

Design Guide: TIDA-01606

11kW、双方向、3相、3レベル(Tタイプ)インバータとPFCのリファレンスデザイン



概要

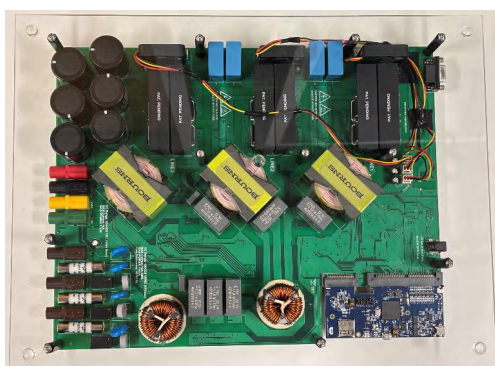
このリファレンスデザインでは、双方向3レベル、3相、SiCベースのアクティブフロントエンド(AFE)インバータおよび力率改善(PFC)段を実装する方法の概要を紹介いたします。このデザインでは、90kHzのスイッチング周波数とLCL出力フィルタを使用して、磁気素子のサイズを縮小しています。98.6%のピーク効率を達成しています。このデザインは、DQドメインに完全な3相AFE制御を実装する方法を示しています。この双方向コンバータにより、DC高速充電とvehicle-to-grid(V2G)の両方のアプリケーションが可能になります。

リソース

TIDA-01606	デザインフォルダ
TMS320F28379D 、 TMS320F280039C	プロダクトフォルダ
UCC21710 、 UCC5350 、 AMC3306M05	プロダクトフォルダ
TMCS1123 、 AMC0330R 、 AMC0381D	プロダクトフォルダ
OPA4388 、 OPA397 、 UCC14341B	プロダクトフォルダ
UCC33421 、 C2000WARE	プロダクトフォルダ
TMDSCNCD28379D	ツールフォルダ
TMDSCNCD280039C	ツールフォルダ



テキサス・インスツルメンツの™ E2E サポートエキスパートにお問い合わせください。

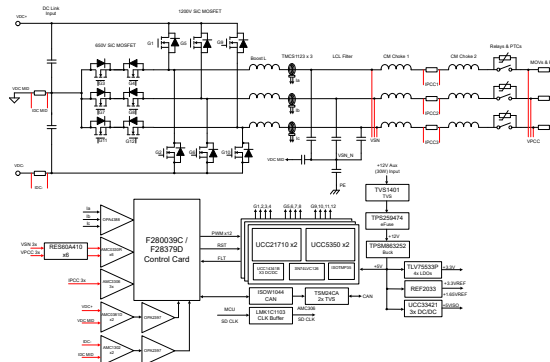


特長

- 定格公称入力電圧および最大入力電圧 800V および 900V DC、最大電力 11kW、11kVA (400V_{L-L} AC、50Hz または 60Hz 時)
- スwitching周波数 90kHz のコンパクトな出力 LCL フィルタ
- 全負荷時の 2.5% 未満の出力電流全高調波歪み (THD)
- 強化絶縁付きの絶縁ドライバ UCC21710 により高電圧の SiC MOSFET を駆動し、UCC5350 により中電圧の SiC MOSFET を駆動
- 負荷電流制御および監視を目的とする TMCS1123 を使用した絶縁型電流検出
- 三角関数演算ユニット (TMU) を使用してフェーズ ロックループ (PLL) の計算を高速化するデジタル制御用 TMS320F28379D および TMS320F280039C 制御カード、保護機能を実装するためのコンパレータサブシステム、制御ループをコプロセッサにオフロードする制御補償器アクセラレータ (CLA)

アプリケーション

- DC 高速充電ステーション
- DC ウォールボックス充電器
- オンボードチャージャ (OBC) およびワイヤレスチャージャ
- 電力変換システム (PCS)
- スtring インバータ
- セントラル インバータ



1 システムの説明

現代の商用スケールのソーラー インバータは、次の 2 つの面で革新を実現しており、市場に出回る製品の小型化および高効率化を達成しています。

- より電圧の高いソーラー アレイへの移行
- オンボード磁石のサイズ削減

アレイからの DC 電圧を 1000V または 1500V に上げることで、同じ電力レベルを保持しながら電流を削減できます。この電流の削減により、設計に必要な電力導通デバイスの銅箔を削減し、サイズを縮小できます。di/dt が減少すると、電気部品へのストレスも減少します。ただし、1kV を超える DC 電圧を持続できるようにするのは設計が困難であり、耐性のある部品を見つけるのも容易ではありません。

高電圧ソーラー アレイによって生成される電圧ストレスを補償するため、ソーラー インバータの新しいトポロジが設計されました。従来のハーフブリッジは、各スイッチング デバイスの入力電圧全体をブロックします。スイッチ式のブロッキング部品と導通部品を追加することで、デバイス全体のストレスを大幅に低減できます。このリファレンス デザインは、3 レベルコンバータを実装する方法を示します。より高いレベルのコンバータも可能で、電圧対応能力がさらに向上します。

パワー コンバータのスイッチング速度を上げることで、ソーラー エレクトロニクスの電力密度も増加できます。この設計で示すように、スイッチング速度を多少増加するだけでも、出力フィルタ段全体のサイズ要件が小さくなります。これは、設計サイズの主要な要因です。

従来のスイッチング デバイスでは、高電圧のスイッチング速度 (デバイスの dV/dt 能力) に制限があります。この低速のランプアップおよびランプダウンにより、デバイスがスイッチング状態にある時間が長くなるため、導通損失が増加します。このようにスイッチ時間が長くなると、貫通電流や短絡を防止するために、制御システムに必要なデッドタイムも長くなります。この問題に対する答えが、電子移動度の高い SiC や GaN デバイスなどの新しいスイッチング半導体技術において開発されました。このリファレンス デザインでは、SiC MOSFET とテキサス・インスツルメンツの SiC ゲートドライバテクノロジーを組み合わせ使用し、電力密度を向上できることを示します。

同様に、オンボードチャージャ (OBC) にも、より大電力のチャージャ (11kW および 22kW) がますます必要となつていきます。これには 3 相 PFC が必要であるため、この設計では DQ 制御を使用した 3 相 PFC を実装し、完全な制御ループモデルを示します。

1.1 主なシステム仕様

表 1-1. 主なシステム仕様

パラメータ	仕様	詳細
出力電力	11kW	セクション 2.3
出力電圧	3 相 400V _{RMS} (最大 V _{L-L})	セクション 2.3
出力周波数	50Hz または 60Hz	セクション 2.3
出力電流	16A _{RMS} (最大値)	セクション 2.3
公称入力電圧	800-V DC	セクション 2.3
入力電圧範囲	600V ~ 900V DC	セクション 2.3
インバータ スwitching 周波数	50~90kHz	セクション 2.3
効率	98.6%	セクション 2.3.1.5
THD	< 3% (11kW)	
電力密度	2.2kW/L+	
寸法	27cm × 35cm × 5cm	

