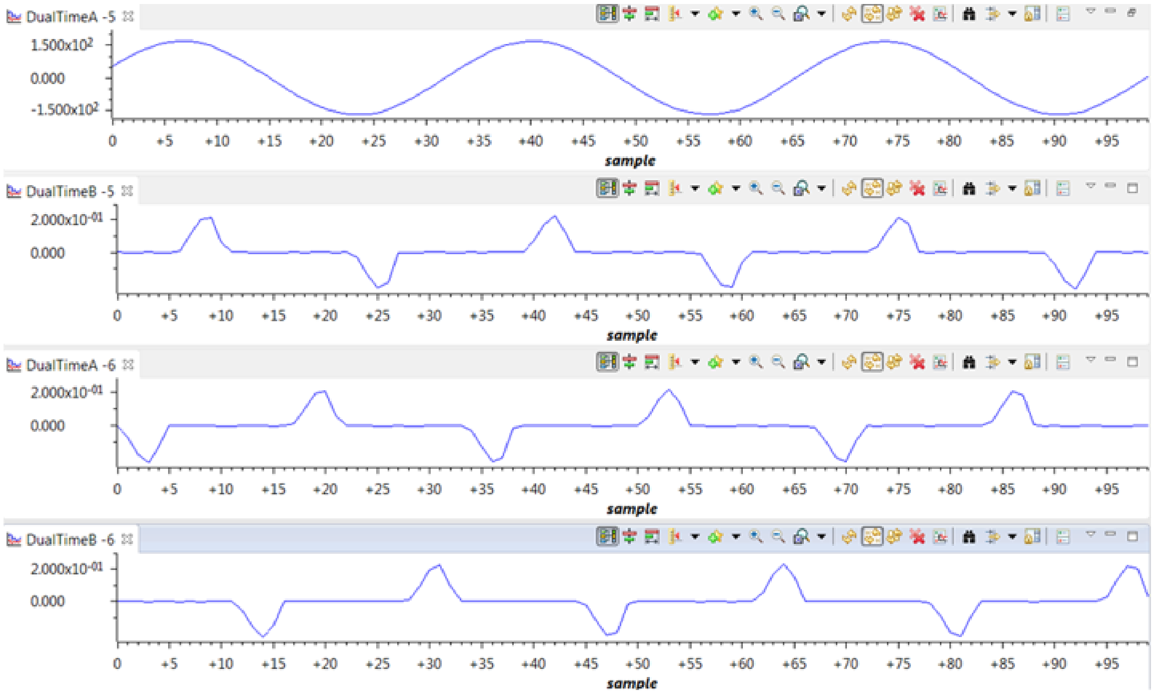
Enable Silicon Real-time Mode (service critical interrupts when halted, allow debugger accesses while running)

2. A message box may appear. If so, select **YES** to enable debug events. This will set bit 1 (DGBM bit) of status register 1 (ST1) to a **0**. The 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.

6.4.1.5 Running Code (Build 1)

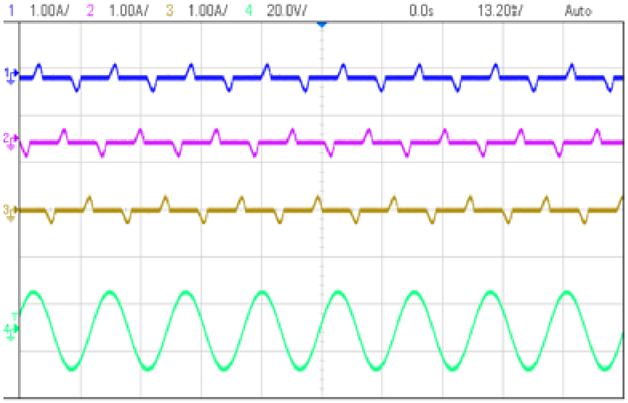
1. Run the project by clicking on ▶.
2. In the watch view, periodically check if the **guiVbus(VIENNA_guiVbus_Volts in the expression window)** variable is updating. If there is no change in the value then make sure the real-time mode is enabled, and the HW is setup correctly. Do not proceed further unless the update is verified.
   - Note: As no power is applied right now, this value will be close to zero.
3. Now slowly increase the input AC voltage from 0 to 120-Vrms L-N.
4. Verifying the voltage sensing: Make sure **guiVbus**, **guiVbusPM**, and **guiVbusMN** display the correct values. For the 120-Vrms L-N **guiVbus** will be close to 320 V, and the **guiVbusPM/MN** will be close to 160 V each. The code runs a sine analyzer module, which computes the RMS value of the voltage and current. Notice the value of **guiVrms1/2/3** will be close to the input value, that is, 120Vrms. This verifies the voltage sensing of the board.
5. Verifying the current sensing: Notice the**guiIrms1/2/3**, for the given test condition this value will be close to 0.92 Amps. Additionally, the graphs must be seen to verify the current measurement. Below the test condition they will look as shown in 図 23.

図 23. Build Level 1: Graph1.GraphProp and Graph2.GraphProp Files Showing Measured Voltage and Currents



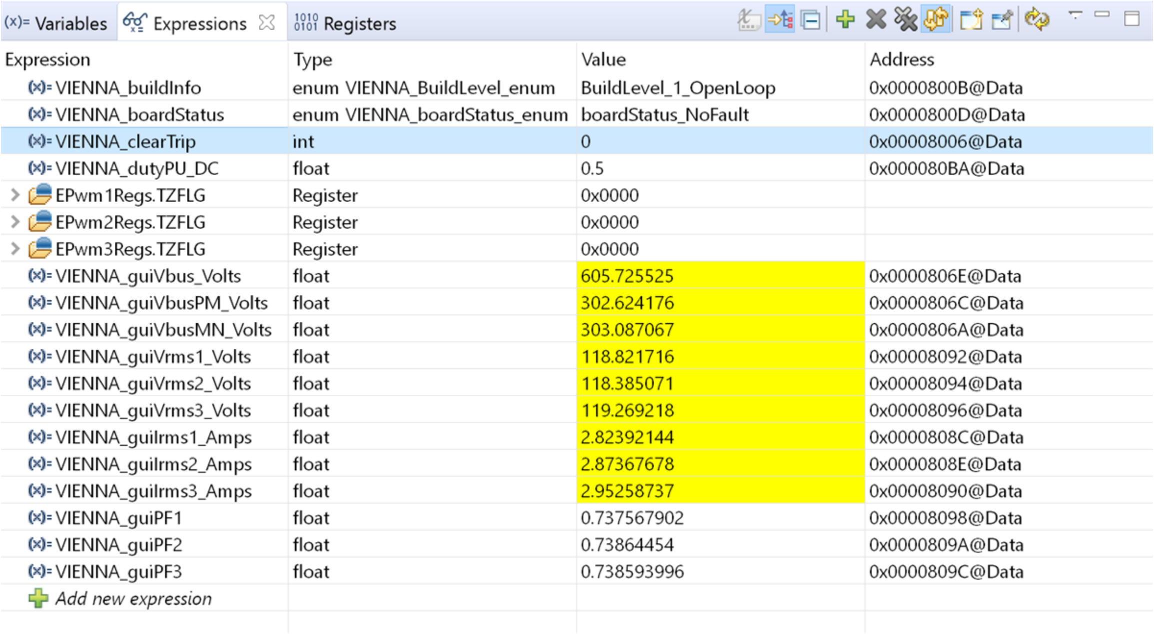
6. The scope capture of the input voltage and currents is shown in 図 24.

図 24. Build Level 1: Scope Capture IL1, IL2, IL3 and V1 (120Vrms L-N) With PWM Tripped



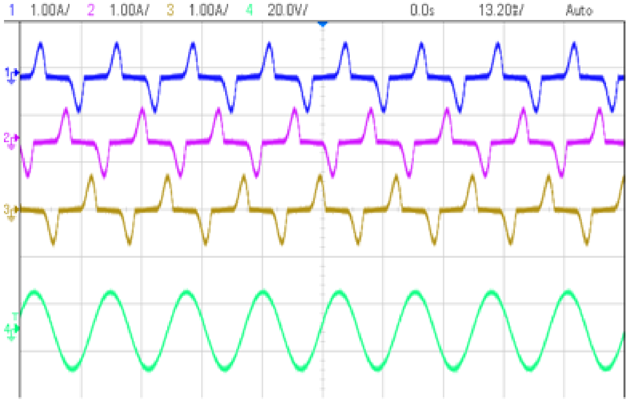
7. Next to verify the PWM action, first reduce the input voltage to zero and wait for all the voltages to go down to zero.
8. Now set the **dutyPU_DC** to 0.5 in the expressions view.
9. Next clear the PWM trip by writing a **1** to **clearTrip**.
10. Slowly increase the input voltage, and keep watch on the input current. The duty cycle will impart a boost action. For example, when Vac is 30 Vrms without switching enabled, the guiVbus will be ~84 V; with switching, the guiVbus will rise up to 140 V. Thus, **guiVbusPM/MN** will also be higher than the input voltage maximum.
11. Below the test conditions described in this build, the **guiVbus** will rise to about 600 V and **guiVbusPM/MN** will be close to 300 V each, and the current will be close to 2.8 Vrms when the input voltage reached 120 Vrms L-N. Expressions view will appear as shown in 図 25. Make sure all the variables are accurate, that is, guiVrms1/2/3, guiIrms1/2/3, guiPF1/2/3. If any variable is not in line with as shown in 図 25, it points to a hardware issue with the sensing circuit. The scope capture is shown in 図 26.

図 25. Build Level 1: Expressions View With Power Measurement



Expression	Type	Value	Address
VIENNA_buildInfo	enum VIENNA_BuildLevel_enum	BuildLevel_1_OpenLoop	0x0000800B@Data
VIENNA_boardStatus	enum VIENNA_boardStatus_enum	boardStatus_NoFault	0x0000800D@Data
VIENNA_clearTrip	int	0	0x00008006@Data
VIENNA_dutyPU_DC	float	0.5	0x000080BA@Data
EPwm1Regs.TZFLG	Register	0x0000	
EPwm2Regs.TZFLG	Register	0x0000	
EPwm3Regs.TZFLG	Register	0x0000	
VIENNA_guiVbus_Volts	float	605.725525	0x0000806E@Data
VIENNA_guiVbusPM_Volts	float	302.624176	0x0000806C@Data
VIENNA_guiVbusMN_Volts	float	303.087067	0x0000806A@Data
VIENNA_guiVrms1_Volts	float	118.821716	0x00008092@Data
VIENNA_guiVrms2_Volts	float	118.385071	0x00008094@Data
VIENNA_guiVrms3_Volts	float	119.269218	0x00008096@Data
VIENNA_guiIrms1_Amps	float	2.82392144	0x0000808C@Data
VIENNA_guiIrms2_Amps	float	2.87367678	0x0000808E@Data
VIENNA_guiIrms3_Amps	float	2.95258737	0x00008090@Data
VIENNA_guiPF1	float	0.737567902	0x00008098@Data
VIENNA_guiPF2	float	0.73864454	0x0000809A@Data
VIENNA_guiPF3	float	0.738593996	0x0000809C@Data
Add new expression			

図 26. Build Level 1: Scope Capture IL1, IL2, IL3 and V1 (120-Vrms L-N) With Duty Cycle 0.5



12. This check verifies at a basic level the PWM driver and connection on hardware.
13. Reduce the input voltage to zero, and watch for the bus voltages to reduce down to zero.
14. This completes the check for this build, the following items are verified on successful completion of this build:
    1. Sensing of voltages and currents and scaling to be correct
    2. Interrupt generation and execution of the build 1 code in the controlISR and tenkHzISR()
    3. PWM driver and switching

    If any issue is observed a careful inspection of the hardware may be required to eliminate any build issues and so forth.
15. The controller can now be halted, and the debug connection terminated.
16. Fully halting the MCU when in real-time mode is a two-step process. First halt the processor by using the **Halt** button on the toolbar (⏸) or by using **Target** > **Halt**. Then take the MCU out of real-time mode by clicking on . Finally, reset the MCU by clicking on .
17. Close CCS debug session by clicking on **Terminate Debug Session** (**Target** > **Terminate all**).

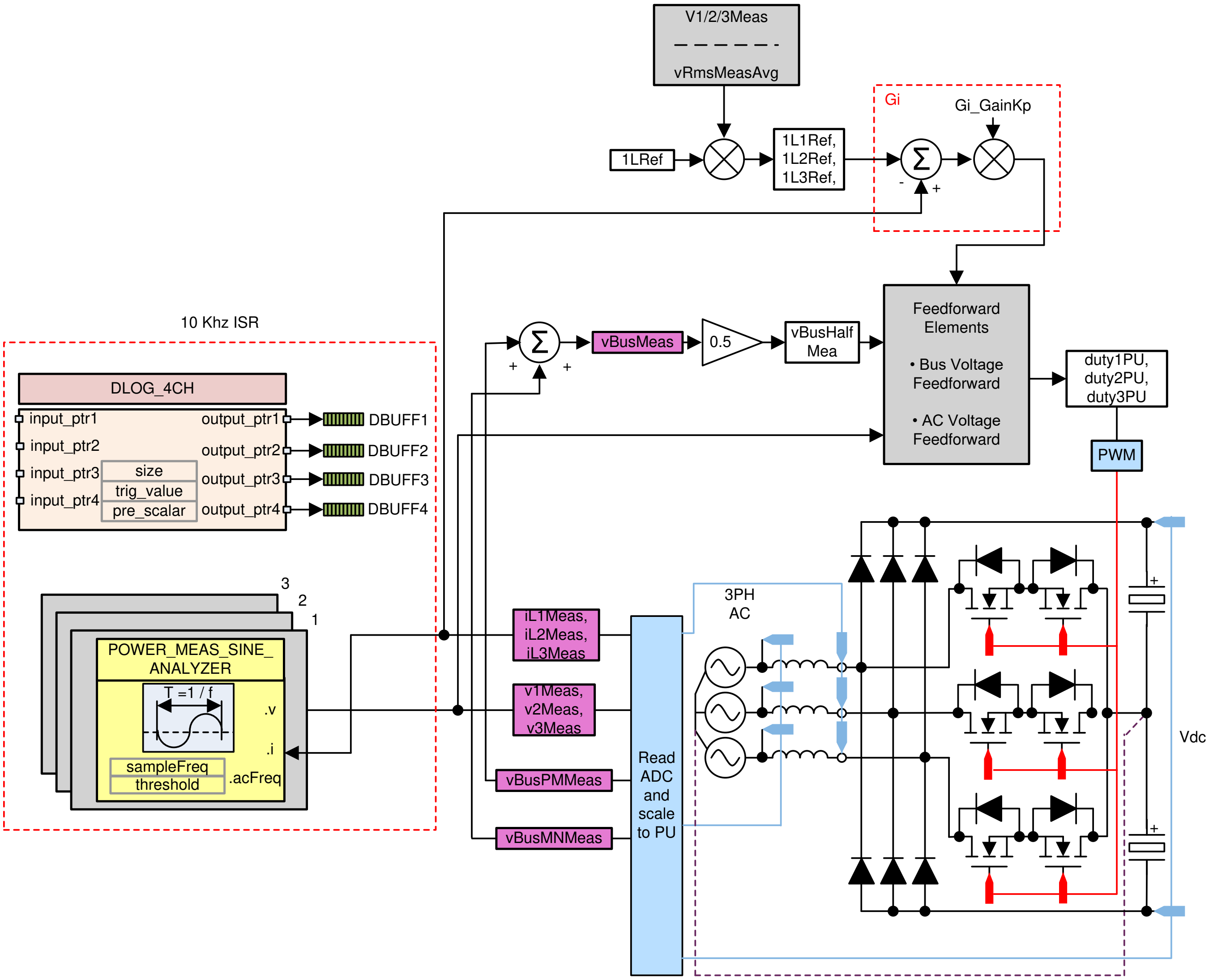
6.4.2 INCR_BUILD 2: Closed Current Loop

In this build, BUILD 2, the inner current loop is closed, that is, the inductor current is controlled using a current compensator Gi. Both DC bus and output voltage feedforward are applied to the output of this current compensator to generate the duty cycle of the inverter, as shown in 式 13. This action makes the plant for the current compensator simple, and a proportional (P) controller can be used to tune the loop of the inner current. The model for the current loop was derived in 3.2.

$$\mathrm{duty1PU} = \frac{(\mathrm{iL1Meas} - \mathrm{iL1Ref}) * \mathrm{Gi\_GainKp} + \mathrm{v1Meas}}{\mathrm{vBusHalfMeas}} \tag{13}$$

Complete software diagram for this build as illustrated in 図 27.

図 27. Build Level 2 Control Software Diagram: Closed Current Loop



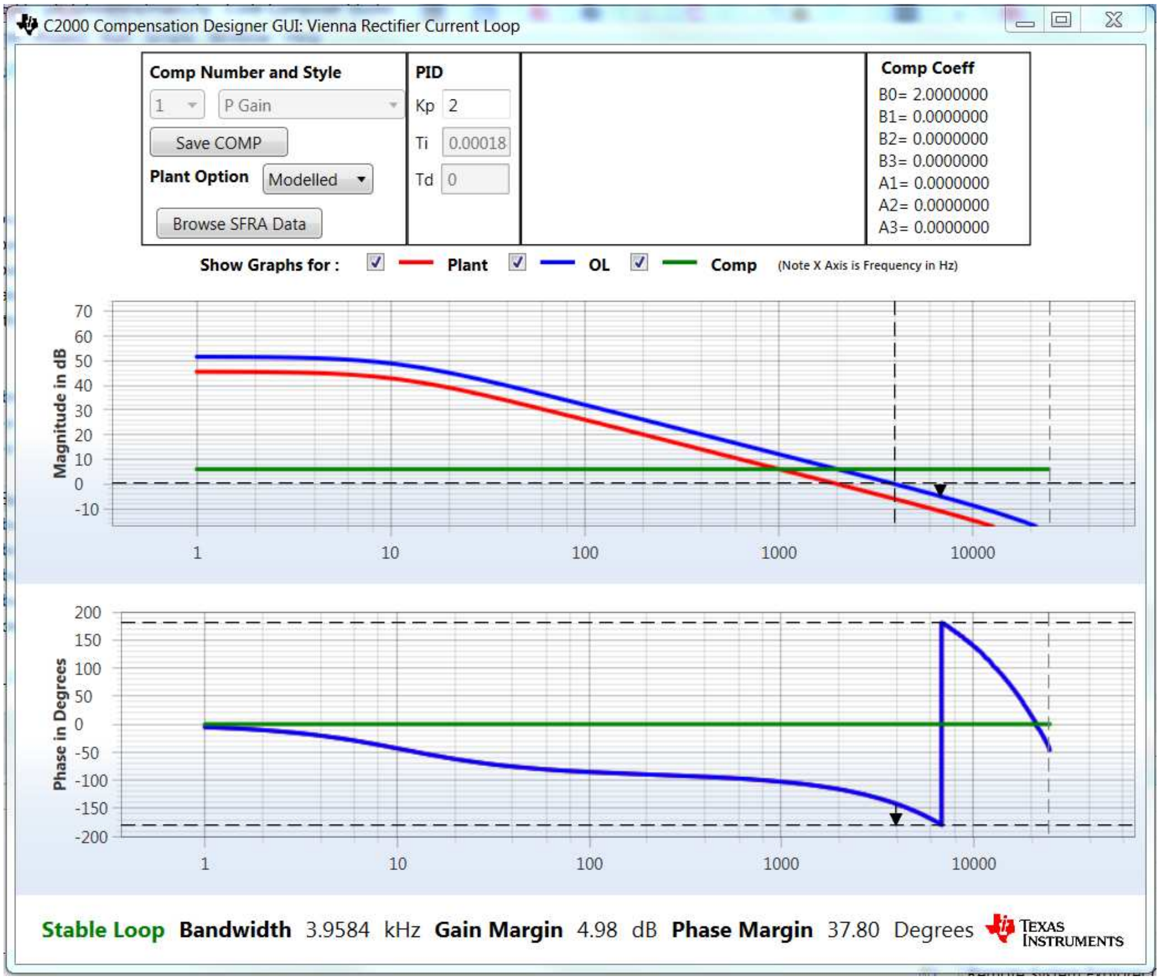
6.4.2.1 Setting Software Options for BUILD 2

1. Make sure the hardware is setup as outlined in 5.1. Do not supply any HV power yet to the board. For this build, connect the neutral from the three-phase power supply to the board, as shown in 図 16.
2. powerSUITE Settings : On the **powerSUITE** page below **Project Options** section, select **Closed Current Loop** for the build level. Save the page.
3. Below **Control Loop Design**, options for the current loop tuning will automatically be selected (**Tuning** > **Current Loop** > **COMP1** > **P**). Now click on the **Compensation Designer** icon ().