2 システム概要

2.1 ブロック図

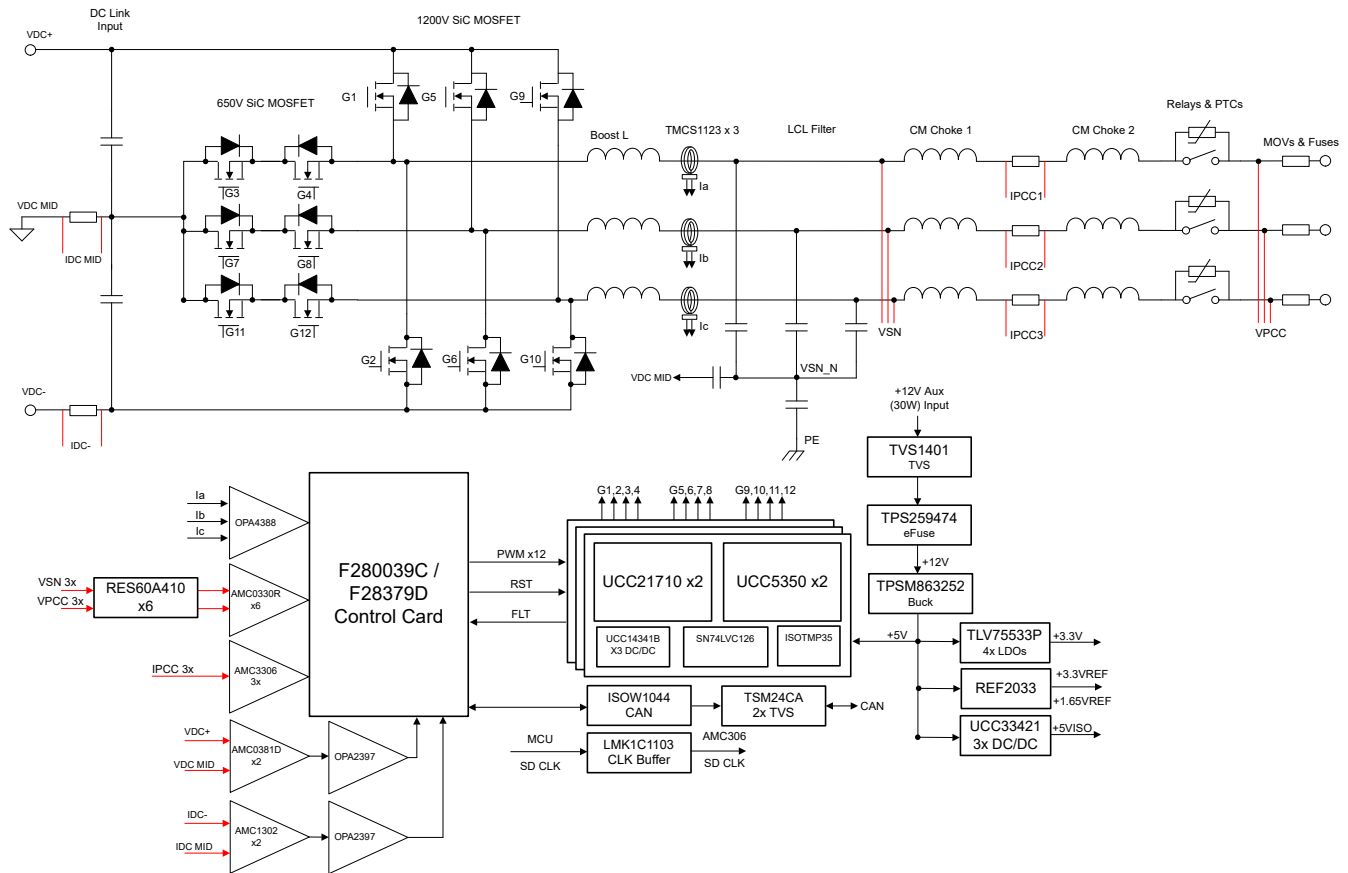


図 2-1. TIDA-01606 のブロック図

このリファレンス デザインは、相互に通信する 2 枚の個別のボードで構成されています。以下のボードが連携して動作し、この 3 相インバータのリファレンス デザインを形成します。

- スwitching デバイス、ゲートドライバ、LCL フィルタ、センシング電子回路、および電源構造をすべて含む電源ボード
- DSP をサポートする TMS320F28379D 制御カードまたは TMS320F280039C

2.2 主な使用製品

2.2.1 UCC21710

UCC21710 デバイスは、絶縁型ゲート・バイポーラ・トランジスタ (IGBT) および SiC MOSFET 用の 5.7kV_{RMS} 強化絶縁ゲート・ドライバで、10A のソース電流と 10A のシンク電流を供給する分割出力を備えています。入力側は、単一の 3V ~ 5.5V 電源で動作します。出力側では、最小 13V から最大 33V までの電源電圧範囲が使用できます。2 つの相補 CMOS 入力により、ゲート・ドライバの出力状態が制御されます。伝搬時間が 130ns と短いため、出力ステージを正確に制御できます。UCC21710 には過電流検出を使用した短絡保護機能が内蔵されており、SiC MOSFET の保護に必要な高速応答時間を実現しています。

- 最小同相過渡耐性 (CMTI): 150kV/μs
- 分割出力により 10A のピーク・ソース電流および 10A のピーク・シンク電流を供給
- 短い伝搬遅延: 90ns (標準値)、130ns (最大値)
- 4A のアクティブ・ミラー・クランプ
- 出力短絡クランプ
- 短絡時のソフト電源オフ (STO) 機能

- 不飽和化検出時のフォルト・アラームは FLT により通知され、RST によりリセット
- 入出力低電圧誤動作防止 (UVLO): レディ (RDY) ピンによる標示付き
- 低電源またはフローティング入力時の、アクティブ出力プルダウンおよびデフォルト Low 出力
- 入力電源電圧: 2.25V~5.5V
- 出力ドライバ供給電圧: 15V~30V
- CMOS 互換の入力
- 40ns 未満の入力パルスとノイズ過渡を除去
- 動作温度: -40°C~+150°C (周囲)
- 絶縁サージ耐久電圧: 12800V_{PK}

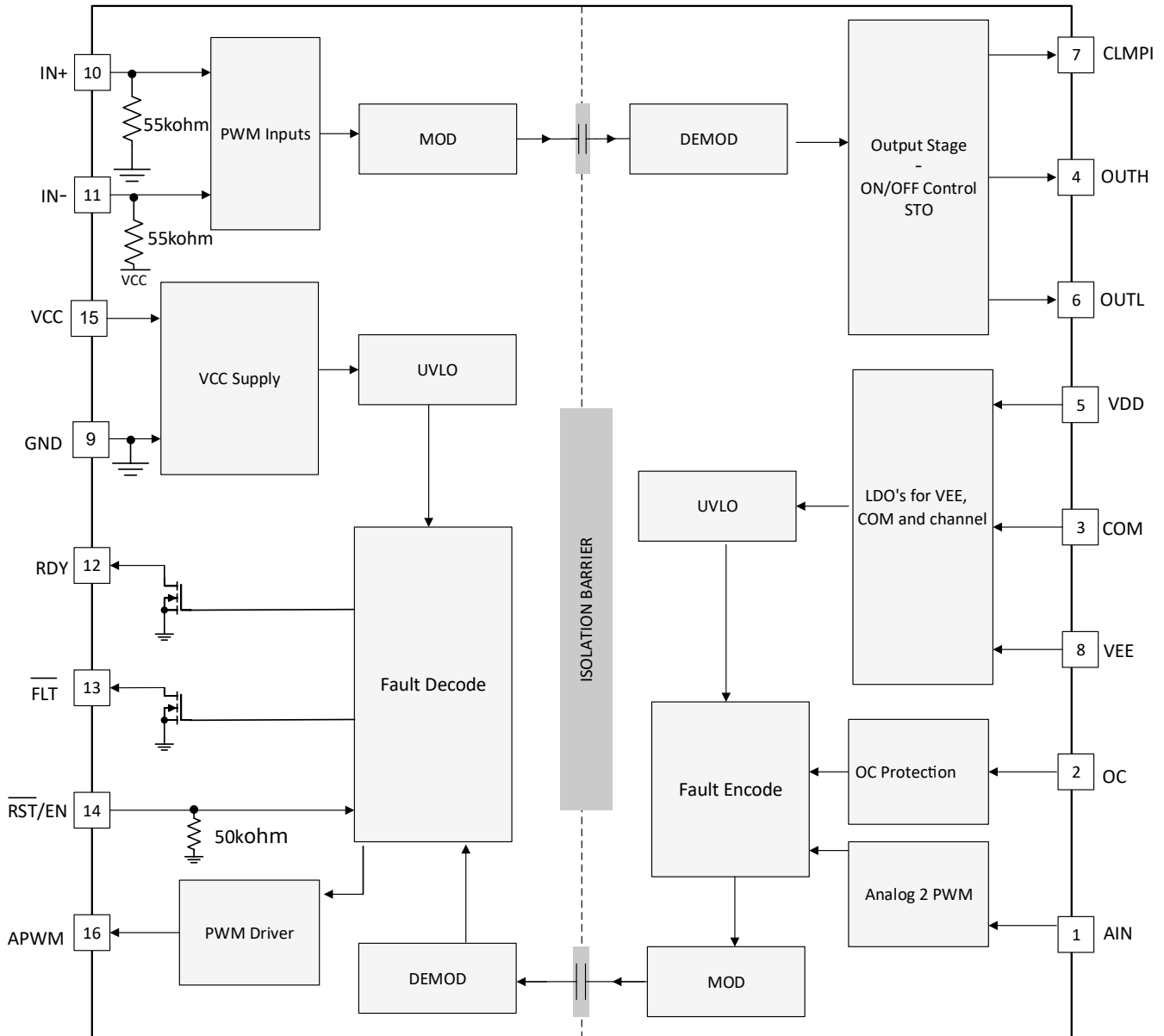


図 2-2. UCC21710 の機能ブロック図

2.2.2 UCC5350

UCC53x0 は小型のシングル チャネル絶縁 IGBT、SiC、および MOSFET ゲートドライバのファミリーで、クラス最高の絶縁定格を持ち、ピン配置構成および駆動強度の異なるバリエーションが用意されています。

UCC53x0 は、8 ピンの SOP (DWV) パッケージで供給されます。このパッケージは沿面距離と空間距離が 8.5mm で、最高 5kV_{RMS} の絶縁電圧に対応でき、強化絶縁が必要なアプリケーションに適しています。これら各種のオプションと広い電力範囲から、UCC53x0 ファミリーはモータドライブや産業用電源に適しています。

- 3V~15V の入力電源電圧
- 出力ドライバ電源電圧: 13.2V~33V
- 機能オプション:
 - 分割出力 (UCC5320S および UCC5390S)
 - MOSFET コレクタに対する UVLO (UCC5320E および UCC5390E)
 - ミラー クランプ オプション (UCC5310M および UCC5350M)
- 負の 5V に対応可能な入力ピン
- UCC5320S、UCC5320E、UCC5310M の伝搬遅延 60ns (標準値)
- CMTI: 100kV/μs (最小値)
- 絶縁サージ耐久電圧: 4242V_{PK}
- 安全関連認証:
 - DIN V VDE V 0884-10 および DIN EN 61010-1 に準拠した 4242V_{PK} の絶縁 (予定)
 - UL 1577 に準拠した絶縁耐性: 3000V_{RMS} (1 分間) (予定)
 - CSA 部品承認通達 5A、IEC 60950-1 および IEC 61010-1 最終機器標準 (予定)
 - GB4943.1-2011 準拠の CQC 認定 (予定)
- すべてのピンで 4kV ESD 保護
- CMOS 入力
- 8 ピンのナロー ボディ SOIC パッケージ
- 動作温度: -40°C~+125°C (周囲)

2.2.3 TMS320F28379D

Delfino™ TMS320F2837xD は優れた 32 ビット浮動小数点マイクロコントローラ・ユニット (MCU) で、産業用ドライブおよびサーボ・モータ制御、ソーラー・インバータおよびコンバータ、デジタル電源、輸送、電源ライン通信など、高度な閉ループ制御アプリケーション用に設計されています。デジタル電源および産業用ドライブ向けの開発パッケージ一式が、powerSUITE および DesignDRIVE イニシアチブの一部として利用できます。Delfino 製品ラインは TMS320C2000™ ポートフォリオでは新しいものではありませんが、F2837xD はシステム性能を大幅に向上させる新しいデュアルコア C28x アーキテクチャをサポートしています。また、内蔵のアナログおよび制御ペリフェラルにより、設計者は制御アーキテクチャを統合し、ハイエンド・システムでマルチプロセッサを不要にできます。

- デュアルコア・アーキテクチャ:
 - 2 つの TMS320C28x 32 ビット CPU
 - 200MHz
 - IEEE 754 単精度浮動小数点ユニット (FPU)
 - 三角関数演算ユニット (TMU)
 - ビタビ / 複素演算ユニット (VCU-II)
- 2 つのプログラム可能な制御補償器アクセラレータ (CLA)
 - 200MHz
 - IEEE 754 単精度浮動小数点命令
 - メイン CPU と独立にコードを実行
- オンチップ・メモリ
 - 512KB (256kW) または 1MB (512kW) のフラッシュ (ECC 保護)
 - 172KB (86kW) または 204KB (102kW) の RAM (ECC 保護またはパリティ保護)
 - デュアル・ゾーンのセキュリティでサードパーティーの開発をサポート
- クロックおよびシステム制御:
 - 2 つの内部ゼロ・ピン 10MHz 発振器
 - オンチップの水晶発振器
 - ウィンドウ付きウォッチドッグ・タイマ・モジュール
 - クロック消失検出回路
- コア 1.2V、I/O 3.3V の設計