6.4.2.2 Designing Current Loop Compensator

1. Compensation Designer will then launch, with the model of the current loop plant with parameters specified on the powerSUITE page. Proportional gain value can then be changed to ensure stable closed loop operation. Stability of the system when using the designed compensator can be verified by observing the gain and phase margins on the open loop transfer function plot in the Compensation Designer, as shown in 図 28.

図 28. Current Loop Design Using Compensation Designer



C2000 Compensation Designer GUI: Vienna Rectifier Current Loop

Comp Number and Style: 1, P Gain

Save COMP

Plant Option: Modelled

Browse SFRA Data

PID
Kp 2
Ti 0.00018
Td 0

Comp Coeff
B0= 2.0000000
B1= 0.0000000
B2= 0.0000000
B3= 0.0000000
A1= 0.0000000
A2= 0.0000000
A3= 0.0000000

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

Stable Loop Bandwidth 3.9584 kHz Gain Margin 4.98 dB Phase Margin 37.80 Degrees

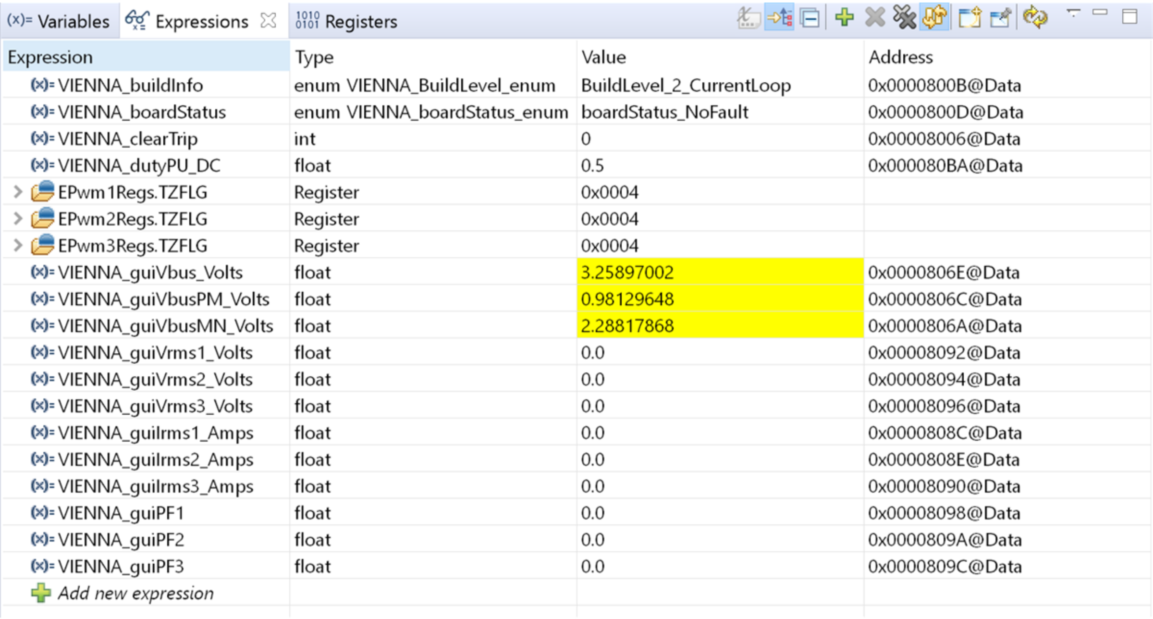
2. When satisfied with the proportional gain, click **Save COMP**. This will save the compensator values into the project.
   - Note: If the project was not selected from the solution adapter, changes to the compensator will not be allowed to design one's own solution. Select the solution through the solution adapter.
3. Close the Compensation Designer, and return to the powerSUITE page.

   Note: The P gain in the code is set to 2.0, this gain would give a OL plot with margins that may look too small; however, this gain has been adjusted keeping in mind the actual measurements from the design where the measured gain is slightly lower. Therefore, a higher gain to the compensator can be applied without sacrificing the actual margins.

6.4.2.3 Building and Loading Project and Setting up Debug

1. Right click on the project name, and click *Rebuild Project*. The project will build successfully. Click **Run** > **Debug**, which will launch a debugging session. In the case of dual CPU devices, a window may appear to select the CPU the debug needs to be performed. In this case, select CPU1. The project will then load on the device, and CCS debug view will become active. The code will halt at the start of the main routine.
2. To add the variables in the watch and expressions window click **View** > **Scripting Console** to open the scripting console dialog box. On the upper right corner of this console, click on **Open** to browse to the **setupdebugenv_build2.js** script file, which is located inside the project folder. This file will populate the watch window with appropriate variables needed to debug the system. Click **Continuous Refresh** button () on the watch window to enable continuous update of values from the controller. The watch window will appear as 図 29.

図 29. Build Level 2: Closed Current Loop Expressions View



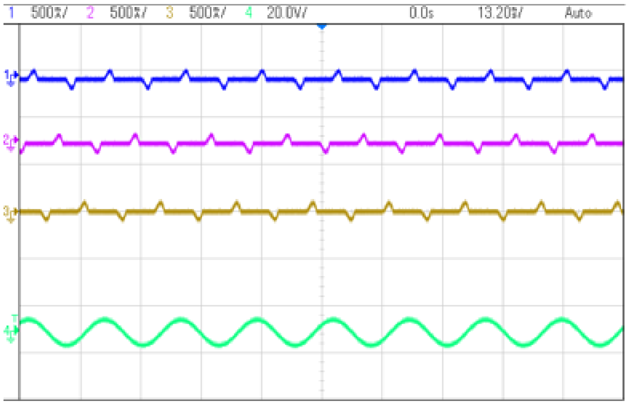
Expression	Type	Value	Address
VIENNA_buildInfo	enum VIENNA_BuildLevel_enum	BuildLevel_2_CurrentLoop	0x0000800B@Data
VIENNA_boardStatus	enum VIENNA_boardStatus_enum	boardStatus_NoFault	0x0000800D@Data
VIENNA_clearTrip	int	0	0x00008006@Data
VIENNA_dutyPU_DC	float	0.5	0x000080BA@Data
EPwm1Regs.TZFLG	Register	0x0004	
EPwm2Regs.TZFLG	Register	0x0004	
EPwm3Regs.TZFLG	Register	0x0004	
VIENNA_guiVbus_Volts	float	3.25897002	0x0000806E@Data
VIENNA_guiVbusPM_Volts	float	0.98129648	0x0000806C@Data
VIENNA_guiVbusMN_Volts	float	2.28817868	0x0000806A@Data
VIENNA_guiVrms1_Volts	float	0.0	0x00008092@Data
VIENNA_guiVrms2_Volts	float	0.0	0x00008094@Data
VIENNA_guiVrms3_Volts	float	0.0	0x00008096@Data
VIENNA_guiIrms1_Amps	float	0.0	0x0000808C@Data
VIENNA_guiIrms2_Amps	float	0.0	0x0000808E@Data
VIENNA_guiIrms3_Amps	float	0.0	0x00008090@Data
VIENNA_guiPF1	float	0.0	0x00008098@Data
VIENNA_guiPF2	float	0.0	0x0000809A@Data
VIENNA_guiPF3	float	0.0	0x0000809C@Data

3. Enable real-time mode by hovering the mouse on the buttons on the horizontal toolbar, and clicking the button.

6.4.2.4 Running Code (Build 2)

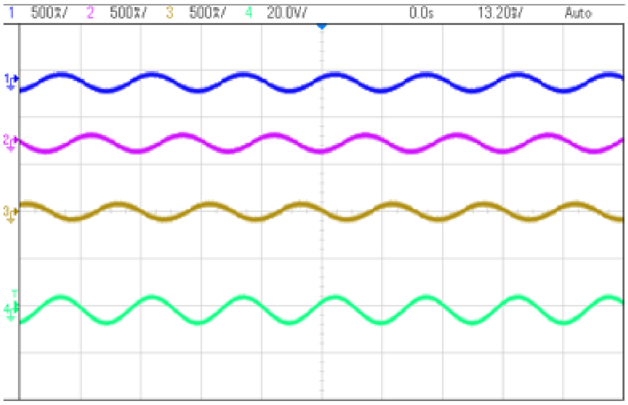
1. Run the project by clicking ▶.
2. It is advisable to test first at a low voltage. Therefore, the input AC voltage is raised to only 40 Vrms, 60 Hz.
3. The input current and voltage will look like 図 30.

図 30. Build Level 2: Scope Capture IL1, IL2, IL3 and V1 (40Vrms L-N) With PWM Tripped



4. A current reference is set by changing the iLRef variable in the expressions view. This variable is set to 0.05.
5. Clear the trip by setting the **clearTrip** variable to **1**.
6. As soon as the trip is cleared, a sinusoidal current will be seen to be drawn from the input, which verifies correct operation of the current loop. The waveform will look like 図 31.

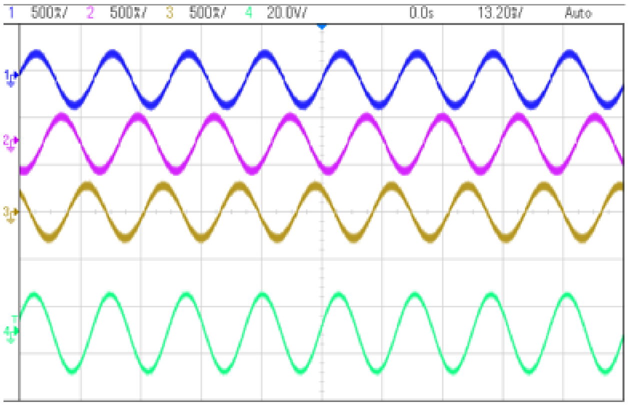
図 31. Build Level 2: Scope Capture IL1, IL2, IL3 and V1 (40Vrms L-N) With Current Loop Closed



7. The guiVbus will be close to 190 V, and the input AC current per phase will be close to 0.6 Amps
8. Raise the current reference iLRef to 0.1. Observe the bus voltage to go to 266 V and the input current to around 1.2 Amps.
9. Now raise the input AC voltage slowly to 120 Vrms. The board will maintain the input current to be constant as the input voltage rises.

10. When the input has stabilized to 120 Vrms, raise the iLRef so the guiVbus is 600 V; for the test conditions specified, this will correspond to iLRef = 0.165. The current waveforms will look like 図 32.

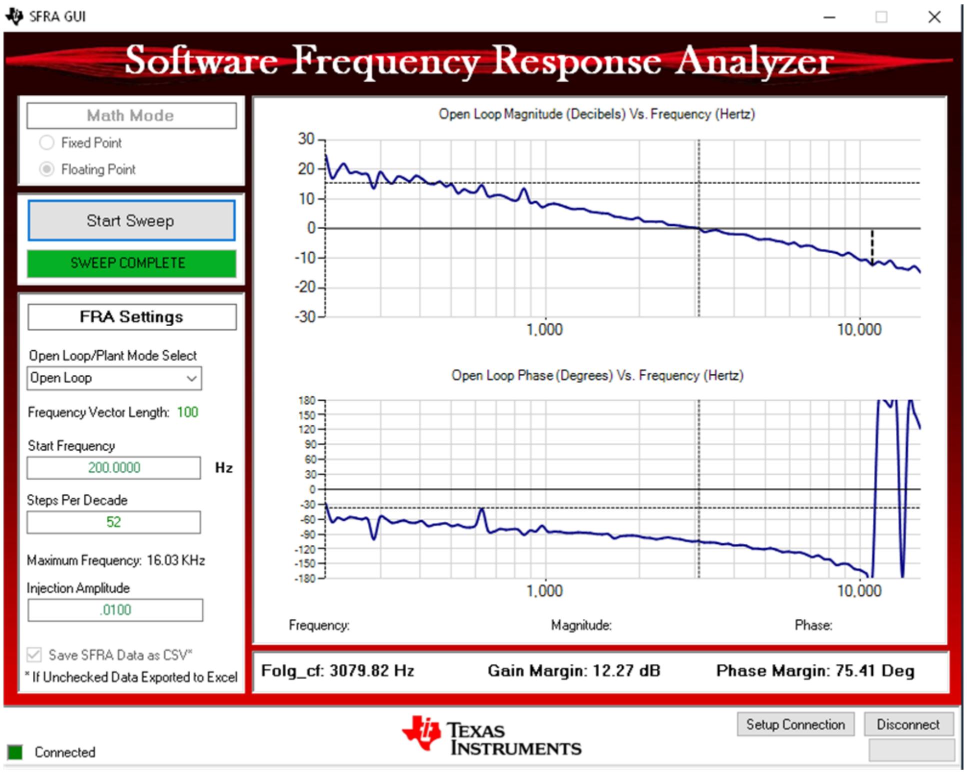
図 32. Build Level 2: Scope Capture IL1, IL2, IL3 and V1 (120Vrms L-N) With Current Loop Closed



11. As only a proportional gain is used in the compensator, the current reference minus the feedback error is never zero. Notice the current drawn to deviate slightly from the reference.
12. SFRA is integrated in the software of this build 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 syscfg page, click on the **SFRA** icon. SFRA GUI will pop-up.
13. Select the options for the device on the SFRA GUI. For example, for F28379D select floating point. Click **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**.

14. The SFRA GUI will connect to the device. A SFRA sweep can now be started by clicking **Start Sweep**. The complete SFRA sweep will take 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. When complete, a graph with the open loop plot will appear, as in 図 33. This verifies that the designed compensator is indeed stable.

図 33. SFRA Run on Closed Current Loop



○ Fixed Point
● Floating Point

Open Loop/Plant Mode Select: Open Loop

Frequency Vector Length: 100

Start Frequency: 200.0000 Hz

Steps Per Decade: 52

Maximum Frequency: 16.03 KHz

Injection Amplitude: .0100

☑ Save SFRA Data as CSV*

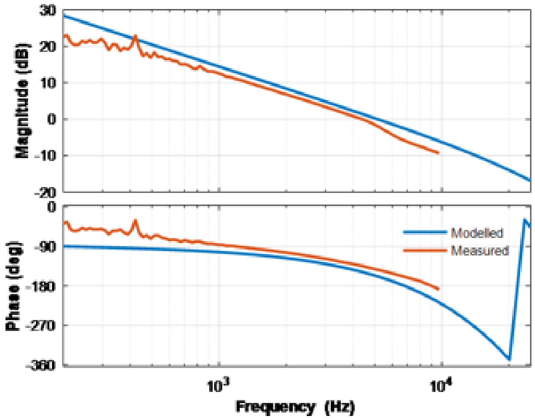
* If Unchecked Data Exported to Excel

Folg_cf: 3079.82 Hz   Gain Margin: 12.27 dB   Phase Margin: 75.41 Deg

🟩 Connected

The frequency response data is also saved in the project folder below 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 modelled values (see 図 34).

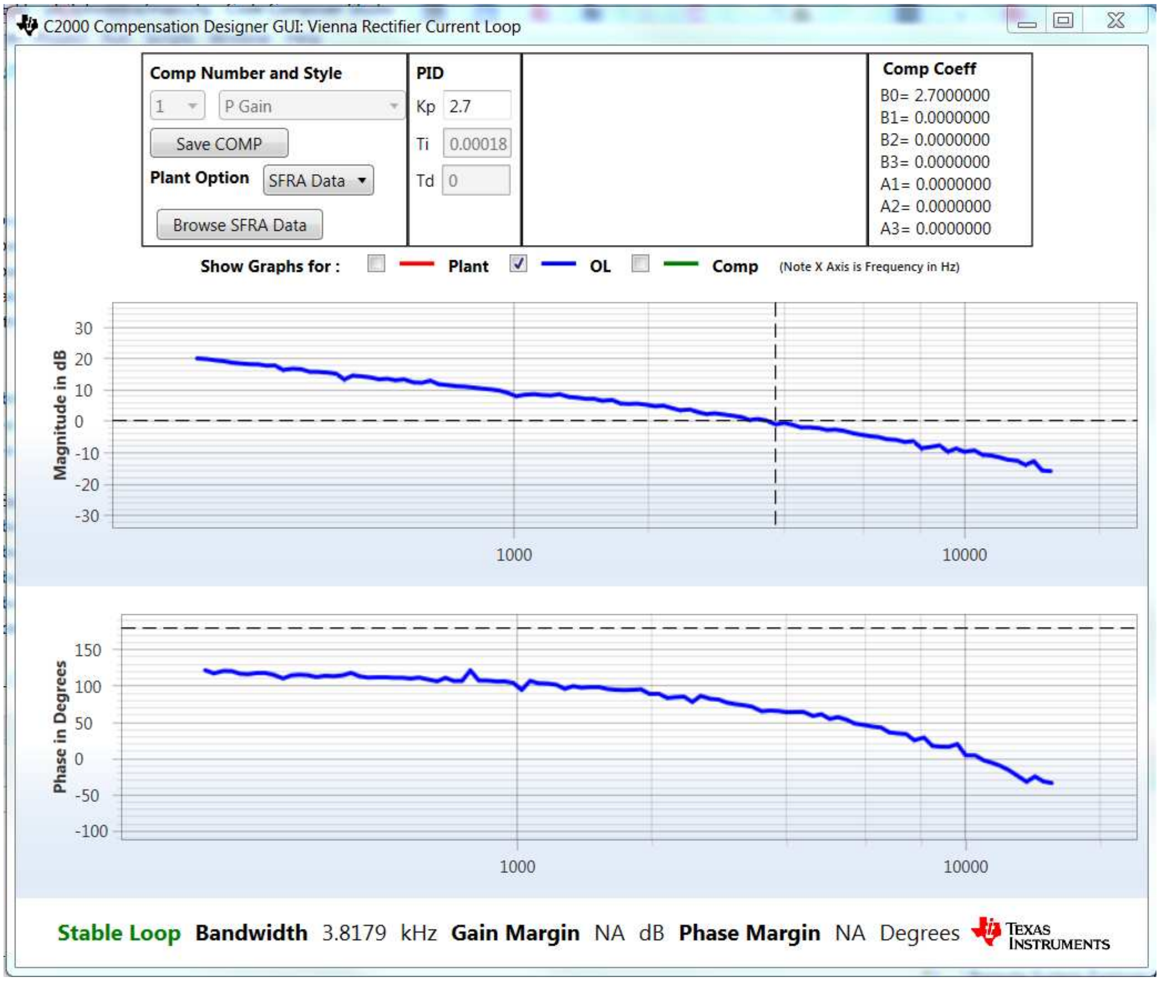
図 34. Modelled Versus Measured OL Response for Current Loop



15. Click the Compensation Designer again from the SYSCFG page, and choose **SFRA Data** for plant option on the GUI. This option will use the measured plant information to design the compensator and can be used to fine tune the compensation. By default the Compensation Designer will point to the latest SFRA run and show up as 図 35. If a previous SFRA run plant information needs to be used, the user can select the *SFRAData.csv* file by browsing to it by clicking on **Browse SFRA Data**. Close

Compensation Designer to return to the syscfg page when this is done.