- システム・ペリフェラル:
 - ASRAM および SDRAM をサポートする 2 つの外部メモリ・インターフェイス (EMIF)
 - デュアル 6 チャンネルの DMA (Direct Memory Access) コントローラ
 - 個別にプログラム可能な最大 169 本の多重化された汎用入出力 (GPIO) ピン (入力フィルタ付き)
 - 拡張ペリフェラル割り込みコントローラ (ePIE)
 - 外部ウェイクアップに対応した複数の低消費電力モード (LPM) をサポート
- 通信ペリフェラル:
 - USB 2.0 (MAC + PHY)
 - 12 ピンの 3.3V 互換ユニバーサル・パラレル・ポート (uPP) インターフェイスをサポート
 - 2 つの CAN (Controller Area Network) モジュール (ピンでブート可能)
 - 3 つの高速 (最高 50MHz) SPI ポート (ピンでブート可能)
 - 2 つのマルチチャンネル・バッファ付きシリアル・ポート (McBSP)
 - 4 つのシリアル通信インターフェイス (SCI/UART) (ピンでブート可能)
 - 2 つの I²C インターフェイス (ピンでブート可能)
- アナログ・サブシステム:
 - 最大 4 つの A/D コンバータ (ADC):
 - 16 ビット・モード
 - 各 1.1MSPS (最大 4.4MSPS のシステム・スループット)
 - 差動入力
 - 最大 12 の外部チャンネル
 - 12 ビット・モード
 - 各 3.5MSPS (最大 14MSPS のシステム・スループット)
 - シングルエンド入力
 - 最大 24 の外部チャンネル
 - 各 ADC に単一のサンプルおよびホールド (S/H)
 - ADC 変換の後処理をハードウェアに内蔵:
 - 飽和オフセット較正
 - 設定点からの誤差の計算
 - 高、低、およびゼロクロス比較、割り込み機能付き
 - トリガからサンプルまでの遅延の捕捉
 - 8 つのウィンドウ付きコンパレータ、12 ビットのデジタル / アナログ・コンバータ (DAC) リファレンス付き
 - 3 つのバッファ付き 12 ビット DAC 出力
- 拡張制御ペリフェラル:
 - 拡張機能付きの 24 のパルス幅変調器 (PWM) チャンネル
 - 16 の高分解能パルス幅変調器 (HRPWM) チャンネル
 - 8 つの PWM モジュールの A チャンネルと B チャンネルの両方が高分解能
 - デッドバンドのサポート (標準と高分解能の両方)
 - 6 つの拡張キャプチャ (eCAP) モジュール
 - 3 つの拡張直交エンコーダ・パルス (eQEP) モジュール
 - 8 つのシグマ-デルタ・フィルタ・モジュール (SDFM) 入力チャンネル (チャンネルごとに 2 つの平行・フィルタ):
 - 標準 SDFM データ・フィルタリング
 - コンパレータ・フィルタによる範囲外に対する迅速なアクション

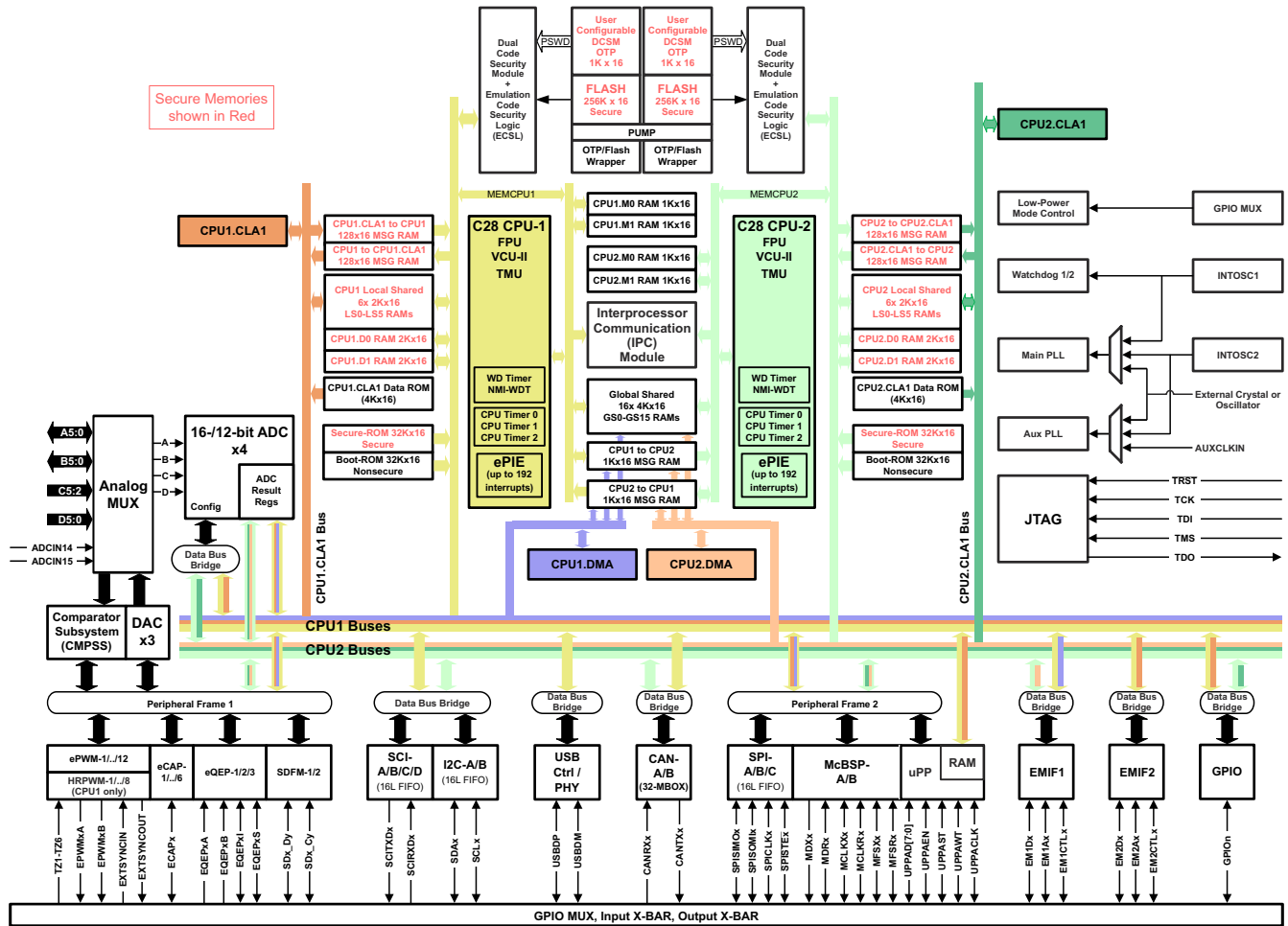


図 2-3. TMS320F28379D の機能ブロック図

2.2.4 AMC3306M05

AMC3306M05 は、シャントを用いた電流測定に最適化された高精度絶縁型デルタシグマ ($\Delta\Sigma$) 変調器です。完全に統合された絶縁型 DC/DC コンバータのおかげで、本デバイスの低電圧側から電力を供給する単一電源動作が可能であるため、スペースに制約があるアプリケーション向けのユニークな解決策として活用できます。その容量性強化絶縁バリアは、VDE V 0884-11 および UL1577 により認証済みであり、最大 1.2kV_{RMS} の使用電圧に対応しています。