図 35. Compensation Designer With Measured Plant Frequency Response Data



C2000 Compensation Designer GUI: Vienna Rectifier Current Loop

Comp Number and Style: 1, P Gain

Save COMP

Plant Option: SFRA Data

Browse SFRA Data

PID

Kp 2.7

Ti 0.00018

Td 0

Comp Coeff

B0= 2.7000000
B1= 0.0000000
B2= 0.0000000
B3= 0.0000000
A1= 0.0000000
A2= 0.0000000
A3= 0.0000000

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

Stable Loop Bandwidth 3.8179 kHz Gain Margin NA dB Phase Margin NA Degrees

16. This action verifies the current compensator design. Note the phase if off because there is a negative sign in the control loop.
17. To bring the system to a safe stop, bring the input AC voltage down to zero, and observe the guiVBus will come down to zero as well.
18. Fully halting the MCU when in real-time mode is a two-step process. First halt the processor by using the **Halt** button on the toolbar (⏸) or by using **Target** > **Halt**. Then take the MCU out of real-time mode by clicking on . Finally, reset the MCU ( ) .
19. Close the CCS debug session by clicking on *Terminate Debug Session* (**Target** > **Terminate all**).

6.4.3 INCR_BUILD 3: Closed Voltage and Current Loop

In this build the outer voltage loop is closed with the inner current loop closed (designed in BUILD 2). The model of the outer voltage loop was derived in 図 36. A PI-based compensator is used and tuned through the Compensation Designer. 図 36 shows the software diagram for this build.

図 36. Build Level 3 Control Diagram: Output Voltage Control With Inner Current Loop

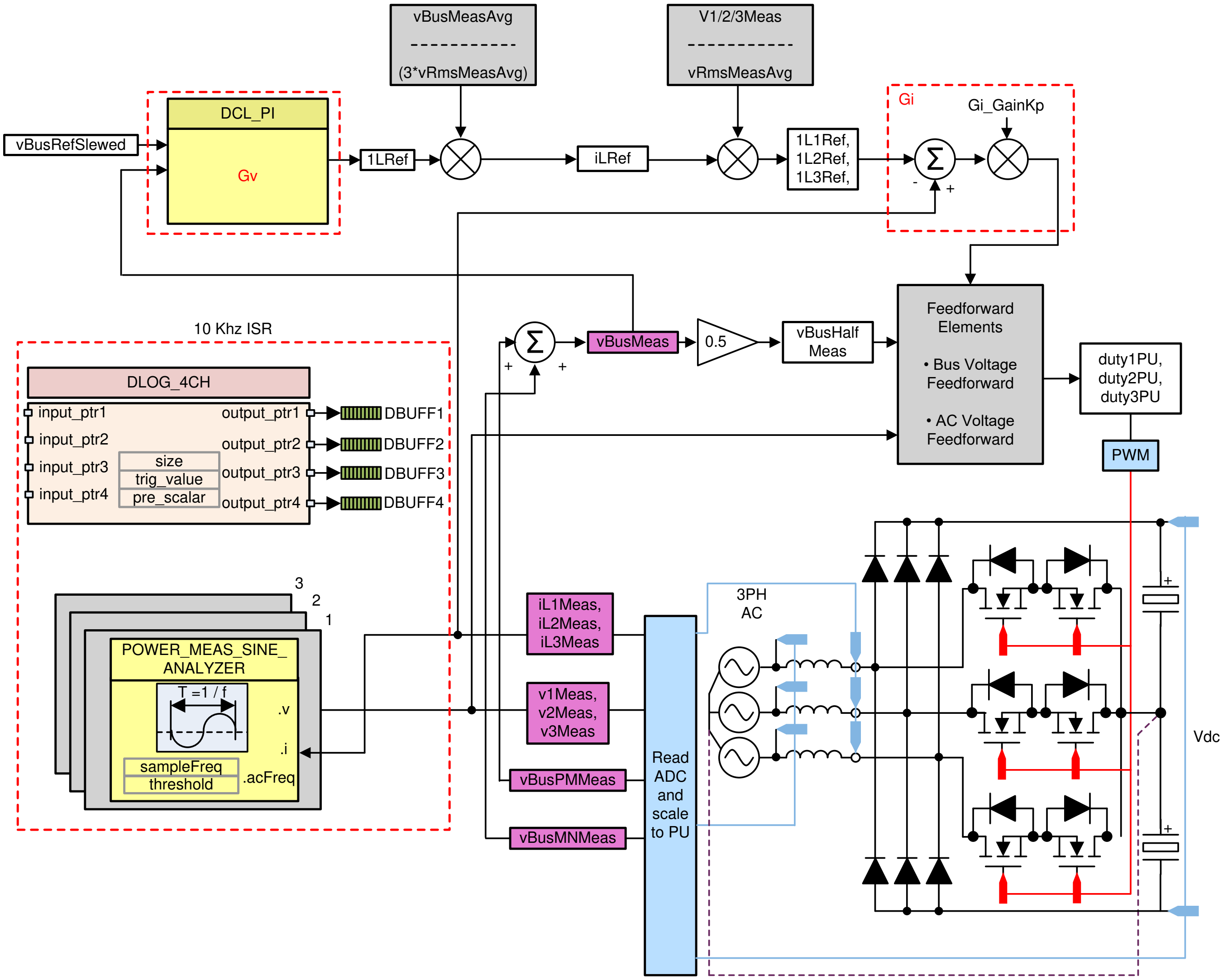


図 37. Build Level 3 Control Diagram: Output Voltage Control With Inner Current Loop

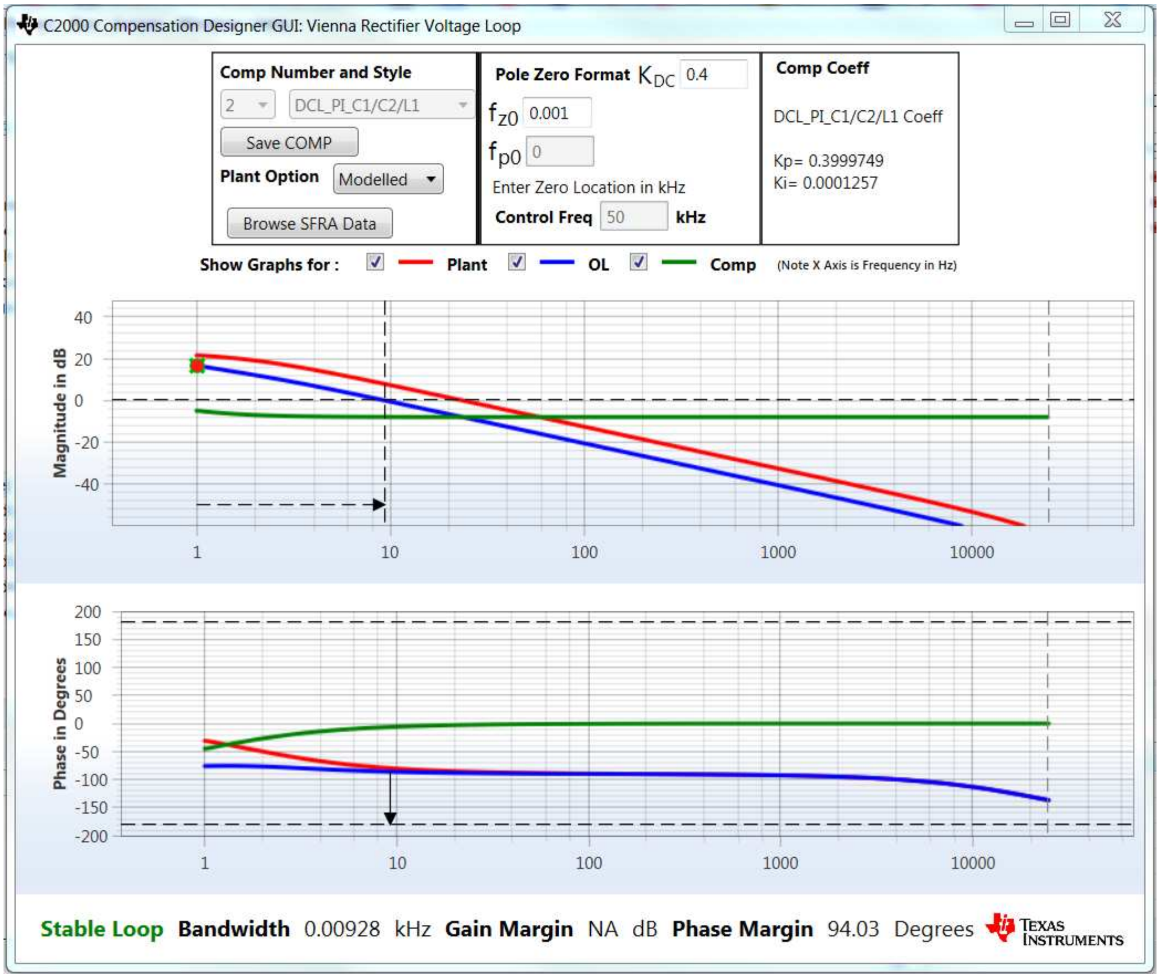
6.4.3.1 Setting Software Options for BUILD 3

1. Make sure the hardware is setup as outlined in 図 16. Do not supply any HV power yet to the board. For this build connect the neutral from the three-phase power supply to the board 図 16.
2. Below **Project Options**, select **Closed Voltage & Current Loop**.
3. Below **Control Loop Design**, select **Tuning** as **Voltage Loop**. Style will be preset to **PI**. Save the page by **Ctrl** + **S**, and click the **Compensation Designer** button ().
4. Make sure the load connected at the output of the board is correctly entered on the powerSUITE syscfg page because this load value is used in the design of the voltage compensator.

6.4.3.2 Designing Voltage Loop Compensator

1. Compensation designer will then launch with the model of the voltage loop plant, as shown in 図 38. The PI compensator can be edited to get the desired gain and phase margin, keeping in mind the bandwidth of the voltage loop has an inverse relationship with the THD achieved. Typically in a PFC application, this bandwidth is kept at ~10 Hz.

図 38. Voltage Loop PI Compensation Tuning Using Compensation Designer



C2000 Compensation Designer GUI: Vienna Rectifier Voltage Loop

Comp Number and Style: 2, DCL_PI_C1/C2/L1

Save COMP

Plant Option: Modelled

Browse SFRA Data

Pole Zero Format K_{DC}: 0.4

f_{z0}: 0.001

f_{p0}: 0

Enter Zero Location in kHz

Control Freq: 50 **kHz**

Comp Coeff

DCL_PI_C1/C2/L1 Coeff

Kp= 0.3999749

Ki= 0.0001257

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

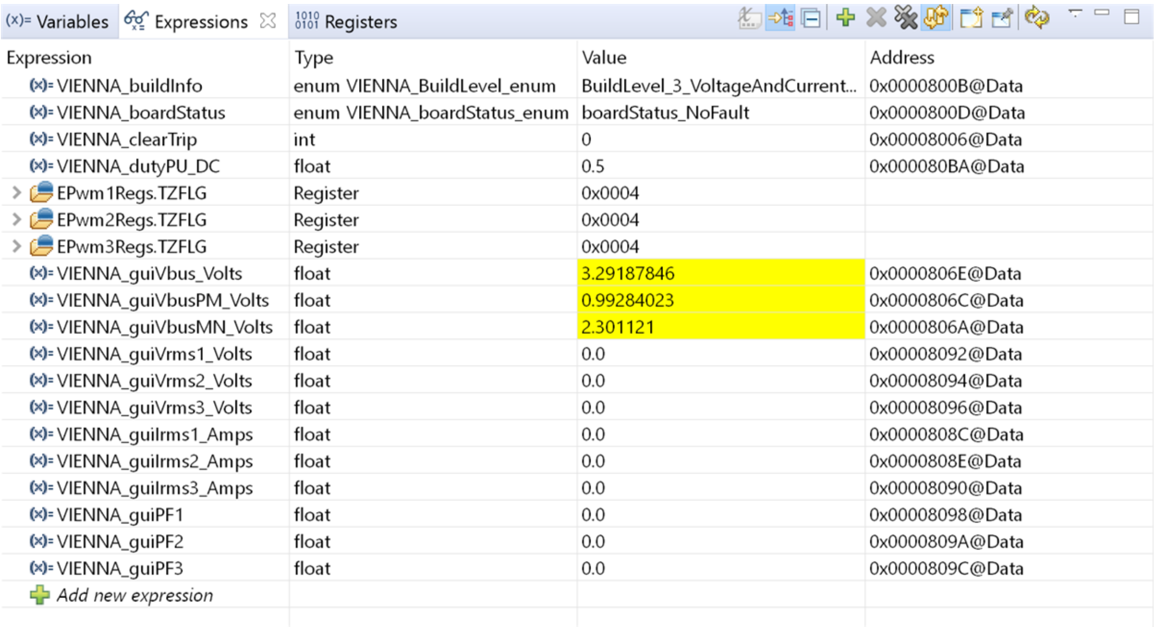
Stable Loop **Bandwidth** 0.00928 kHz **Gain Margin** NA dB **Phase Margin** 94.03 Degrees

2. When satisfied with the compensator design, click **Save COMP**. This will save the compensator values into the project.
   - Note: If the project was not selected from the solution adapter, changes to the compensator will not be allowed. To design one's own, select the solution through the solution adapter.
3. Close the Compensation Designer, and return to the powerSUITE page. Save using **Ctrl** + **S**.

6.4.3.3 Building and Loading the Project and Setting up Debug

1. Right click on the project name, and click **Rebuild Project**. The project will build successfully. Click **Run** > **Debug**, which will launch a debugging session. In the case of dual CPU devices, a window may appear to select the CPU the debug needs to be performed. In this case, select CPU1. The project will then load on the device, and CCS debug view will become active. The code will halt at the start of the main routine.
2. To add the variables in the watch and expressions window, click **View** > **Scripting Console** to open the scripting console dialog box. On the upper right corner of this console, click on **Open** to browse to the **setupdebugenv_build3.js** script file located inside the project folder. This file will populate the watch window with appropriate variables needed to debug the system. Click on the **Continuous Refresh** button ( ) on the watch window to enable continuous update of values from the controller. The watch window will appear as shown in 図 39.

図 39. Build Level 3: Expressions View



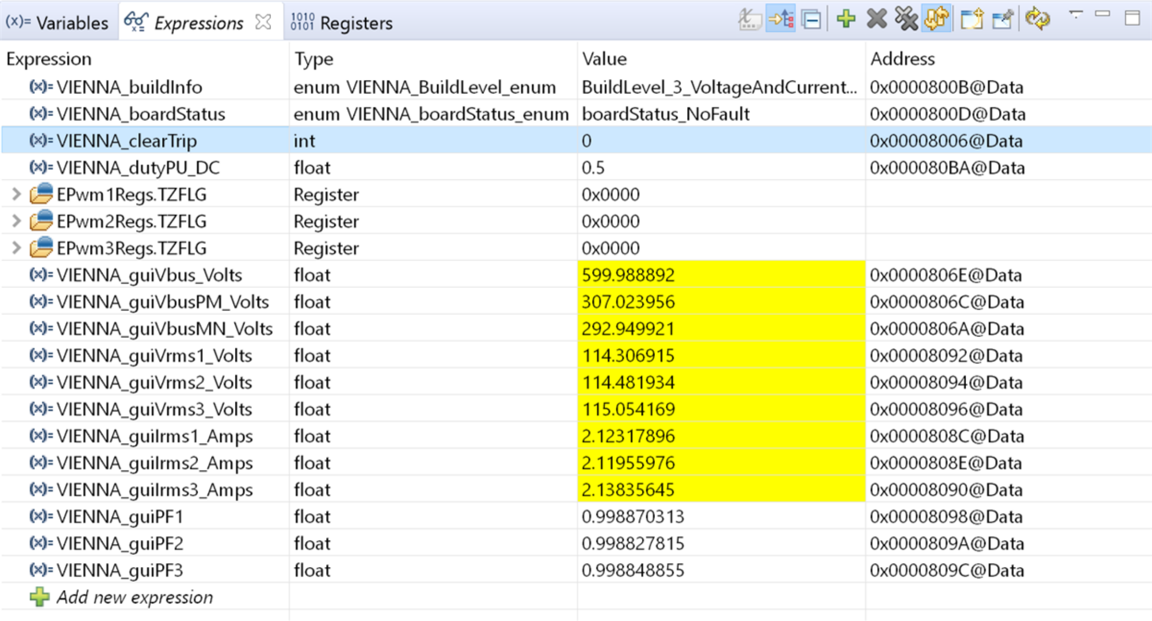
Expression	Type	Value	Address
VIENNA_buildInfo	enum VIENNA_BuildLevel_enum	BuildLevel_3_VoltageAndCurrent...	0x0000800B@Data
VIENNA_boardStatus	enum VIENNA_boardStatus_enum	boardStatus_NoFault	0x0000800D@Data
VIENNA_clearTrip	int	0	0x00008006@Data
VIENNA_dutyPU_DC	float	0.5	0x000080BA@Data
EPwm1Regs.TZFLG	Register	0x0004	
EPwm2Regs.TZFLG	Register	0x0004	
EPwm3Regs.TZFLG	Register	0x0004	
VIENNA_guiVbus_Volts	float	3.29187846	0x0000806E@Data
VIENNA_guiVbusPM_Volts	float	0.99284023	0x0000806C@Data
VIENNA_guiVbusMN_Volts	float	2.301121	0x0000806A@Data
VIENNA_guiVrms1_Volts	float	0.0	0x00008092@Data
VIENNA_guiVrms2_Volts	float	0.0	0x00008094@Data
VIENNA_guiVrms3_Volts	float	0.0	0x00008096@Data
VIENNA_guiIrms1_Amps	float	0.0	0x0000808C@Data
VIENNA_guiIrms2_Amps	float	0.0	0x0000808E@Data
VIENNA_guiIrms3_Amps	float	0.0	0x00008090@Data
VIENNA_guiPF1	float	0.0	0x00008098@Data
VIENNA_guiPF2	float	0.0	0x0000809A@Data
VIENNA_guiPF3	float	0.0	0x0000809C@Data
Add new expression			

3. Enable real-time mode by hovering the mouse on the buttons on the horizontal toolbar and clicking the button.

6.4.3.4 Running Code (Build 3)

1. Run the project by clicking ▶.
2. Raise the input AC voltage to 120-Vrms VL-N or 208-Vrms VL-L, 60 Hz.
3. The DC voltage reference is set by the variable vBusRef. This value is set to as 1.32, which corresponds to 600 V for this design. Refer to *calculations.xls* sheet for details.
4. Clear the trip by setting the **clearTrip** variable to **1**. The bus voltage will then rise to be 600 V.
5. Closed loop operation can be verified by comparing the **vBusRef** and **vBusMeas** in the expressions window as shown in 図 40.

図 40. Build Level 3: Expressions Window

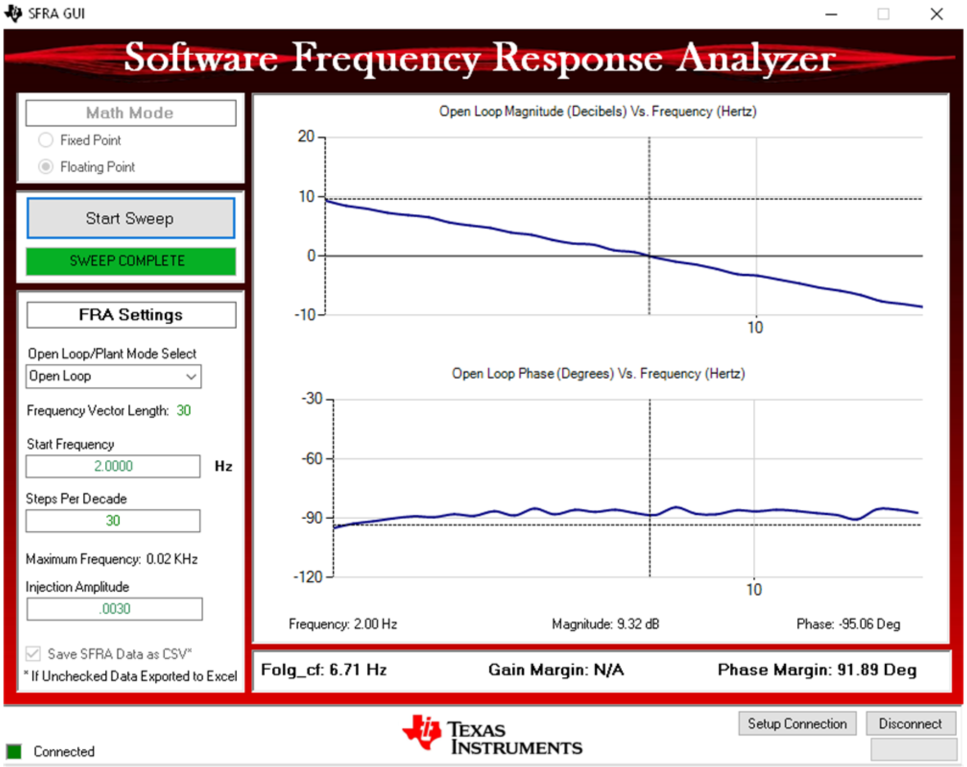


Expression	Type	Value	Address
VIENNA_buildInfo	enum VIENNA_BuildLevel_enum	BuildLevel_3_VoltageAndCurrent...	0x0000800B@Data
VIENNA_boardStatus	enum VIENNA_boardStatus_enum	boardStatus_NoFault	0x0000800D@Data
VIENNA_clearTrip	int	0	0x00008006@Data
VIENNA_dutyPU_DC	float	0.5	0x000080BA@Data
EPwm1Regs.TZFLG	Register	0x0000	
EPwm2Regs.TZFLG	Register	0x0000	
EPwm3Regs.TZFLG	Register	0x0000	
VIENNA_guiVbus_Volts	float	599.988892	0x0000806E@Data
VIENNA_guiVbusPM_Volts	float	307.023956	0x0000806C@Data
VIENNA_guiVbusMN_Volts	float	292.949921	0x0000806A@Data
VIENNA_guiVrms1_Volts	float	114.306915	0x00008092@Data
VIENNA_guiVrms2_Volts	float	114.481934	0x00008094@Data
VIENNA_guiVrms3_Volts	float	115.054169	0x00008096@Data
VIENNA_guiIrms1_Amps	float	2.12317896	0x0000808C@Data
VIENNA_guiIrms2_Amps	float	2.11955976	0x0000808E@Data
VIENNA_guiIrms3_Amps	float	2.13835645	0x00008090@Data
VIENNA_guiPF1	float	0.998870313	0x00008098@Data
VIENNA_guiPF2	float	0.998827815	0x0000809A@Data
VIENNA_guiPF3	float	0.998848855	0x0000809C@Data
Add new expression			

6. SFRA is integrated in the software of this build 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 syscfg page, click on the **SFRA** icon. SFRA GUI will pop up.
7. Select the options for the device on the SFRA GUI. For example, for F28379D, select floating point. Click on **Setup Connection**, and 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.

8. The SFRA GUI will connect to the device. A SFRA sweep can now be started by clicking **Start Sweep**. The complete SFRA sweep will take a few minutes to finish. Activity can be monitored by seeing the progress bar on the SFRA GUI and checking the flashing of blue LED on the back on the control card that indicates UART activity. When complete, a graph with the open loop plot will appear, as seen in 図 41. This action verifies that the designed compensator is indeed stable.

図 41. SFRA Run on Closed Voltage Loop



SFRA GUI

Software Frequency Response Analyzer

Math Mode
○ Fixed Point
● Floating Point

Start Sweep
SWEEP COMPLETE

FRA Settings
Open Loop/Plant Mode Select: Open Loop
Frequency Vector Length: 30
Start Frequency: 2.0000 Hz
Steps Per Decade: 30
Maximum Frequency: 0.02 KHz
Injection Amplitude: .0030
☑ Save SFRA Data as CSV*
* If Unchecked Data Exported to Excel

Frequency: 2.00 Hz Magnitude: 9.32 dB Phase: -95.06 Deg

Folg_cf: 6.71 Hz Gain Margin: N/A Phase Margin: 91.89 Deg

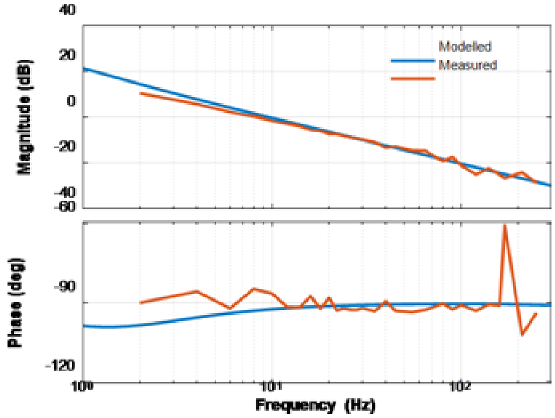
🟩 Connected

Open Loop Magnitude (Decibels) Vs. Frequency (Hertz)
20
10
0
-10
10
Open Loop Phase (Degrees) Vs. Frequency (Hertz)
-30
-60
-90
-120
10
Texas Instruments
Setup Connection
Disconnect

The frequency response data is also saved in the project folder below an SFRA data folder and is time stamped with the time of the SFRA run.

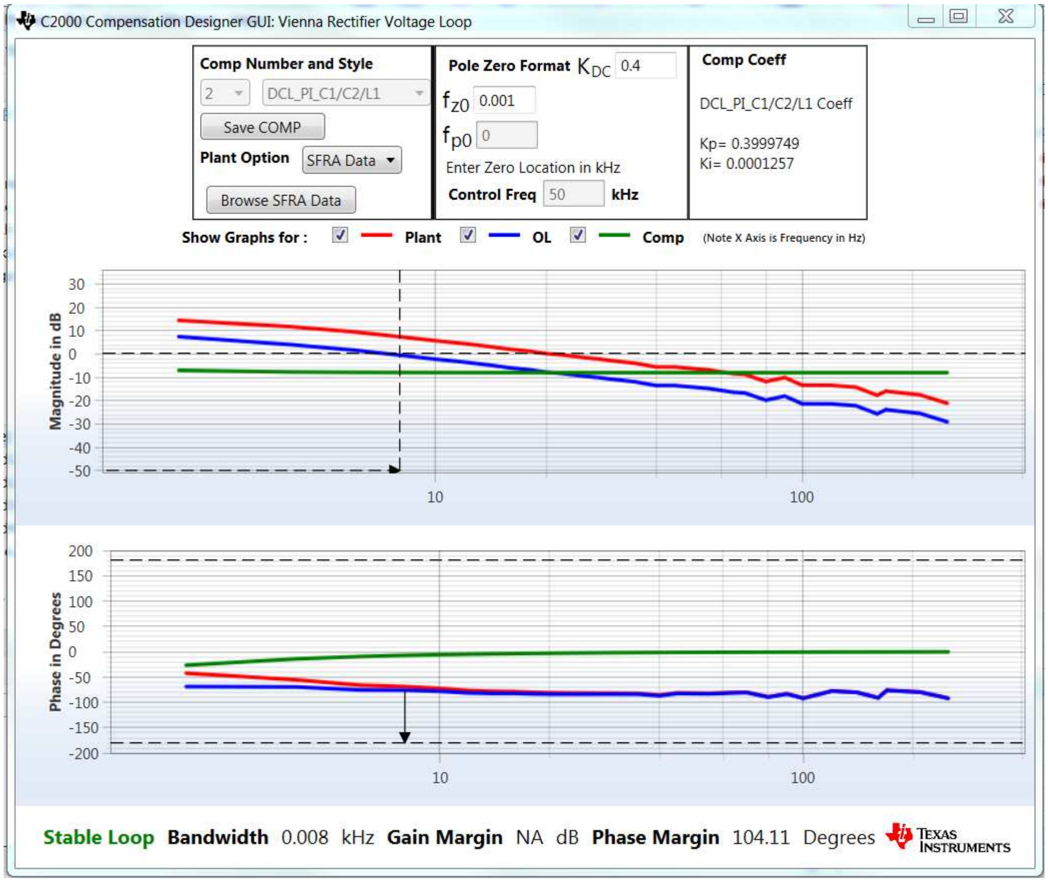
Note the measured gain and phase margin are close to the modelled values, as shown in 図 42.

図 42. Voltage Loop Modeled Versus Measured Comparison



9. Click the Compensation Designer again from the SYSCFG page, and choose **SFRA Data** for plant option on the GUI. This option uses the measured plant information to design the compensator, as shown in 図 43. This option can be used to fine tune the compensation. By default the Compensation Designer will point to the latest SFRA run. If a previous SFRA run plant information needs to be used, the user can select the **SFRAData.csv** file by browsing to it by clicking **Browse SFRA Data**. Close Compensation Designer to return to the syscfg page when done.

図 43. Voltage Loop Compensation Designed With Measured Plant Information



C2000 Compensation Designer GUI: Vienna Rectifier Voltage Loop

Comp Number and Style 2 DCL_PI_C1/C2/L1

Save COMP

Plant Option SFRA Data

Browse SFRA Data

Pole Zero Format K_{DC} 0.4

f_{z0} 0.001

f_{p0} 0

Enter Zero Location in kHz

Control Freq 50 **kHz**

Comp Coeff

DCL_PI_C1/C2/L1 Coeff

Kp= 0.3999749
Ki= 0.0001257

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

Stable Loop **Bandwidth** 0.008 kHz **Gain Margin** NA dB **Phase Margin** 104.11 Degrees

10. This verifies the voltage compensator design.

11. To bring the system to a safe stop bring the input AC voltage down to zero, observe the guiVBus will

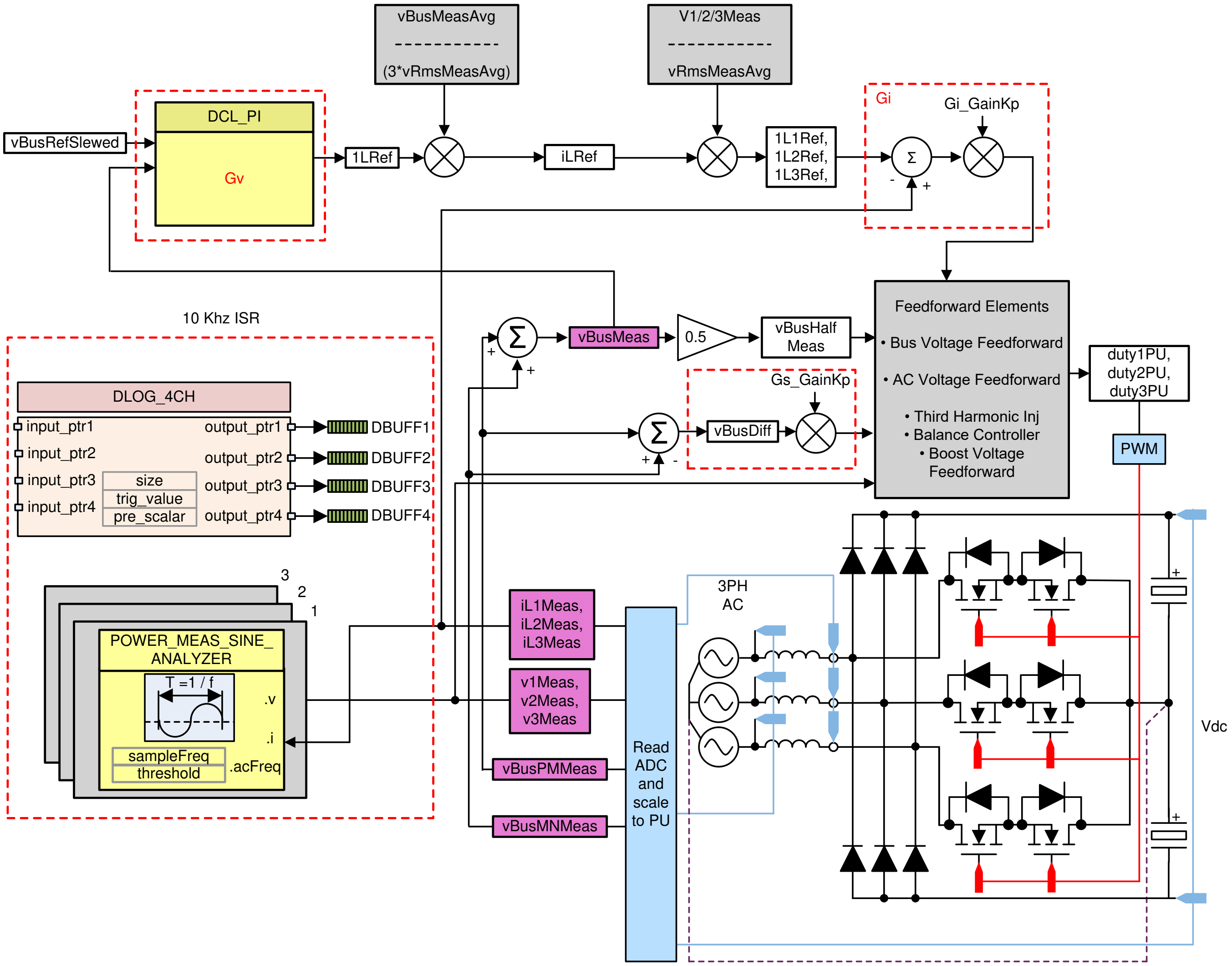
come down to zero as well.

12. Fully halting the MCU when in real-time mode is a two-step process. First halt the processor by using the **Halt** button on the toolbar (⏸) or by using **Target** > **Halt**. Then take the MCU out of real-time mode by clicking on . Finally, reset the MCU ( ).
13. Close CCS debug session by clicking on **Terminate Debug Session** (**Target** > **Terminate all**).

6.4.4 INCR_BUILD 4: Closed Balance, Voltage, and Current Loop

In this build the board is operated as a three-wire system, that is, the neutral of the power supply is not connected to the DC midpoint of the output. To maintain the DC bus balance, a balance loop with a simple proportional gain is added in the control structure as shown in 図 44. In this build a third harmonic injection is also carried out, which helps in stabilizing the DC bus balance point.

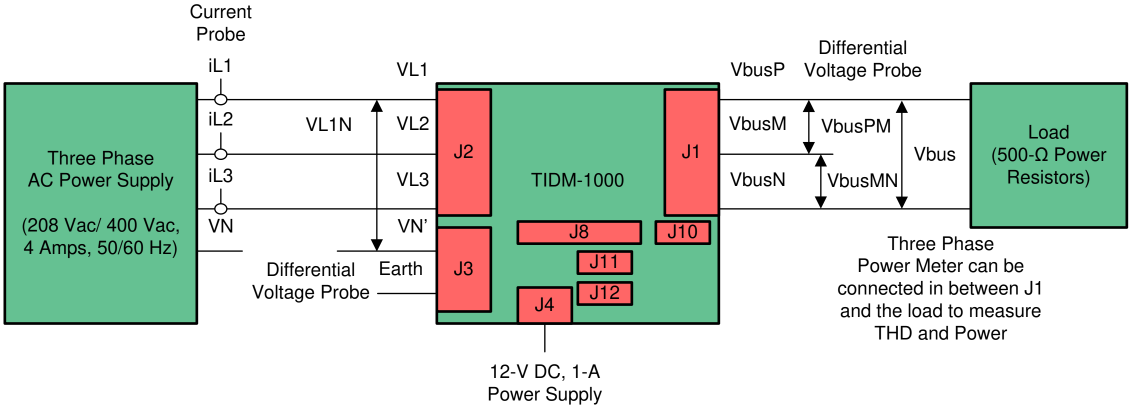
図 44. Build Level 4 Control Diagram: Output Voltage, Inductor Current, and Bus Cap Balance Loop



6.4.4.1 Setting Software Options for BUILD 4

1. Make sure the hardware is setup, as shown in 図 45. One major difference from the previous builds is the Neutral is no longer connected to the midpoint of the DC bus capacitor.

図 45. Build Level 4 Hardware Setup Diagram

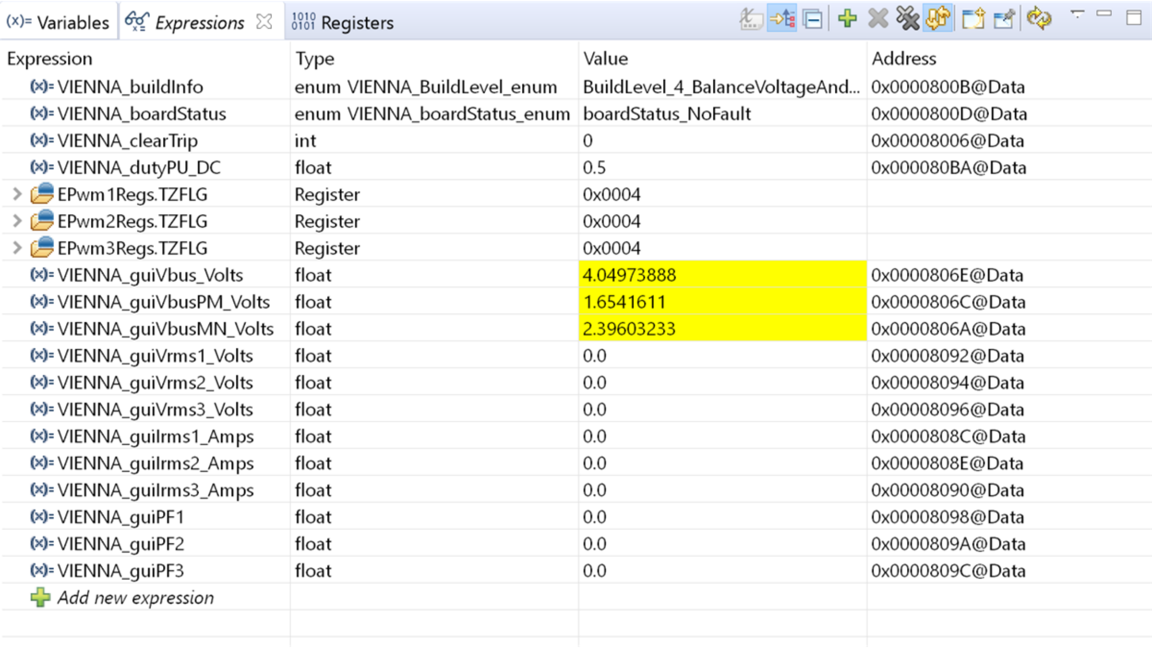

Current Probe
iL1
iL2
iL3
VN
Three Phase AC Power Supply
(208 Vac/ 400 Vac, 4 Amps, 50/60 Hz)
VL1
VL1N
VL2
VL3
VN'
Differential Voltage Probe
Earth
J2
J3
TIDM-1000
J1
J8
J10
J11
J12
J4
12-V DC, 1-A Power Supply
VbusP
VbusM
VbusN
Differential Voltage Probe
VbusPM
VbusMN
Vbus
Load (500-Ω Power Resistors)
Three Phase Power Meter can be connected in between J1 and the load to measure THD and Power

2. On the powerSUITE page below *Project Options* select *Closed Voltage, Current and Bus Cap Balance Loop*.
3. Save the page.