2.2.5 OPA4388

OPAx388 (OPA388、OPA2388、OPA4388)シリーズの高精度オペアンプは非常にノイズが低く、セトリングが高速で、ゼロドリフト、ゼロクロスオーバーのデバイスで、レールツーレールの入力および出力動作を行います。これらの機能に加え、非常に優れた AC 性能と、わずか $0.25\mu\text{V}$ のオフセット、温度範囲全体で $0.005\mu\text{V}/^\circ\text{C}$ のドリフト係数により、OPAx388 は高精度のアナログ / デジタル コンバータ (ADC) の駆動や、高分解能のデジタル / アナログ コンバータ (DAC) の出力のバッファリングに理想的です。この設計により、ADC の駆動時に直線性の劣化がなく、優れた性能を発揮します。OPA388 (シングルバージョン) は VSSOP-8、SOT23-5、SOIC-8 パッケージで供給されます。OPA2388 (デュアルバージョン) は VSSOP-8 および SO-8 パッケージで供給されます。OPA4388 (クワッドバージョン) は TSSOP-14 および SO-14 パッケージで供給されます。すべてのバージョンは、産業用温度範囲 ($-40^\circ\text{C}\sim+125^\circ\text{C}$) で仕様が規定されています。

2.2.6 TMCS1123

TMCS1123 は、業界をリードする絶縁性と精度を備えたガルバニック絶縁ホール効果電流センサです。入力電流に比例する出力電圧により、優れた直線性と、あらゆる感度オプションで低ドリフトを実現しています。ドリフト補償を内蔵した高精度のシグナルコンディショニング回路は、システムレベルのキャリブレーションなしで、温度範囲と寿命全体にわたり 1.4% 未満の最大感度誤差を達成しており、室温キャリブレーションを 1 回のみ実行した場合には、寿命および温度ドリフトを考慮しても 0.9% 未満の最大感度誤差を達成します。

2.2.7 AMC0330R

AMC0x30R-Q1 は、 $\pm 1\text{V}$ 、高インピーダンス入力とシングルエンドのレシオメトリック出力を備えた、高精度、電氣的絶縁型アンプです。高インピーダンス入力は、高インピーダンスの抵抗分圧器や出力抵抗の高い他の電圧信号源と接続するよう最適化されています。

2.2.8 AMC0381D

AMC0381D-Q1 は、高電圧 DC、高インピーダンス入力、固定ゲイン差動出力を備えた高精度、電氣的絶縁型アンプです。入力は、高電圧信号源に直接接続できる設計を採用しています。

2.2.9 UCC14341

UCC14341 は、IGBT または SiC ゲートドライバへの電力供給を目的として設計された高絶縁電圧 DC/DC モジュールです。このモジュールは、変圧器と DC/DC コントローラを独自のアーキテクチャで統合しており、非常に低い放射で高効率を実現します。高精度の出力電圧により、パワー デバイスのゲートに過大なストレスを与えずにシステム効率が向上します。

2.2.10 UCC33421

UCC33421-Q1 は、トランス技術を内蔵し、 1.5W の絶縁出力電力を供給するように設計された、車載認定済み DC/DC パワーモジュールです。 $4.5\text{V}\sim 5.5\text{V}$ の入力電圧動作範囲に対応し、 5.0V の安定化出力電圧を供給し、 5.5V のヘッドルームを選択可能です。

2.3 システム設計理論

2.3.1 3 相 T タイプ インバータ

2.3.1.1 アーキテクチャの概要

3 レベル T タイプ インバータが開発された経緯を理解するには、従来の 2 レベル インバータに関するある程度の背景知識が必要です。図 2-4 は、このアーキテクチャの代表的な実装を示しています。