6.4.4.2 Building and Loading the Project and Setting up Debug

1. Now right click the project name and click **Rebuild Project**. The project will build successfully. Click *Run* > **Debug**; this will launch a debugging session. In the case of dual CPU devices, a window may appear to select the CPU the debug needs to be performed. In this case, select CPU1. The project will then load on the device and CCS debug view will become active. The code will halt at the start of the main routine.
2. To add the variables in the watch or expressions window click **View** > **Scripting Console** to open the scripting console dialog box. On the upper-right corner of this console, click open to browse to the **setupdebugenv_build4.js** script file located inside the project folder. This file will populate the watch window with appropriate variables needed to debug the system. Click the **Continuous Refresh** button () on the watch window to enable continuous update of values from the controller. The watch window will appear as in 図 46.

図 46. Build Level 4: Expressions View



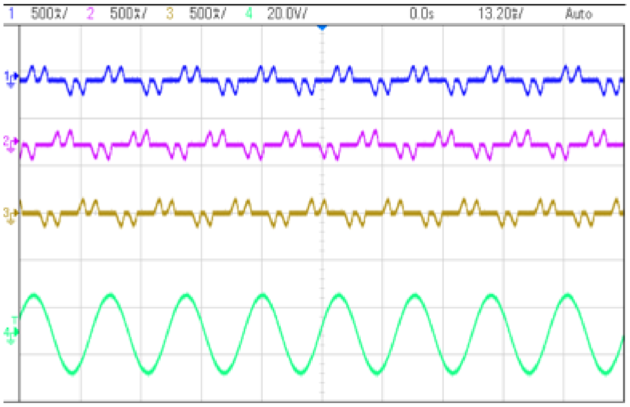
Expression	Type	Value	Address
VIENNA_buildInfo	enum VIENNA_BuildLevel_enum	BuildLevel_4_BalanceVoltageAnd...	0x0000800B@Data
VIENNA_boardStatus	enum VIENNA_boardStatus_enum	boardStatus_NoFault	0x0000800D@Data
VIENNA_clearTrip	int	0	0x00008006@Data
VIENNA_dutyPU_DC	float	0.5	0x000080BA@Data
EPwm1Regs.TZFLG	Register	0x0004	
EPwm2Regs.TZFLG	Register	0x0004	
EPwm3Regs.TZFLG	Register	0x0004	
VIENNA_guiVbus_Volts	float	4.04973888	0x0000806E@Data
VIENNA_guiVbusPM_Volts	float	1.6541611	0x0000806C@Data
VIENNA_guiVbusMN_Volts	float	2.39603233	0x0000806A@Data
VIENNA_guiVrms1_Volts	float	0.0	0x00008092@Data
VIENNA_guiVrms2_Volts	float	0.0	0x00008094@Data
VIENNA_guiVrms3_Volts	float	0.0	0x00008096@Data
VIENNA_guiIrms1_Amps	float	0.0	0x0000808C@Data
VIENNA_guiIrms2_Amps	float	0.0	0x0000808E@Data
VIENNA_guiIrms3_Amps	float	0.0	0x00008090@Data
VIENNA_guiPF1	float	0.0	0x00008098@Data
VIENNA_guiPF2	float	0.0	0x0000809A@Data
VIENNA_guiPF3	float	0.0	0x0000809C@Data
Add new expression			

3. Enable real-time mode by hovering the mouse on the buttons on the horizontal toolbar and clicking the button.

6.4.4.3 Running Code (Build 4)

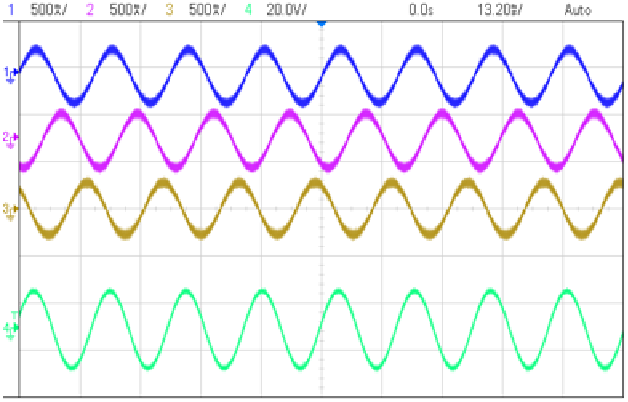
1. Run the project by clicking.
2. Raise the AC input to 120-Vrms VL-L and 208-Vrms VL-L, 60 Hz. A rectified current is going to be drawn from the input with PF close to 0.7 as shown in 図 47.

図 47. Build Level 4: Scope Capture IL1, IL2, IL3 and V1 (120Vrms L-N) with PWM Tripped



3. Bus voltage is set by the variable vBusRef and is set at 1.32 which corresponds to 600 V for this design.
4. Start the PFC action by writing a **1** to clearTrip variable
5. The board will now draw sinusoidal current and the PF will be close to 0.99 and THD will be around 2.5%. The scope capture will look as shown in 図 48.

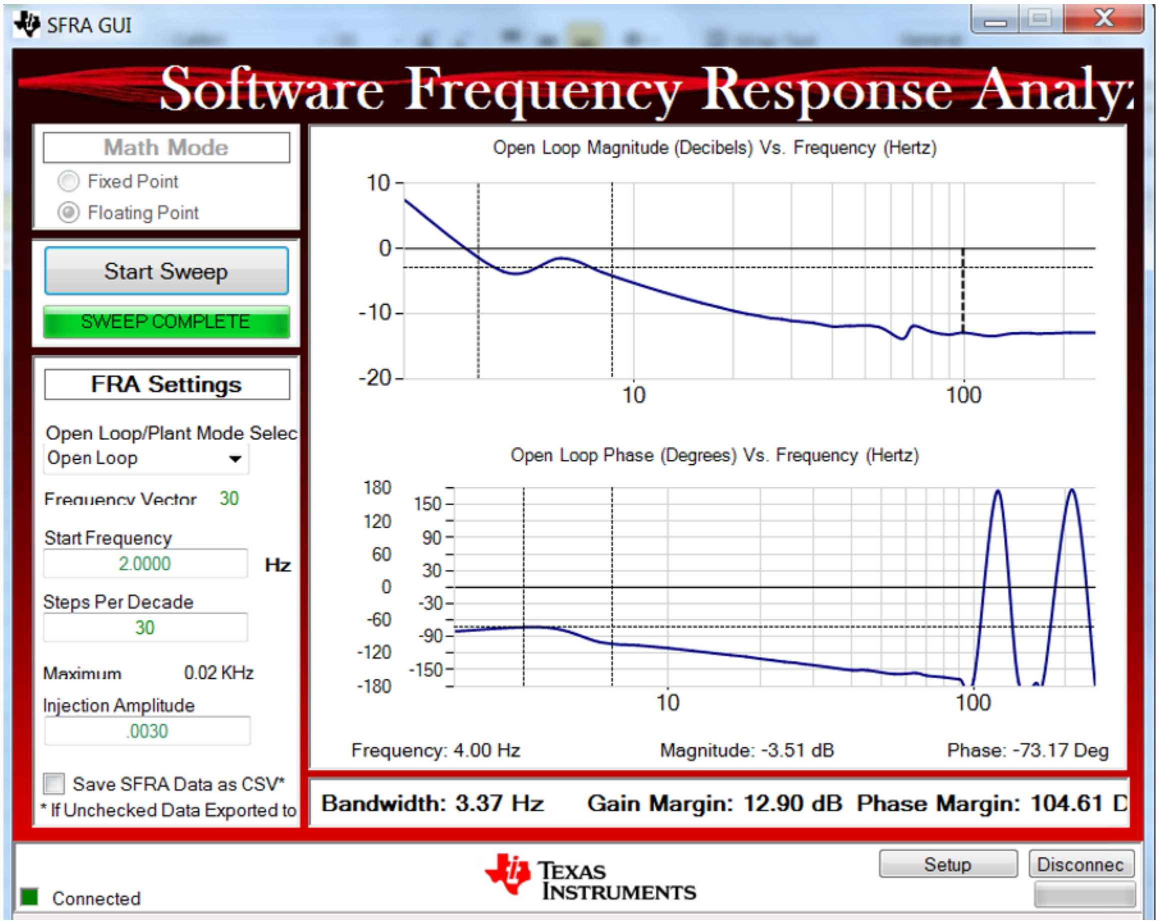
図 48. Build Level 4: Scope Capture IL1, IL2, IL3 and V1 (120-Vrms L-N) With Full PFC Build



6. The DC bus voltages will also be balanced, that is, guiVbusPM and guiVbusMN will be almost equal, which shows that the closed loop balance controller is working.
7. SFRA is integrated in the software of this build 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 syscfg page, click on the **SFRA** icon. SFRA GUI will pop-up.
8. Select the options for the device on the SFRA GUI. For example for F28379D, select floating point. Click setup connection, and on the pop-up window ,uncheck the boot on connect option and select an appropriate COM port and Click **OK**. Return to the SFRA GUI and Click **Connect**.

9. The SFRA GUI will connect to the device. A SFRA sweep can now be started by clicking **Start Sweep**. The complete SFRA sweep will take a few minutes to finish. Activity can be monitored by seeing the progress bar on the SFRA GUI and also by checking the flashing of blue LED on the back on the control card that indicate UART activity. When complete a graph with the open loop plot will appear, as shown in 図 49. This graph verifies that the designed compensator is indeed stable.

図 49. SFRA Run on Balance Voltage Loop



SFRA GUI

Software Frequency Response Analy:

Math Mode

○ Fixed Point

● Floating Point

Start Sweep

SWEEP COMPLETE

FRA Settings

Open Loop/Plant Mode Selec: Open Loop

Frequency Vector: 30

Start Frequency: 2.0000 Hz

Steps Per Decade: 30

Maximum: 0.02 KHz

Injection Amplitude: .0030

□ Save SFRA Data as CSV*

* If Unchecked Data Exported to

Frequency: 4.00 Hz Magnitude: -3.51 dB Phase: -73.17 Deg

Bandwidth: 3.37 Hz Gain Margin: 12.90 dB Phase Margin: 104.61 D

Setup Disconnec

🟩 Connected

Open Loop Magnitude (Decibels) Vs. Frequency (Hertz)
10
0
-10
-20
10
100
Open Loop Phase (Degrees) Vs. Frequency (Hertz)
180
120
60
0
-60
-120
-180
150
90
30
-30
-90
-150
10
100
Texas Instruments

10. The balance loop open loop gain is controlled by the variable Gs_GainKp and can be adjusted in case the BW is not enough. Though, for the balance loop, the bandwidth needs to be lower than the outer voltage loop and only 1 to 2 Hz of bandwidth is sufficient.

11. Fully halting the MCU when in real-time mode is a two-step process. First, halt the processor by using the **Halt** button on the toolbar ( ) or by using **Target** > **Halt**. Then take the MCU out of real-time mode by clicking on . Finally reset the MCU .

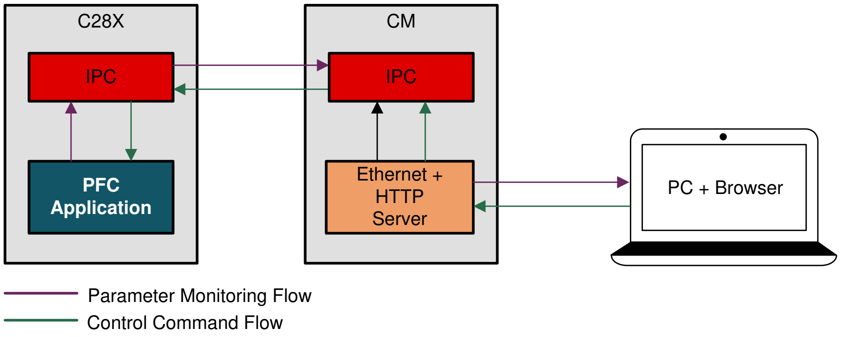
12. Close CCS debug session by clicking on *Terminate Debug Session* (**Target** > **Terminate all**). .

6.4.5 Monitoring and Control of Vienna Rectifier Based on the HTTP GUI Page and Ethernet Support (F2838x only)

In this section, the ethernet support is demonstrated together with Vienna rectifier. The ethernet based monitoring can be applied to all four Incremental builds and the ethernet based control can be applied to INCR_BUILD 3 and INCR_BUILD 4. Test in this section is based on INCR_BUILD 4.

6.4.5.1 System configuration

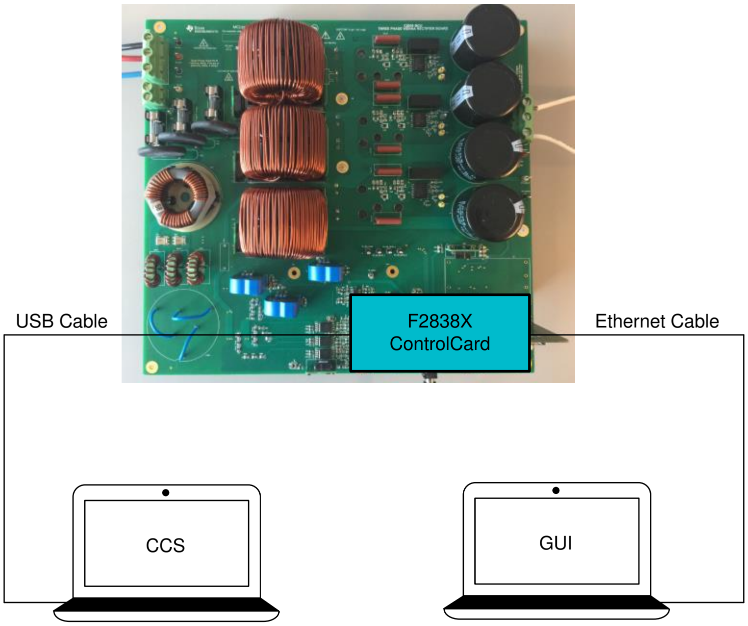
図 50. Vienna PFC and Ethernet Configuration



There are two data flows in this test as shown in 図 50. In the parameter monitoring flow, the three phase current, voltage, and power factors are monitored and shown in HTTP GUI page. In the control command flow, the output voltage reference and command signal (Start PFC --cleartrip) can be set in the GUI page and control the Vienna rectifier.

6.4.5.2 Test setup

図 51. Vienna PFC and Ethernet Test Setup



According to 図 51

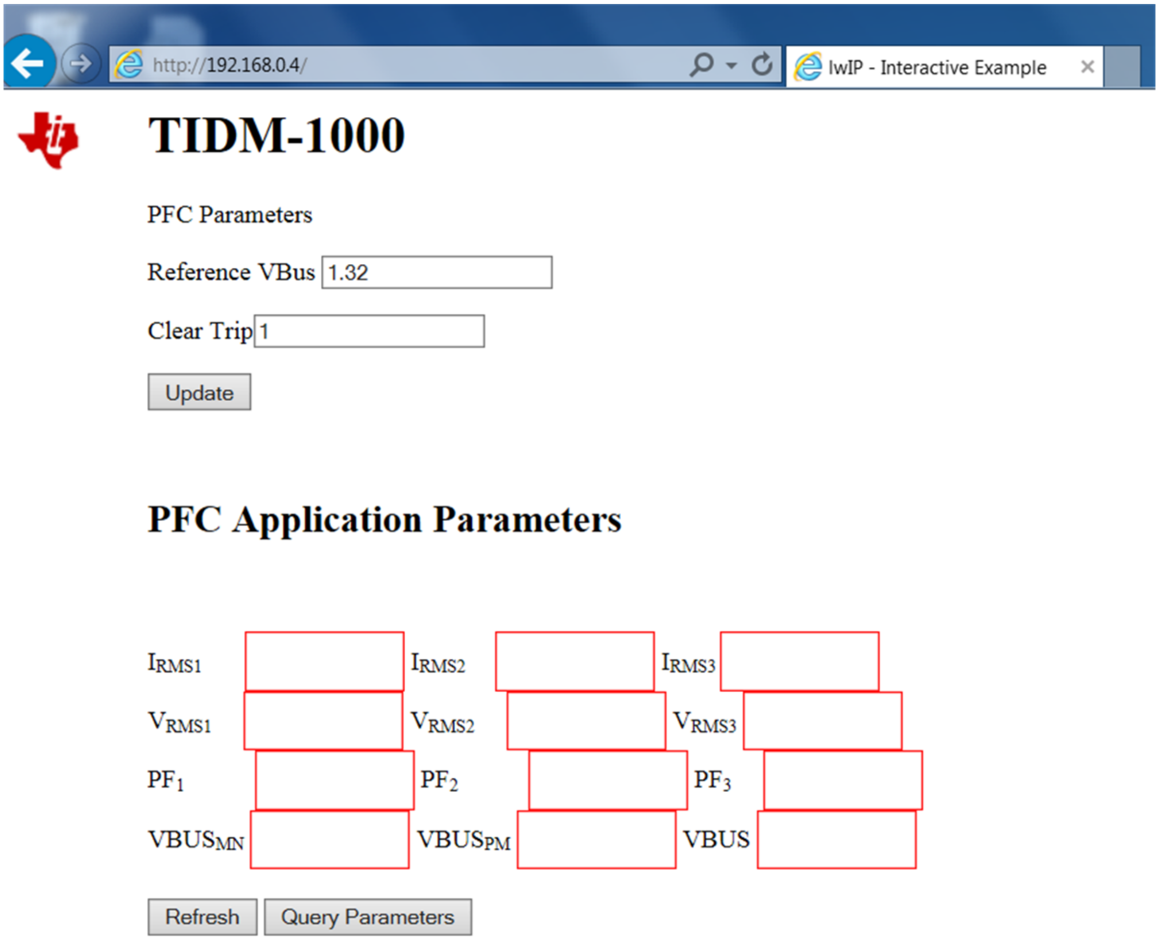
1. Prepare two laptop computers: CCS is running on PC1. PC2 is used to control and monitor the PFC through a GUI HTTP page.
2. Make sure the hardware setup is the same with INCR_BUILD 4. Connect PC2 to Port J4 (Ethernet connector – RJ45 connector) of TMDSCNCD28388D control card using an ethernet cable.
3. Change the network setting of PC2 to enable static IP, configure it to 192.168.0.7 (anything other than 192.168.0.4 which is the IP Address of F2838x)

6.4.5.3 Test procedure

1. Follow the step 1~3 in section 6.4.4.1

2. Open the **vienna_user_settings.h**, Change the setting from **#define ETHERNET_DEMO ETHERNET_DISABLED** to **#define ETHERNET_DEMO ETHERNET_ENABLED**. Click **Save**. This enables to send the command signal by GUI page through ethernet. Notes: Do not forget to change this settings back if ethernet is not used and normal INCR_BUILD 4 is selected)
3. Follow the step 1~2 in section 6.4.4.2
4. Connect CM core and load enet_lwip_cm.out to the CM core. enet_lwip_cm.out is in the folder c2000ware-digital-power-sdk\solutions\tidm_1000\f2838x\enet_lwlp_cm\examples\enet_lwip\cm\ccs\Flash. Notes: It is not suggested to change enet_lwip_cm.out file. However, if customers would like to change it, the project (c2000ware-digital-power-sdk\solutions\tidm_1000\f2838x\enet_lwlp_cm\examples\enet_lwip) needs to be open independently in the CCS and the driverlib (c2000ware-digital-power-sdk\c2000ware\driverlib\f2838x\driverlib_cm) needs to be added to the project to recompile.
5. Click **Texas Instruments XDS100v2 USB Debug Probe_0/C28xx_CPU1** and click **Run**
6. Click **Texas Instruments XDS100v2 USB Debug Probe_0/C28xx_CPU1** and click **Halt**
7. Enable real-time mode by hovering the mouse on the buttons on the horizontal toolbar and clicking the button.
8. Click **Texas Instruments XDS100v2 USB Debug Probe_0/C28xx_CPU1** and click **Run**
9. Click **Texas Instruments XDS100v2 USB Debug Probe_0/Cortex_M4_0** and click **Run**
10. Open a Web browser (Chrome/IE) on the PC2 to which TMDSCNCD28388D control card is connected and type 192.168.0.4. You should see a GUI page similar to 図 52

図 52. Ethernet GUI Page of TIDM-1000



TIDM-1000

PFC Parameters

Reference VBus 1.32

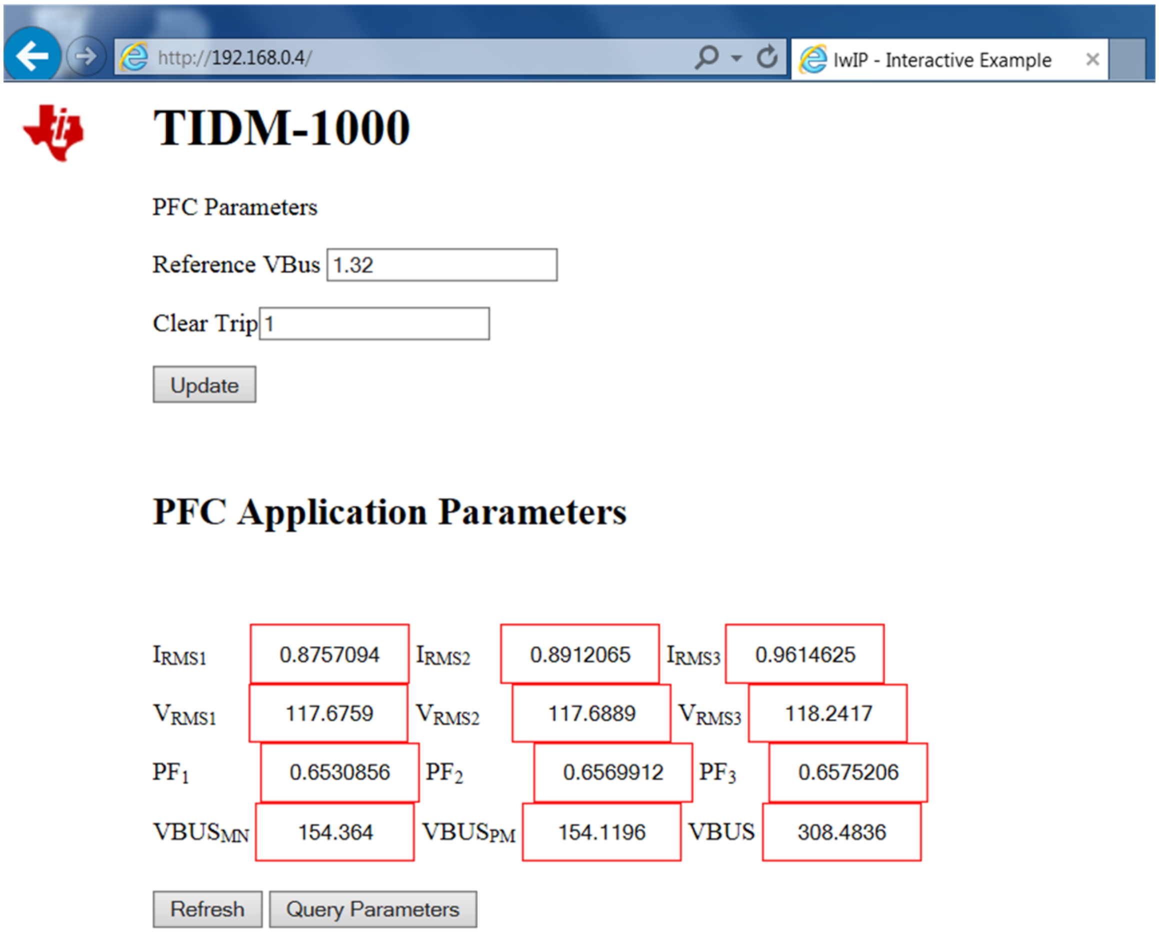
Clear Trip 1

PFC Application Parameters

IRMS1 IRMS2 IRMS3
VRMS1 VRMS2 VRMS3
PF1 PF2 PF3
VBUSMN VBUSPM VBUS

11. Gradually increase the input voltage to 120V. Check all the parameters in the expression window. Click the “Query parameters” and then click Refresh to display the parameters in the web page. All the parameters should appear in the GUI page 図 53. Compare those values with the values shown in the expression window. Make sure all the displayed parameters in GUI are correct. Do not go forward if this step is not fully verified.

図 53. Ethernet GUI Page Before Starting PFC



TIDM-1000

PFC Parameters

Reference VBus 1.32

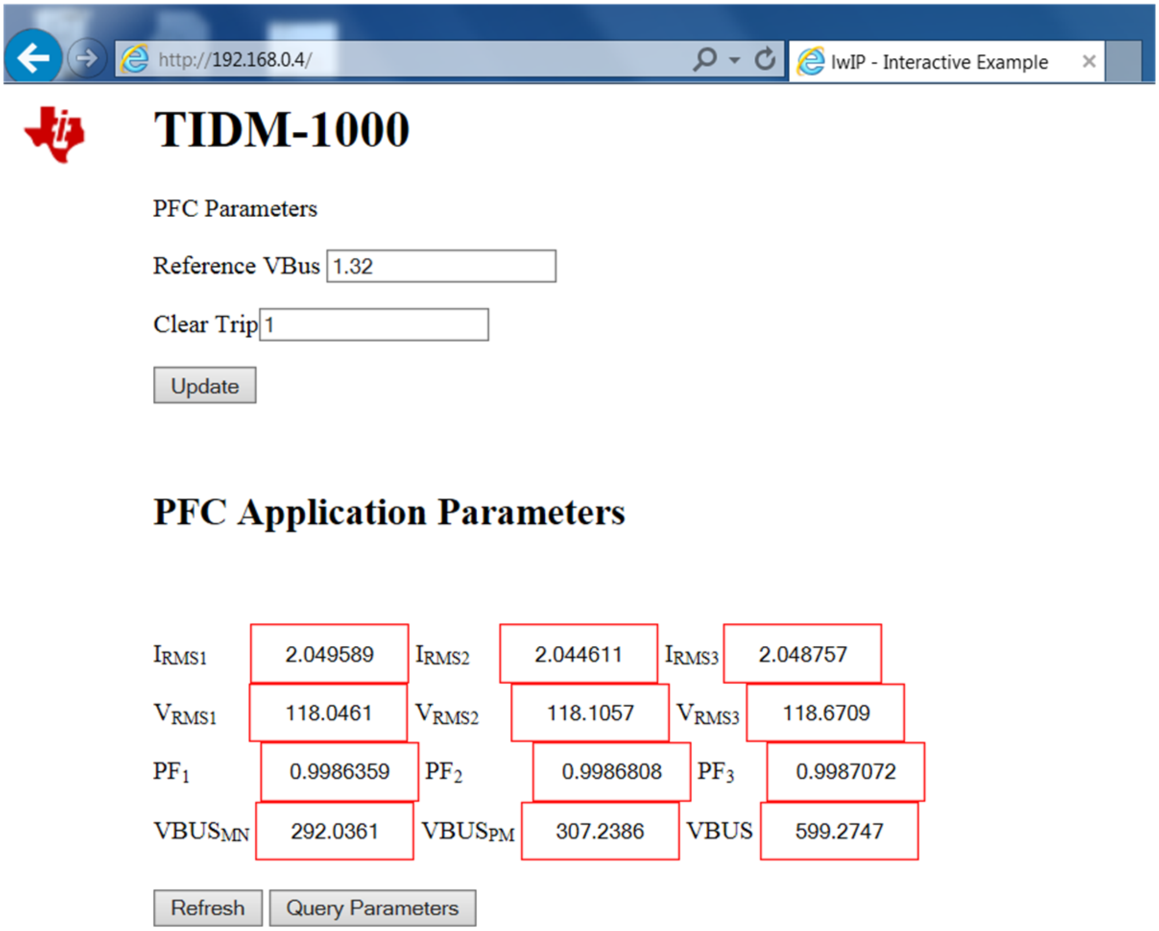
Clear Trip 1

PFC Application Parameters

I_{RMS1}	0.8757094	I_{RMS2}	0.8912065	I_{RMS3}	0.9614625
V_{RMS1}	117.6759	V_{RMS2}	117.6889	V_{RMS3}	118.2417
PF_1	0.6530856	PF_2	0.6569912	PF_3	0.6575206
$VBUS_{MN}$	154.364	$VBUS_{PM}$	154.1196	VBUS	308.4836

12. Now, make sure the **Reference Vbus** is 1.32 which is the per unit value and the **Clear Trip** is 1. Click the **Update** button and it should start PFC. Click the **Query parameters** and then click **Refresh** to display the parameters in the webpage. If everything is correct, all the parameters should be updated as 図 54.

図 54. Ethernet GUI Page After Starting PFC



TIDM-1000

PFC Parameters

Reference VBus 1.32

Clear Trip 1

PFC Application Parameters

I_{RMS1}	2.049589	I_{RMS2}	2.044611	I_{RMS3}	2.048757
V_{RMS1}	118.0461	V_{RMS2}	118.1057	V_{RMS3}	118.6709
PF_1	0.9986359	PF_2	0.9986808	PF_3	0.9987072
$VBUS_{MN}$	292.0361	$VBUS_{PM}$	307.2386	VBUS	599.2747

13. The final step is to change the "Reference Vbus" to 1 and click Update. The output voltage should decrease to around 450V. Other "Reference Vbus" less than 1.32 can be chosen.

6.4.6 Running on CLA

This solution is supported with an option to run the code on the CLA. This option is selected using a drop down box below project option on the powerSUITE main.syscfg page. Running on CLA can be selected for any build level option.

注: SFRA libray does not support CLA, hence the SFRA cannot be run when using CLA.

DLOG is also not used when using CLA, hence the data logging graphs will not work when using CLA

When the option is changed the SYSCFG file must be saved and the project must be rebuilt. When recompiled follow the steps as outlined in the specific incremental build level documentation.

Depending on the device, for example for F2837x CLA only supports non nestable CLA tasks, only one ISR can be offloaded to the CLA. By default if the selection from the powerSUITE page is made the faster ISR is moved to the CLA. Which is the 50kHz ISR in this TID. In case of device such as F28004x and F2838x, where CLA supports a background task from which a CLA task can be nested to. Both the 50kHz control ISR and 10kHz instrumentation ISR are offloaded to the CLA by default. If for some reason the user does not want to run the 10kHz ISR on the CLA, the option to do so is available below the "USER SECTION" in the solutions-settings.h file.

```
#if VIENNA_CONTROL_RUNNING_ON== CLA_CORE
#define VIENNA_INSTRUMENTATION_ISR_RUNNING_ON CLA_CORE
#else
#define VIENNA_INSTRUMENTATION_ISR_RUNNING_ON C28x_CORE
#endif
```

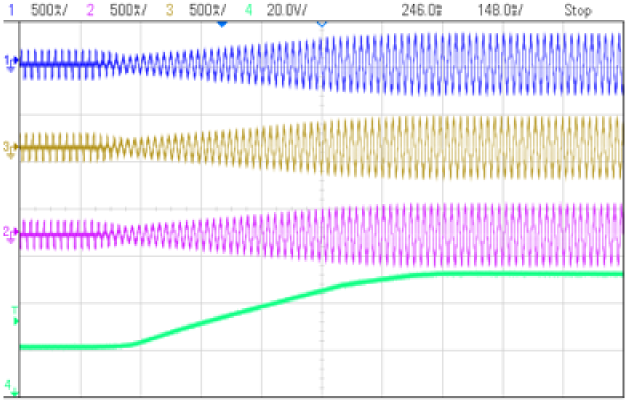
7 Testing and Results

7.1 Test Results at Input 208-Vac L-L, 60Hz, Output 600-V DC

7.1.1 Startup

The startup sequence of the power stage is shown in 図 55 with input three phase 208-Vrms VL-L and output bus regulated at 600 V and a 612 W load.

図 55. Startup of PFC Operation at 208-Vac IN, 600-V DC OUT and 612-WΩ Load



7.1.2 Steady State Condition

Steady state current waveform are shown in 図 56 and 図 57 for different load conditions 612 W and 1364 W, respectively.

図 56. Steady State 208-Vac IN, 600-V DC OUT 612W, iTHD 2.5%

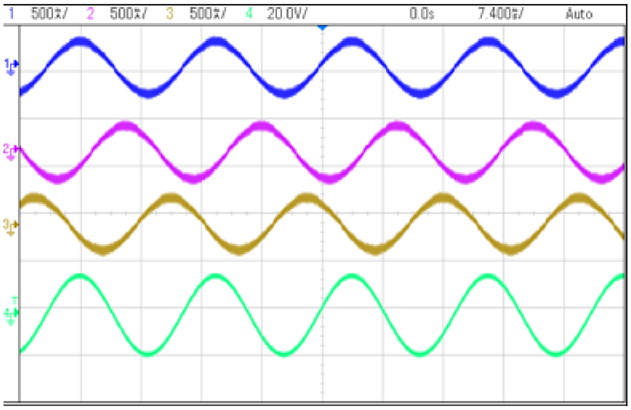


図 57. Steady State 208-Vac, IN 600-V DC OUT 1364W, iTHD 0.96%

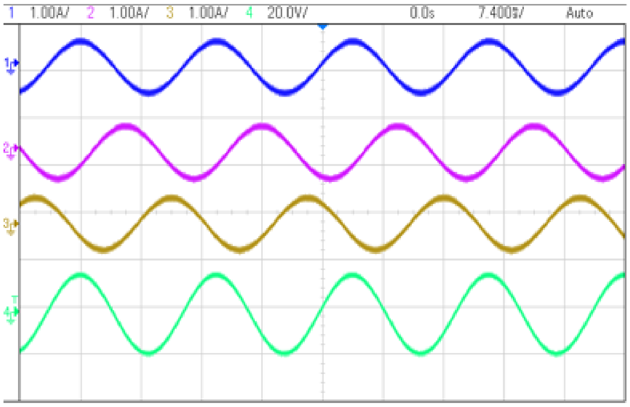


表 4 lists the detailed test results of this design below varying load conditions with 208-Vac input and 600-V DC output.

表 4. Detailed Test Results With 208Vac IN, 600V DC OUT, and Varying Power Levels

Vbus_PM	Vbus_MN	VoutTotal	PIN	Iout	Pout	Eff	THD%	PF
297.85	300.5	598.35	74	0.117	70.00695	0.946039865	23.20%	0.8263
298.58	300	598.58	124.4	0.2	119.716	0.962347267	17%	0.9206
299.12	299.8	598.92	245.1	0.4	239.568	0.977429621	8.09%	0.9777
299.78	299	598.78	481.3	0.788	471.83864	0.980342074	3%	0.994
299.52	299.6	599.12	623.7	1.022	612.30064	0.981723008	2.54%	0.9964
299.66	299.4	599.06	858.8	1.405	841.6793	0.980064392	1.71%	0.998
299.84	299.1	598.94	1001.6	1.637	980.46478	0.978898542	1.52%	0.9984
299.3	299.82	599.12	1150.6	1.878	1125.14736	0.977878811	1.22%	0.9988
300.13	298.8	598.93	1399.7	2.278	1364.36254	0.974753547	0.96%	0.999

図 58 shows the efficiency data plotted below these test conditions.

図 58. Efficiency at 208-Vac IN and 600-V DC OUT

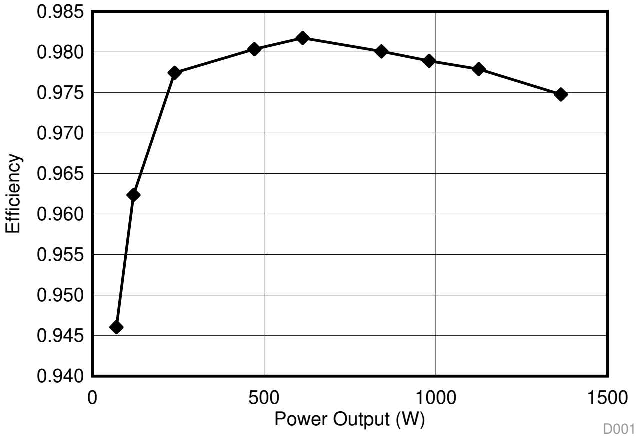


図 59 shows the THD data plotted below these test conditions.

図 59. THD at 208-Vac IN and 600-V DC OUT

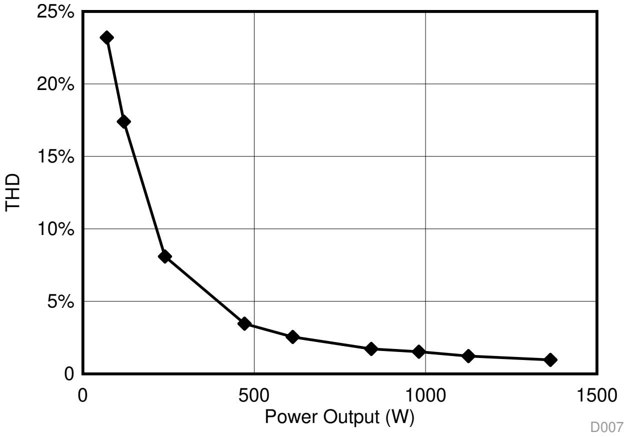


図 60 shows the PF data plotted below these test conditions.