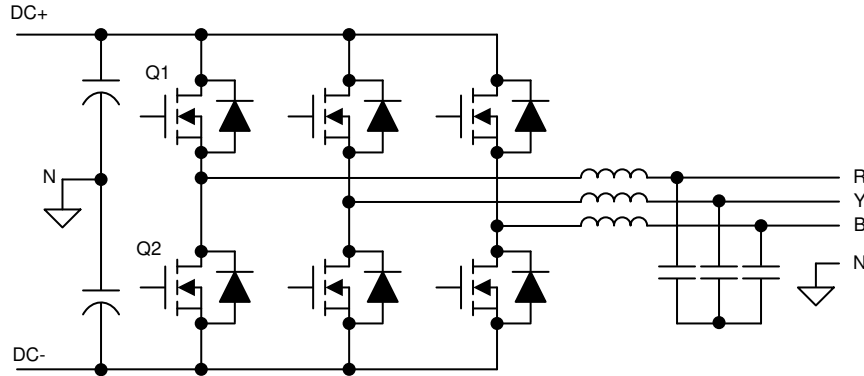


図 2-4. 2 レベル、3 相インバータ アーキテクチャ

分析を簡潔にするため、1 つのレッグを分離して考えます。

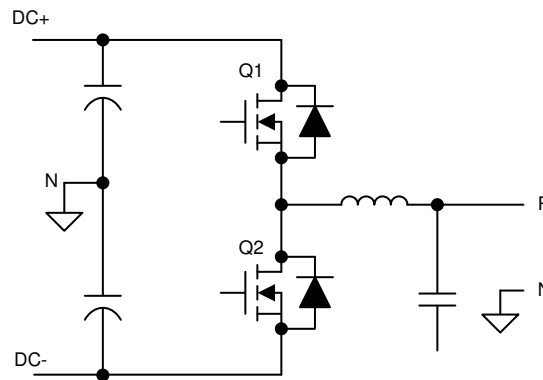


図 2-5. 2 レベル、単相インバータのレッグ

この例では、2 つのスวิตチング デバイスをペアとして使用することで、他の相とは無関係に、4 つの導通状態を持つことができます。

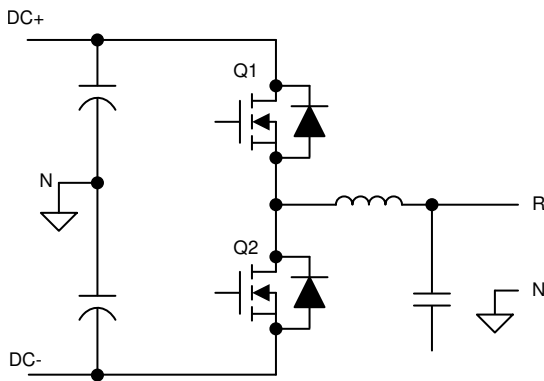


図 2-6. Q1 と Q2 がオフ

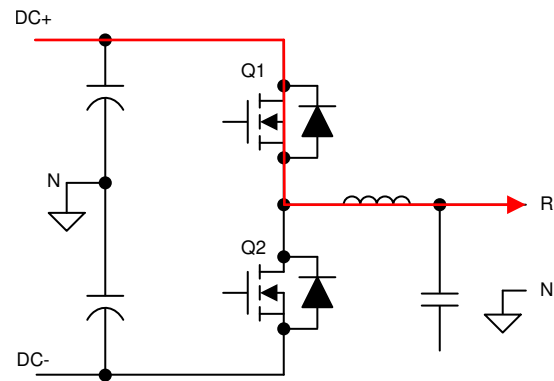


図 2-7. Q1 がオン、Q2 がオフ

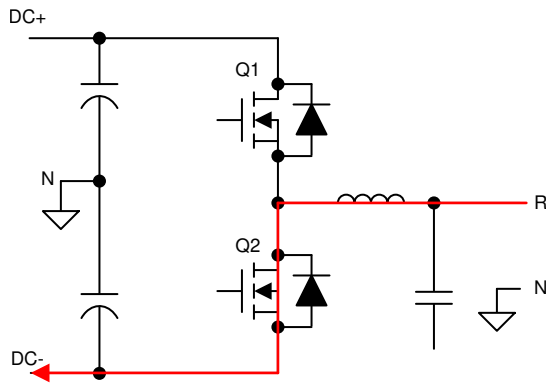


図 2-8. Q1 がオフ、Q2 がオン

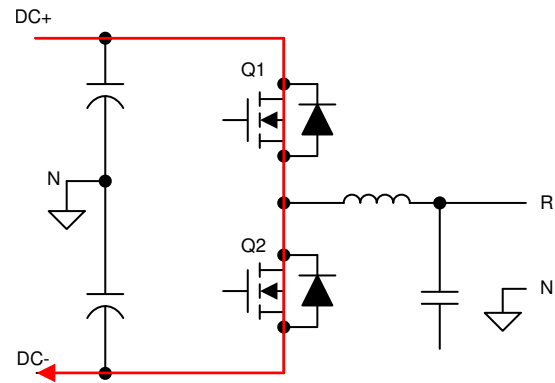


図 2-9. Q1 と Q2 がオン (無効)

インバータを流れる電流パスを見るとわかるように、各スイッチング デバイスは DC+ と DC- の間の DC リンク電圧を完全にブロックできる必要があります。従来型の低電圧システム (600V 未満) では、一般的な既製の IGBT を使用してこれを簡単に達成できます。ただし、パワー エレクトロニクス一般的な傾向として見られるように、電力スループットを向上させるために電流を増加せずに DC リンク電圧を高くすると、この制限がサポートされる電圧範囲の上限となります。

また、従来型の IGBT では、電圧を高くするとスイッチング損失も増加します。これらのデバイスがより高い電圧に対応できたとしても、低い dV/dt がさらに悪化します。この dV/dt により 1 つのデバイスがオンからオフ (またはオフからオン) にどれだけ迅速に遷移できるかが決まり、これらの各状態間のデッド タイムが決定されます。スイッチ時間またはデッドタイムが長くなると、スイッチが完全に導通状態である時間が短くなり、効率が低下します。

2 レベル インバータにはこれら 2 つの主な欠点があるため、この設計の実装が推進されます。

標準的な 2 レベル インバータからの次のステップは、T タイプの 3 レベル インバータです。このタイプは、スイッチ ノードと、バルク入力コンデンサによって作成される DC リンクの中性点との間に 2 つのスイッチング デバイスを連続させて挿入することで実装します。これら 2 つのスイッチ デバイスは、どちらか一方のオン / オフを切り替えることで電流フローを制御できるように、共通のエミッタ構成で配置されています。この構成により、ゲートとエミッタ間の電圧が同一に参照されるため、両方のスイッチ間で同じバイアス電源を共有することもできます。図 2-10 に、この実装の簡略図を示します。

注

E6 ハードウェア ミドル スイッチは、共通のソース スイッチとして設定されます。ただし、E7 ハードウェア ミドル スイッチは、共通のドレイン スイッチとして設定されます。次の T タイプの例の図は、共通ソース構成を使用しています。