図 60. PF at 208-Vac IN and 600-V DC OUT

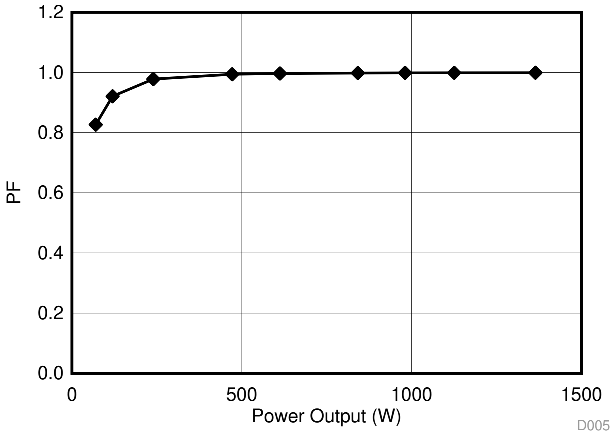
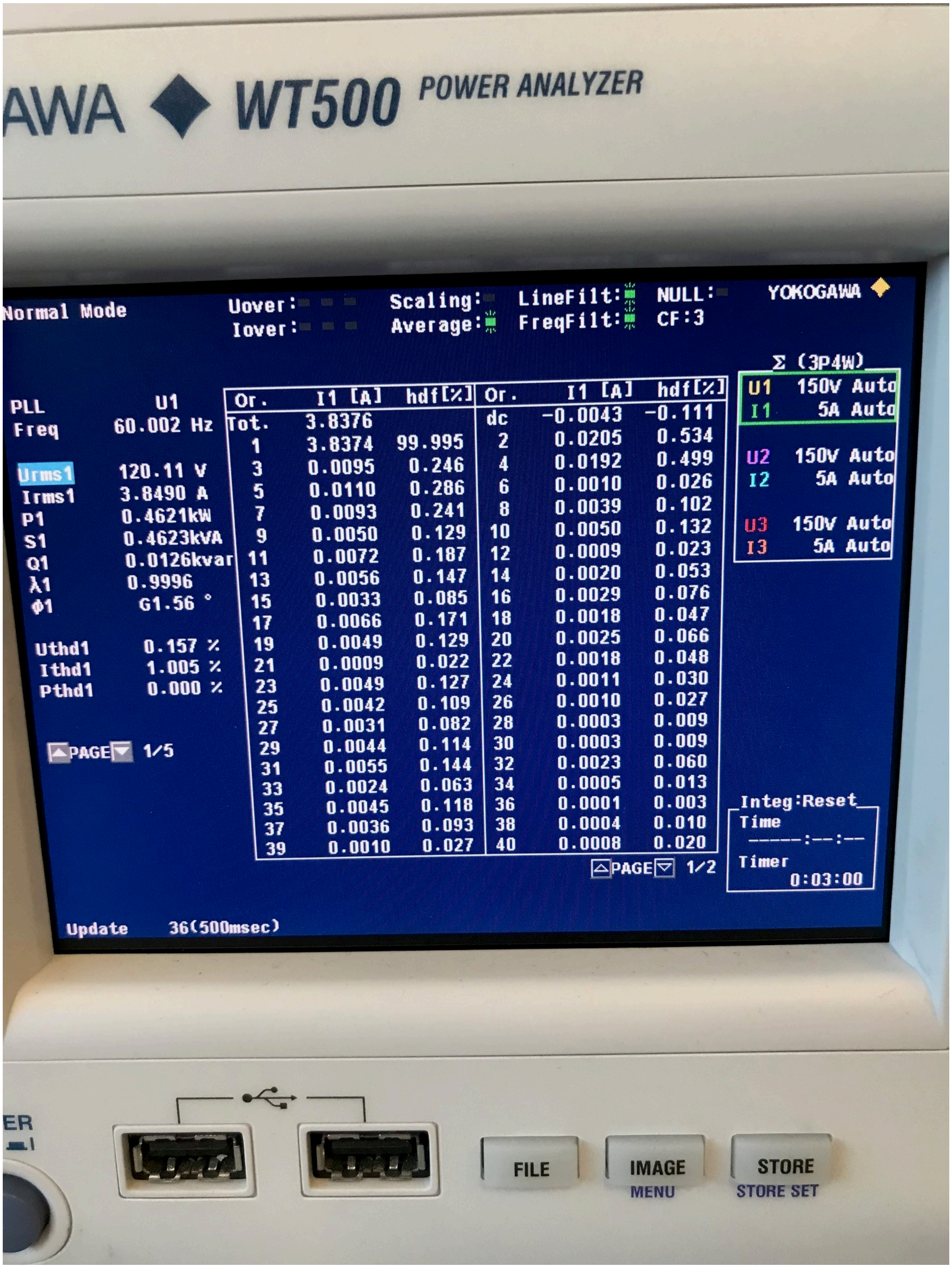


図 61 shows the power analyzer capture of the harmonics of the current when operating at 208Vrms input, 600V DC output and 1.2kW load.

図 61. Power Analyzer Capture of Harmonics at 208-V AC Input, 600-V DC Output, 1.2-kW Load



```
Normal Mode          Uover:      Scaling:    LineFilt:   NULL:    YOKOGAWA
                     Iover:      Average:    FreqFilt:   CF:3

PLL         U1
Freq    60.002 Hz

Urms1     120.11 V
Irms1     3.8490 A
P1        0.4621kW
S1        0.4623kVA
Q1        0.0126kvar
λ1        0.9996
Φ1         G1.56 °

Uthd1      0.157 %
Ithd1      1.005 %
Pthd1      0.000 %

PAGE 1/5
```

Or.	I1 [A]	hdf[%]	Or.	I1 [A]	hdf[%]
Tot.	3.8376		dc	-0.0043	-0.111
1	3.8374	99.995	2	0.0205	0.534
3	0.0095	0.246	4	0.0192	0.499
5	0.0110	0.286	6	0.0010	0.026
7	0.0093	0.241	8	0.0039	0.102
9	0.0050	0.129	10	0.0050	0.132
11	0.0072	0.187	12	0.0009	0.023
13	0.0056	0.147	14	0.0020	0.053
15	0.0033	0.085	16	0.0029	0.076
17	0.0066	0.171	18	0.0018	0.047
19	0.0049	0.129	20	0.0025	0.066
21	0.0009	0.022	22	0.0018	0.048
23	0.0049	0.127	24	0.0011	0.030
25	0.0042	0.109	26	0.0010	0.027
27	0.0031	0.082	28	0.0003	0.009
29	0.0044	0.114	30	0.0003	0.009
31	0.0055	0.144	32	0.0023	0.060
33	0.0024	0.063	34	0.0005	0.013
35	0.0045	0.118	36	0.0001	0.003
37	0.0036	0.093	38	0.0004	0.010
39	0.0010	0.027	40	0.0008	0.020

```
PAGE 1/2

Σ (3P4W)
U1   150V Auto
I1     5A Auto

U2   150V Auto
I2     5A Auto

U3   150V Auto
I3     5A Auto

Integ:Reset
Time
  ------:--:--
Timer
       0:03:00

Update    36(500msec)
```

7.1.3 Transient Test With Step-Load Change

The PFC stage control is composed of an inner current loop, which tries to follow the input voltage and an outer voltage loop that tries to maintain a constant DC bus voltage at the output. The voltage loop is thus in conflict with the current loop and hence must be designed to be very low bandwidth (approximately 10 Hz) to achieve good power factor. The slow voltage loop results in significant overshoot and undershoot below transients, see 図 62 and 図 63.

図 62. Voltage Overshoot With 1-kW Transient Without Non-Linear Voltage Loop

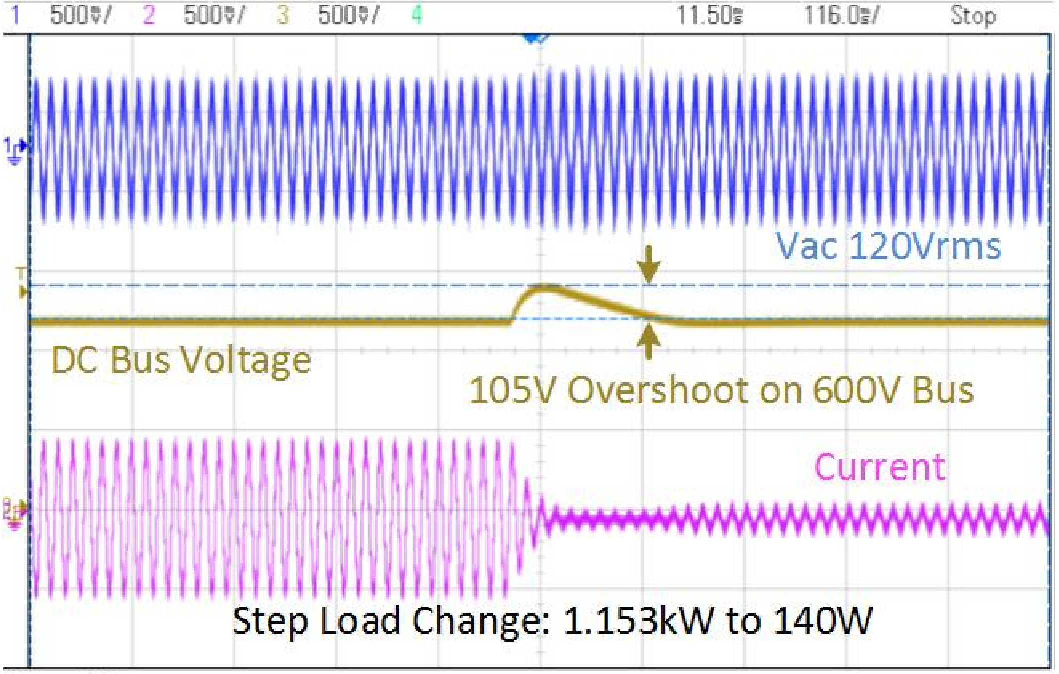
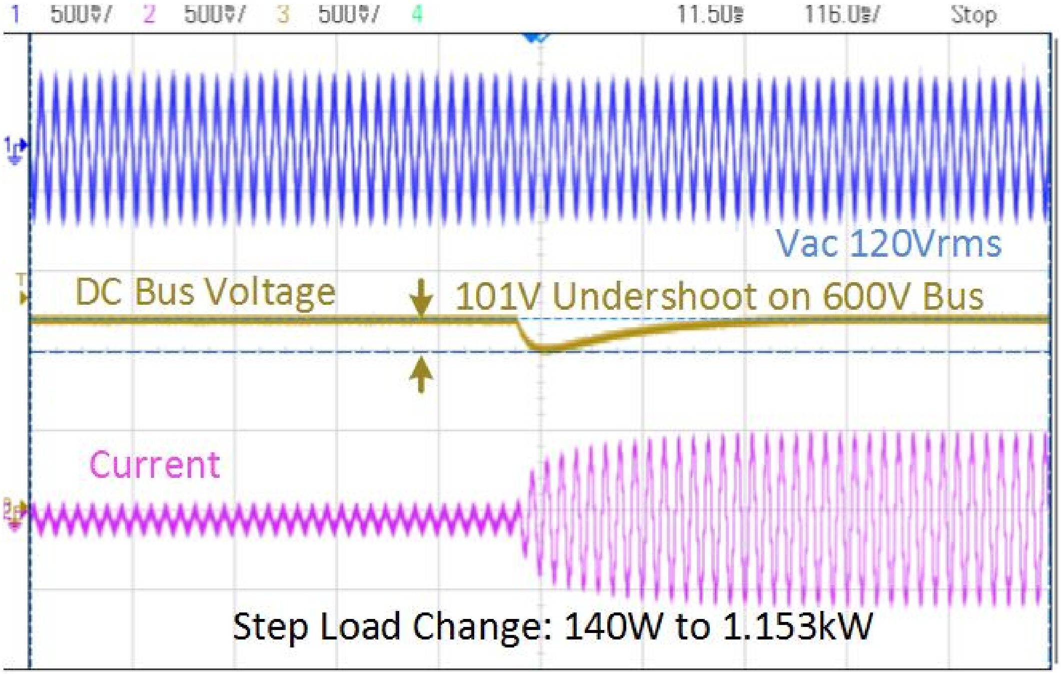
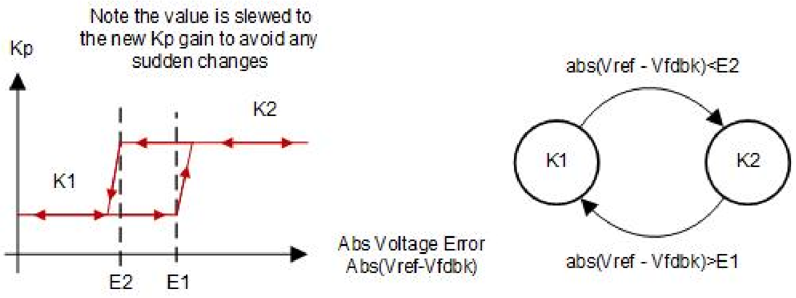


図 63. Voltage Undershoot With 1-kW Transient Without Non-Linear Voltage Loop



To improve voltage overshoot and undershoot, while maintaining good power factor a non-linear voltage control loop is implemented as shown in 図 64.

図 64. Non-Linear Voltage Loop With Hysteresis



A hysteresis band is added in the non-linear voltage loop to avoid oscillation between high-gain and low-gain mode. Furthermore the gain change is slewed to avoid any sudden changes. 図 65 and 図 66 show the results with a non-linear voltage loop, which shows significant reduction in the overshoot and undershoot compared to the results without the non-linear voltage loop.

図 65. Voltage Overshoot With 1-kW Transient, With Non-Linear Voltage Loop

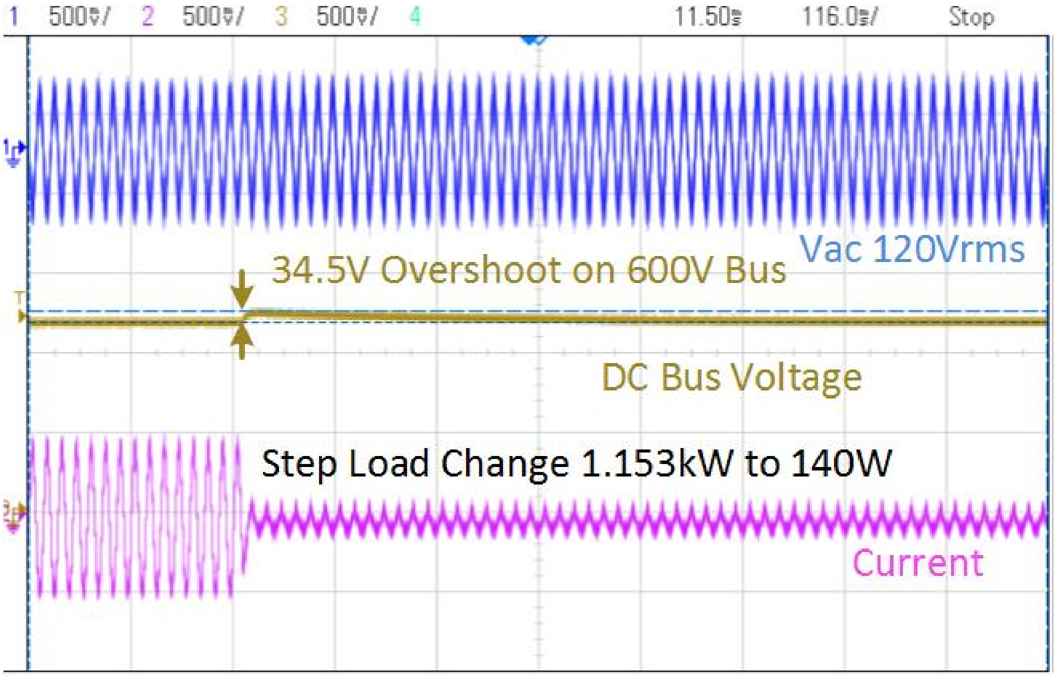
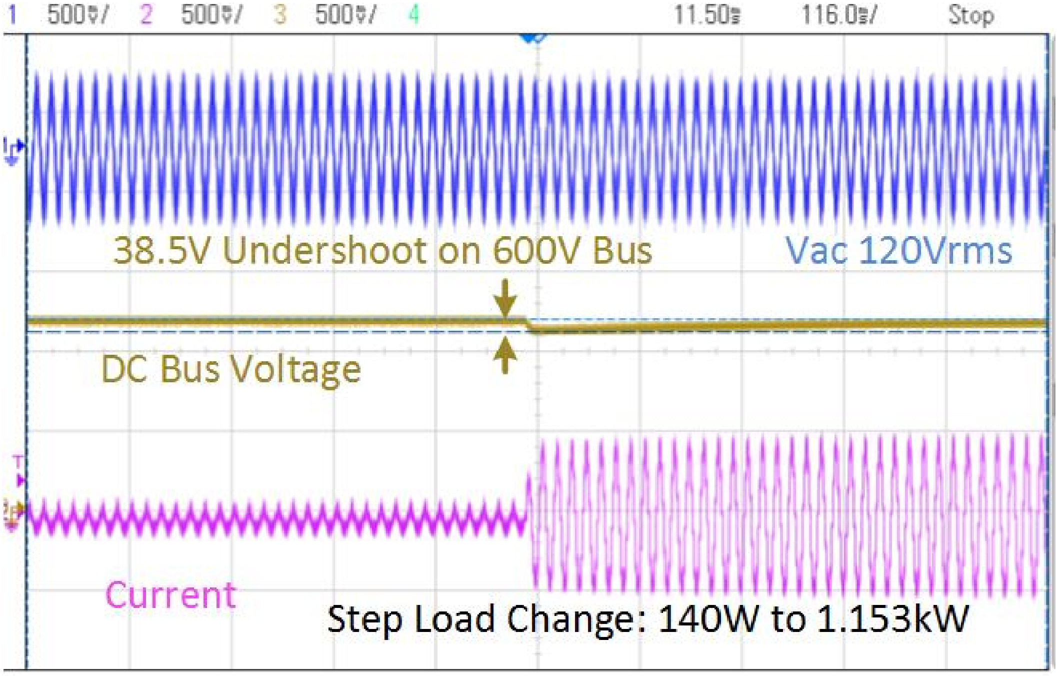


図 66. Voltage Undershoot With 1-kW Transient, With Non-Linear Voltage Loop



7.2 Test Results at Input 400-Vac 50-Hz, Output 700-V DC

7.2.1 Steady State Condition

Steady state current waveform for input Vac 400-Vrms/50H and output 700-V DC are shown in 図 67 and 図 68 for different load conditions 960 W and 1865 W, respectively.