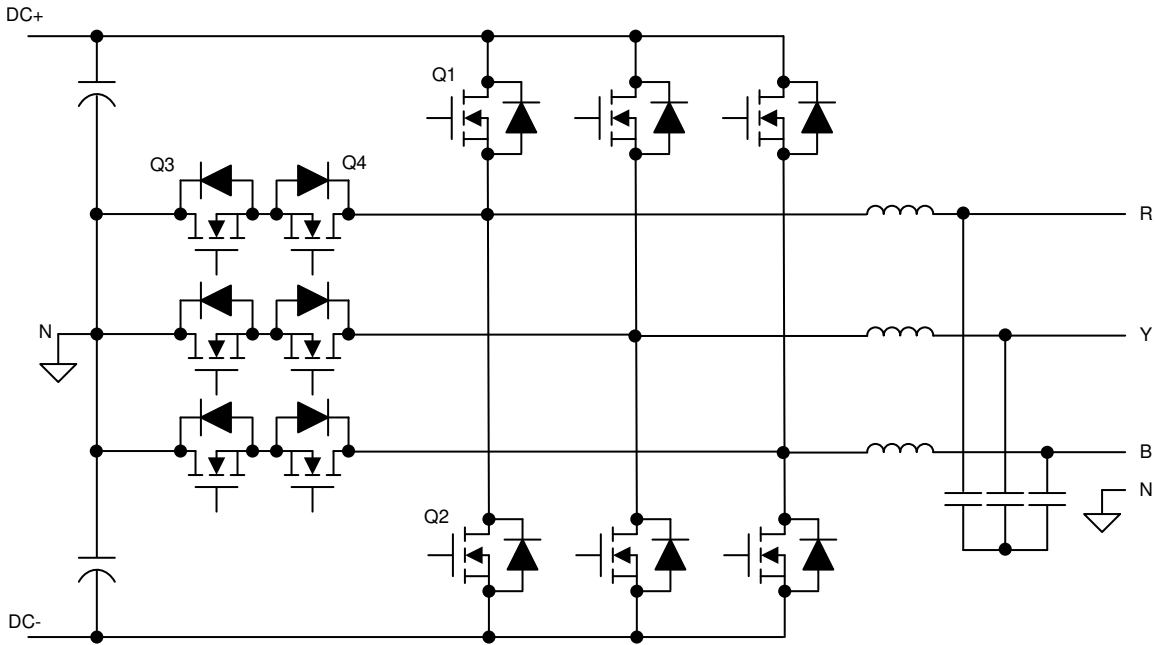


図 2-10. 3 レベル T タイプ、3 相インバータアーキテクチャ

このアーキテクチャの利点を理解しやすくするため、ここでもインバータの 1 つのレッグのみを分離して考えてみます。

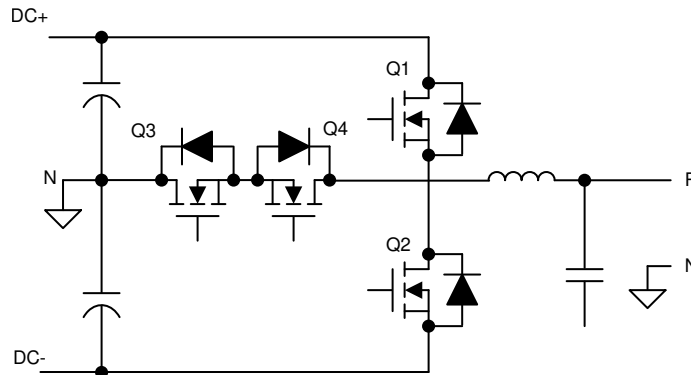


図 2-11. 3 レベル T タイプ、単相インバータレッグ

スイッチング デバイスを 2 個追加するとシステムの制御が複雑になりますが、さまざまな変調点における電流フローを同じプロセスを使用して評価すると、このアーキテクチャの利点を理解できます。また、簡略化した整流方式を示すことで、T タイプ インバータの制御が従来の 2 レベル アーキテクチャに比べてそれほど複雑でないことがわかります。

1 つのレッグには、DC+、DC-、N の 3 つの接続状態があります。この接続は、それぞれ Q1 を閉じる、Q3 と Q4 を閉じる、Q2 を閉じることで実現できます。ただし、この方式はシステムの電流パスによって異なります。DC+ 接続では Q1 と Q3 を閉じ、中性 (N) 接続では Q2 と Q4 を閉じ、DC- 接続では Q2 と Q4 を閉じることができます。この方式は、次の図に示すように、電流の方向とは無関係に動作します。

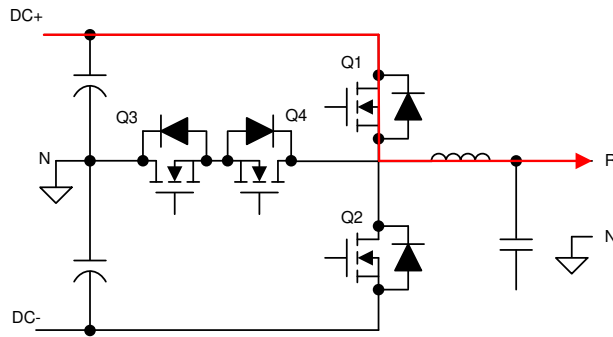


図 2-12. Q1 オン、Q2 オフ、Q3 オン、Q4 オフ

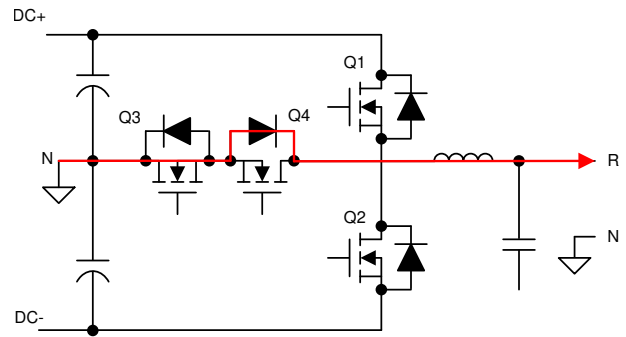


図 2-13. Q1 オフ、Q2 オフ、Q3 オン、Q4 オフ

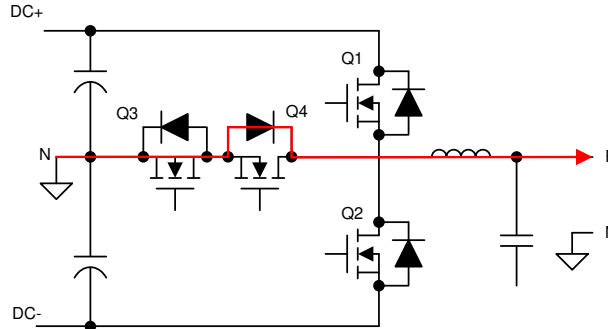


図 2-14. Q1 オフ、Q2 オフ、Q3 オン、Q4 オン

この例では、まず Q1 と Q3 を閉じて出力位相を DC+ に接続し、システムから電流を出力します。N 接続に移行するには、Q1 を開き、デッドタイム遅延後に Q4 を閉じます。この設定により、電流は自然に Q3 と Q4 のダイオードを通して流れます。