図 67. Steady State 400-Vac IN 50 Hz, 700-V DC OUT 960 W, iTHD 7%

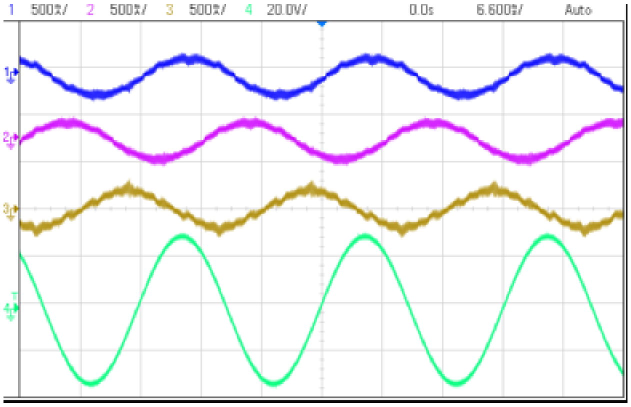


図 68. Steady State 400-Vac IN, 700-V DC OUT 1865 W, iTHD 3.5%

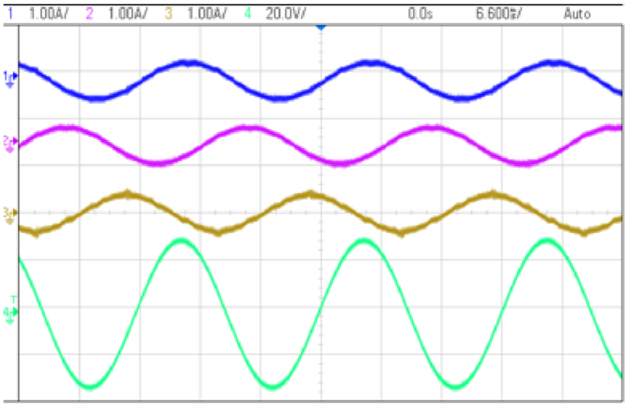


表 5 lists the detailed test results of this design under varying load conditions.

表 5. Detailed Test Results With 400-Vac IN, 700-V DC OUT, and Varying Power Levels

Vbus_PM	Vbus_MN	VoutTotal	PIN	Iout	Pout	Eff	THD%	PF
349.41	351.2	700.61	101.9	0.137	95.98357	0.941938862	29.10%	0.7526
349.41	351.9	701.31	265.3	0.37	259.4847	0.978080286	22%	0.9309
349.43	352	701.43	581.9	0.819	574.47117	0.987233494	11.30%	0.9823
349.49	351.8	701.29	900.7	1.27	890.6383	0.988829022	7%	0.9922
349.46	351.9	701.36	1094.6	1.543	1082.19848	0.988670272	6.20%	0.9982
349.61	351.7	701.31	1354.1	1.913	1341.60603	0.990773229	5.20%	0.9964
349.64	351.4	701.04	1553.4	2.191	1535.97864	0.988785014	4.70%	0.9972
349.75	351.2	700.95	1884.4	2.66	1864.527	0.989453938	3.50%	0.998

図 69 shows the efficiency data plotted under these test conditions.

図 69. Efficiency at 400-Vac IN and 700-V DC OUT

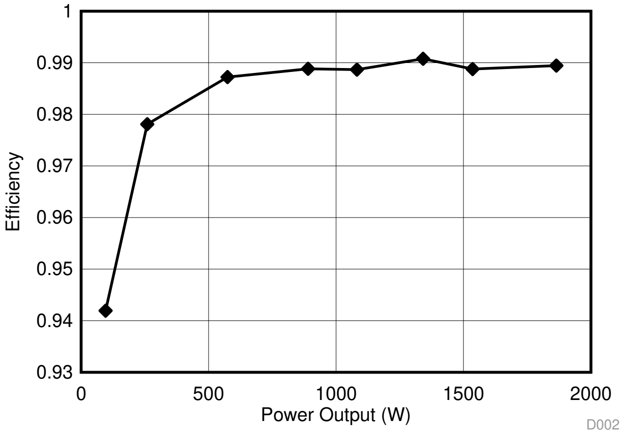


図 70 shows the THD data plotted under these test conditions.

図 70. THD at 400-Vac IN and 700-V DC OUT

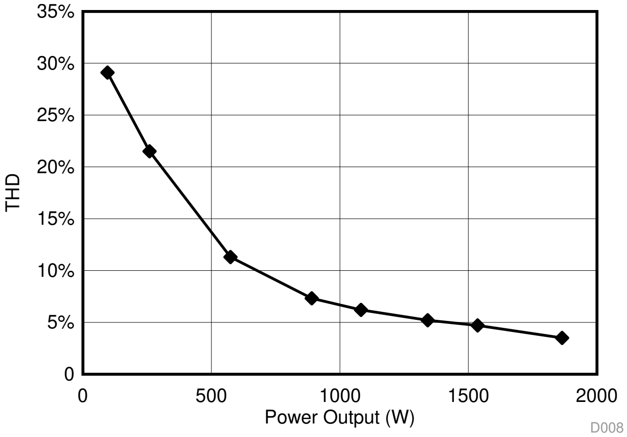


図 71 shows the PF data plotted under these test conditions.

図 71. PF at 400-Vac IN and 700-V DC OUT

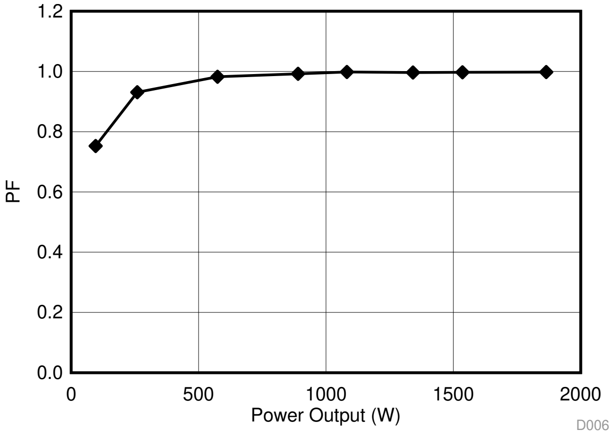
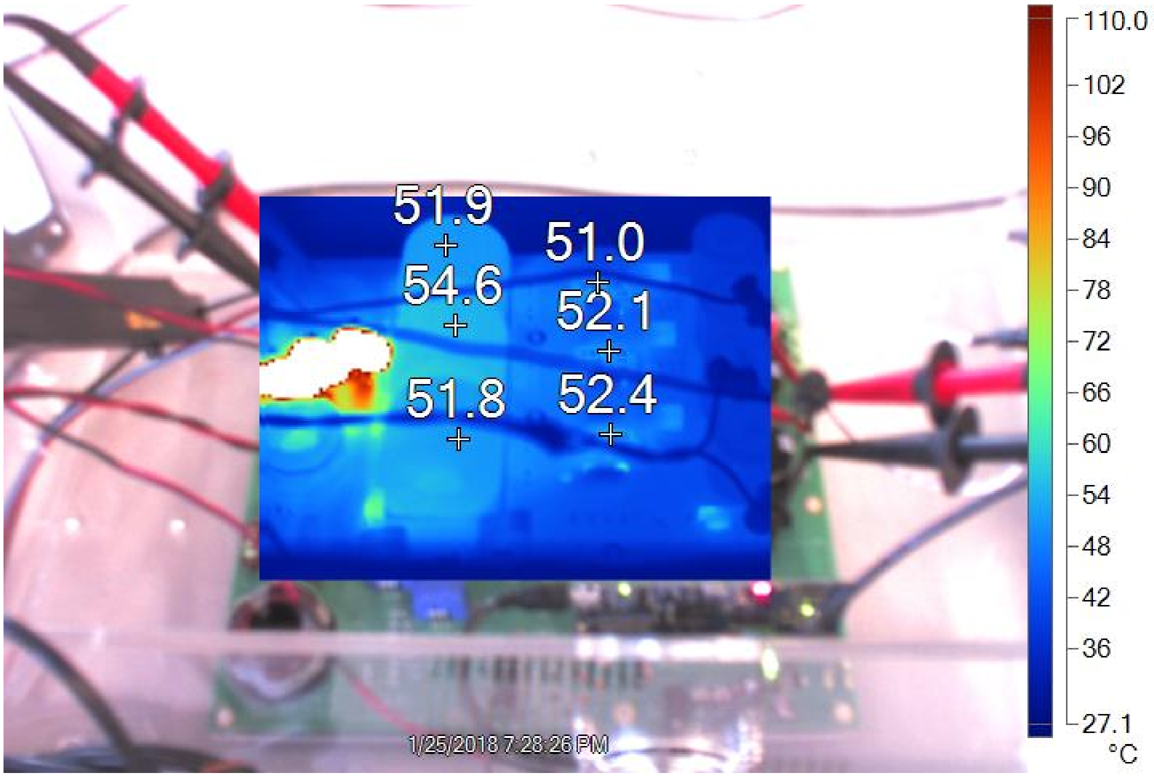


図 72 shows the thermal camera image when operating the board is at 2.4 kW. Except for the inrush-current control thermistors, the board runs below 60º Celsius. 図 72 shows thermal image of the vienna rectifier board when operating at 2.4 kW with high-line 230-Vrms L-N input voltage and 700-V DC bus output voltage.

図 72. Thermal Camera Image of Vienna Rectifier Operating at 2.4-kW, 230-Vrms VL-N Input, 700V DC Output



8 Design Files

See the design files at TIDM-1000 or under the DigitalPower SDK package at *C2000Ware_DigitalPower_SDK/solutions/tidm_1000/hardware* .

8.1 Schematics

To download the schematics, see the design files at TIDM-1000.

8.2 Bill of Material (BOM)

To download the BOM, see the design files at TIDM-1000.

8.3 PCB Layout Recommendations

8.3.1 Layout Prints

To download the layer plots, see the design files at TIDM-1000.

8.4 Altium Project

To download the Altium project files, see the design files at TIDM-1000.

8.5 Gerber Files

To download the Gerber files, see the design files at TIDM-1000.

8.6 Assembly Drawings

To download the assembly drawings, see the design files at TIDM-1000.

9 Software Files

To download the software files, see the design files at C2000Ware DigitalPower SDK. When installed, this reference design can be found at:

- C2000Ware_DigitalPower_SDK\solutions\tidm_1000\
  - \docs > Documentation
  - \hardware > PCB Altium Project, Gerbers, BOM, sense_calculation.xlsx
  - \<device>
    - <pfc3phvienna> > CCS Project

10 Related Documentation

1. Hartmann, M., S.d. Round, H. Ertl, and J.w. Kolar. "Digital Current Controller for a 1 MHz, 10 KW Three-Phase VIENNA Rectifier." *IEEE Transactions on Power Electronics* 24, no. 11 (2009): 2496-508. doi:10.1109/tpel.2009.2031437.
2. Texas Instruments, *C2000™ Software Frequency Response Analyzer (SFRA) Library and Compensation Designer User's Guide*
3. Texas Instruments, *TMS320F2837xD Dual-Core Delfino™ Microcontrollers Data Sheet*
4. Texas Instruments, *TMS320F28004x Piccolo™ Microcontrollers Data Sheet*

10.1 商標

C2000, E2E, Delfino are trademarks of Texas Instruments.
すべての商標および登録商標はそれぞれの所有者に帰属します。

11 About the Author

MANISH BHARDWAJ is a Systems Application Engineer with C2000 Microcontrollers System Solutions Group at Texas Instruments, where he is responsible for developing reference design solutions for digital power, motor control, and solar power applications. Before joining TI in 2009, Manish received his Masters of Science in Electrical and Computer Engineering from Georgia Institute of Technology, Atlanta and his Bachelor of Engineering from Netaji Subhash Institute of Technology, University of Delhi, India.

CHEN JIANG is a Systems Application Engineer with C2000 Microcontrollers System Solutions Group at Texas Instruments, where he is responsible for developing reference design solutions for digital power applications. Before joining TI in 2017, Chen received his PhD in Electrical and Computer Engineering from Georgia Institute of Technology, Atlanta and his Bachelor of Engineering from Zhejiang University, China.

改訂履歴

資料番号末尾の英字は改訂を表しています。その改訂履歴は英語版に準じています。

Revision F (November 2019) から **Revision G** に変更	**Page**
• *main.cfg* to *main.syscfg* throughout document 変更	2
• Figure17: powerSUITE Page for Vienne Rectifier Solution 変更	22
• The changes will only happen after rebuilding the entire project 追加	23
• *Kit.xml* to *Kit.json* 変更	23
• *powerSUITE CFG* to *powerSUITE SYSCFG* 変更	25
• *CFG page* to *SYSCFG page* 変更	37
• *vienna_settings.h* to *vienna_user_settings.h* 変更	51
• *powerSUITE main.cfg* to *powerSUITE main.syscfg* 変更	54

Revision E (June 2019) から **Revision F** に変更	**Page**
• F2804x を F28004x に 変更	1

Revision D (May 2018) から **Revision E** に変更	**Page**
• 「また、HTTP GUI ページとイーサネットのサポートにより、Vienna 整流器を監視・制御することもできます (F2838X のみ)。」を 追加	1
• UCC21620DW 削除	1
• UCC21520DW 追加	1
• 2838x、F 追加	1
• 「HTTP GUI ページとイーサネットのサポートにより、Vienna 整流器を監視・制御 (F2838X のみ)」を 追加	1
• TMS320F280049M to TMS320F280049C 変更	17
• text and bulleted list items. 追加	20
• TMS320F280049M to TMS320F280049C 変更	20
• TMDSCNCD280049M to TMDSCNCD280049C 変更	20
• TMS320F280049M to TMS320F280049C 変更	20
• text To route them to the corrent control card pins for the SDFM 削除	20
• down to up 変更	20
• TMS320F280049M to TMS320F280049C 変更	21
• version 7.4 to 9.0.1 変更	21
• pfc3phvienna to vienna 変更	23
• /F2838xD 追加	25
• /F2838xD 追加	25
• note 追加	26
• (VIENNA_guiVbus_Volts in the expression window) 追加	29
• guiVbus will be close to 320 V, and the guiVbusPM/MN will be close to 160 V each. The code runs a sine analyzer module, which computes the RMS value of the voltage and current. Notice the value of 削除	29
• Monitoring and Control of Vienna Rectifier Based on the HTTP GUI Page and Ethernet Support (F2838x_only) 追加	49
• sentence: The three-phase current, voltage, and power factors are monitord and shown in HTTP GUI page. The output voltage reference and command signal (Start PFC - cleartrip) can be set in the GUI page and control the Vienna rectifier. 削除	49
• sentence from The ethernet based control and monitoring can be applied to all four incremental builds, but the test in this section is based on INCR_BUILD 3 to The ethernet based monitoring can be applied to all four Incremental builds and the ethernet based control can be applied to INCR_BUILD 3 and INCR_BUILD 4. Test in this section is based on INCR_BUILD 4. 変更	49
• paragraph 追加	50
• INCR_BUILD_3 to INCR_BUILD_4 変更	50
• the step from 1~2 in section 6.4.3.1 to the step from 1~3 in section 6.4.4.1 変更	50

- list item 追加 51
- Follow the step from 1~2 in section 6.4.3.3 to Follow the step from 1~2 in section 6.4.4.2 変更 51
- link 追加 51
- link 追加 51
- which is the per unit value 追加 52
- F2804x to F28004x and F2838x 変更 54
- VIENNA_ 追加 54
- VIENNA_ 追加 54
- VIENNA 追加 54
- additional author 追加 66

Revision C (July 2017) から Revision D に変更 **Page**

- F28004x to Highlighted Products section and updated text 追加 6
- Updated equation definition list in Current Loop Model section 9
- equation definition list to 式 5 in DC Bus Regulation Loop section 追加 11
- note to Bus Capacitor Selection section 追加 14
- TMS320F280049M to Protection (CMPSS) section 追加 17
- 700-800Ω to 500Ω in Base Board Settings section 変更 17
- Corrected duplicate J2 in Key Connectors and Function table 19
- F283779D to F28379D in Step 3 of Control Card Settings section 変更 20
- F283779D to F28379D in Step 3 of Control Card Settings section 変更 20
- controlSUITE to C2000Ware_DigitalPower_SDK in Getting Started Firmware section 変更 21
- note in *Firmware: powerSUITE and Incremental Build Software* 追加 21
- Steps in Opening the Project Inside Code Composer Studio section 変更 21
- Using CLA on C2000 MCU to Alleviate CPU Burden section 追加 25
- iLRef = 1.65 to 0.165 in Step 10 of Running Code (Build 2) section 変更 36
- F28377D to F28379D in Step 13 of Running Code (Build 2) section 変更 36
- DC bus to input AC 変更 42
- 1.34 to 1.32 in Running Code (Build 4) 変更 48
- F28377D to F28379D in Step 8 of Running Code (Build 4) section 変更 48
- Running on CLA section 追加 54
- Power Analyzer Capture of Harmonics at 208-V AC Input, 600-V DC Output, 1.2-kW Load image 追加 58
- Thermal Camera Image of Vienna Rectifier Operating at 2.4-kW, 230-Vrms VL-N Input, 700-V DC Output 追加 64
- Design Files section 変更 65
- Related Documentation section 変更 65

Revision B (June 2017) から Revision C に変更 **Page**

- 図 4. 変更 8
- 図 5 変更 9
- 図 27. 変更 32
- 図 36. 変更 39
- the *TMS320F28004x Piccolo™ Microcontrollers Data Sheet* to Section 6: *Related Documentation* 追加 65

Revision A (February 2017) から Revision B に変更 **Page**

- location of 1 from Section 1.1. 変更 2
- location of 1.1 from Section 1.2. 変更 3
- warning symbol to Warning note. 追加 3
- warning symbols to Caution note. 追加 4

- **High voltage!** There are ***accessible high voltages present on the board***. Electric shock is possible. The board operates at voltages and currents that may cause shock, fire, or injury if not properly handled. Use the equipment with necessary caution and appropriate safeguards to avoid injuring yourself or damaging property. For safety, use of isolated test equipment with overvoltage and overcurrent protection is highly recommended. 追加.................................... 4
- **Hot surface! Contact may cause burns. Do not touch!** 追加.. 4
- title of 2 from System Specifications to System Overview. 変更 4

2016年11月発行のものから更新 **Page**

- TMS320F28377DからTMS320F28379Dに 変更.. 1
- 「リソース」セクションにTMS320F280049Mを 追加 1
- 「リソース」セクションに TMS320F280049M を 追加.. 1
- 2838x, F to title and text 追加 6
- For version 1.3 of the F283779D control card, this means SW3 and SW2 are moved to the end with "." that is, to the left, which puts 3.3-V VDDA as the reference for the ADC. 追加 20
- For version 1.3 of the F283779D control card, this means SW3 and SW2 are moved to the end with "." that is, to the left, which puts 3.3-V VDDA as the reference for the ADC. 追加 20
- 図 20 変更... 26
- If there is no change in the value then make sure the real-time mode is enabled, and the HW is setup correctly. Do not proceed further unless the update is verified. 追加 29
- For example, when Vac is 30 Vrms without switching enabled, the guiVbus will be ~84 V; with switching, the guiVbus will rise up to 140 V. 追加 30
- Make sure all the variables are accurate, that is, guiVrms1/2/3, guiIrms1/2/3, guiPF1/2/3. If any variable is not in line with as shown in 図 25, it points to a hardware issue with the sensing circuit. 追加 30
- 図 27 変更... 32
- 2.7 to 2.0. 変更 33
- 図 37 変更... 39
- 図 44 変更... 45
- and THD will be around 2.5%. 追加... 48

重要なお知らせと免責事項

TI は、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス・デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含むいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、TI 製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した TI 製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとします。

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている TI 製品を使用するアプリケーションの開発の目的でのみ、TI はその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。TI や第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、TI およびその代理人を完全に補償するものとし、TI は一切の責任を拒否します。

TI の製品は、TI の販売条件、または ti.com やかかる TI 製品の関連資料などのいずれかを通じて提供する適用可能な条項の下で提供されています。TI がこれらのリソースを提供することは、適用される TI の保証または他の保証の放棄の拡大や変更を意味するものではありません。

お客様がいかなる追加条項または代替条項を提案した場合でも、TI はそれらに異議を唱え、拒否します。

郵送先住所：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265