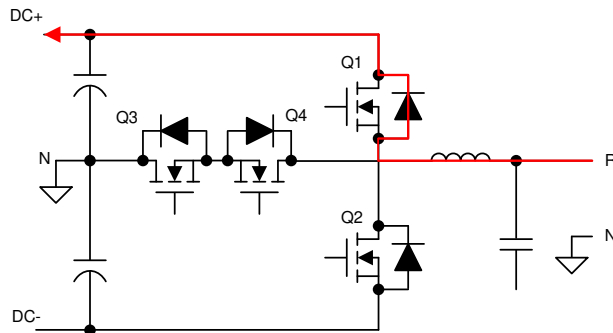


図 2-15. Q1 オン、Q2 オフ、Q3 オン、Q4 オフ

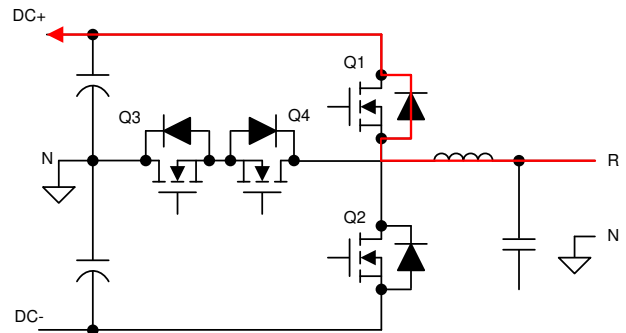


図 2-16. Q1 オフ、Q2 オフ、Q3 オン、Q4 オフ

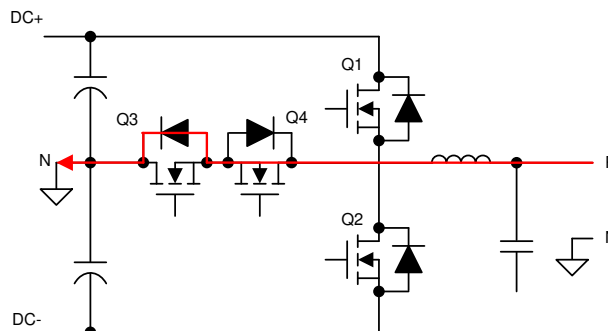


図 2-17. Q1 オフ、Q2 オフ、Q3 オン、Q4 オン

負の電流にも、同じシーケンスを使用できます。Q4 を閉じると、電流は Q4 を通り、Q1 のダイオードではなく Q3 のダイオードを流れます。

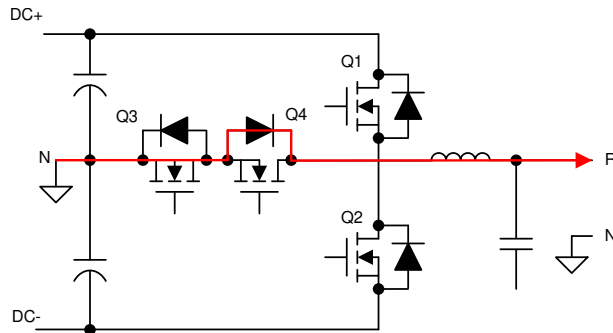


図 2-18. Q1 オフ、Q2 オフ、Q3 オン、Q4 オン

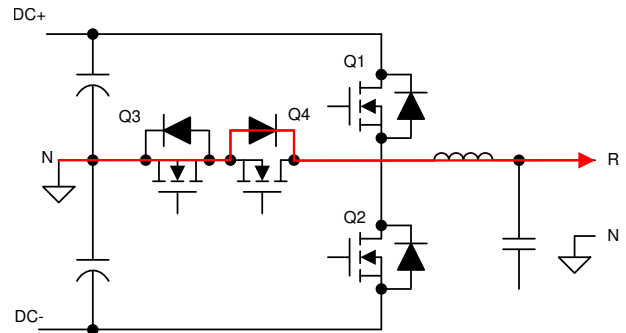


図 2-19. Q1 オフ、Q2 オフ、Q3 オン、Q4 オフ

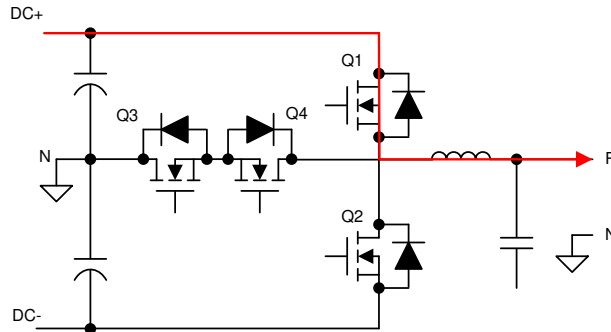


図 2-20. Q1 オン、Q2 オフ、Q3 オン、Q4 オフ

出力レグを N から DC+ に正の電流で接続すると、同様の自然電流を観測できます。Q3 と Q4 を閉じ、完全な N 接続で開始します。Q4 はオフになりますが、電流は関連するダイオードを流れます。ここで Q1 を閉じると、電流は N から DC+ に自然に切り替わります。

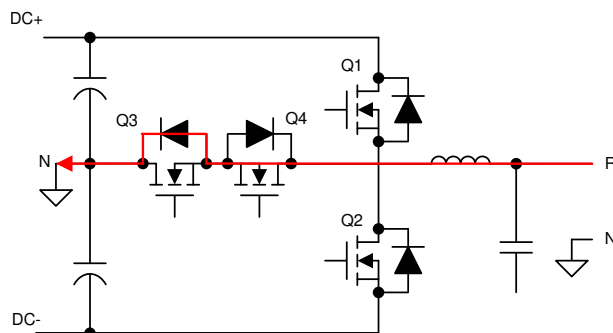


図 2-21. Q1 オフ、Q2 オフ、Q3 オン、Q4 オン

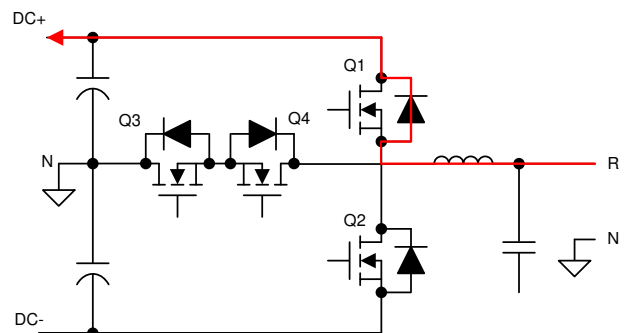


図 2-22. Q1 オフ、Q2 オフ、Q3 オン、Q4 オフ

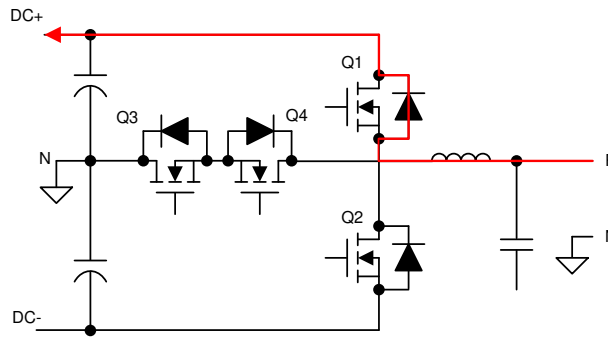


図 2-23. Q1 オン、Q2 オフ、Q3 オン、Q4 オフ

先ほど説明した負の電流で DC+ 接続から N 接続に移行する例と同様に、正の電流に対して同じ方式を使用できます。Q3 と Q4 を閉じた状態で開始し、電流が N に向かって流れるようにします。Q4 を開くと、電流が Q1 のダイオードを通して流れます。最後に Q1 は閉じると、電流は同じ方向に流れ続けます。

これら 4 つの遷移状態 (順方向電流と逆方向電流で DC+ から N、N から DC+) はすべて、2 つの簡単なスイッチング方式を共有します。これは、Q2 を介する DC- との遷移にも当てはまります。この方式をすべてのスイッチング サイクルで維持することで、貫通電流を防止するのに必要なのはスイッチング イベント間の単純なデッドゾーン遅延のみとなります。追加の保護機能も制御ソフトウェアに比較的簡単に追加できます。

この変調方式のもう 1 つの利点は、Q3 と Q4 が同時にスイッチングされないことです。この利点により、デバイスの電圧ストレスが減少すると同時にバイアス電源の電力定格を低くできるので、これらのデバイスを効果的に駆動できます。先ほど説明したように、Q3 と Q4 は、2 つのドライバではなく 1 つのドライバ用のサイズの単一電源を共有できます。

Q1 と Q2 は、従来のアーキテクチャにおけるインバータと同様に、DC リンク電圧を完全にブロックする必要があります。より高い DC バス電圧を使用するには、フル電圧 FET を配置する必要がありますが、インバータは隣接して配置され、同時にスイッチングすることはないため、センター レッグの 2 つのスイッチの定格を低くすることができます。

2.3.1.2 LCL フィルタの設計

送電網に電力を伝送するシステムは、高調波成分に関する特定の出力仕様を満たす必要があります。最新の太陽光発電インバータのような電圧ソース システムでは通常、高次 LCL フィルタによって十分な高調波減衰が実現されるだけでなく、単純なフィルタ設計よりも設計全体のサイズを小さくできます。ただし、高次であることから、設計では注意して共振を制御する必要があります。図 2-24 に、代表的な LCL フィルタを示します。

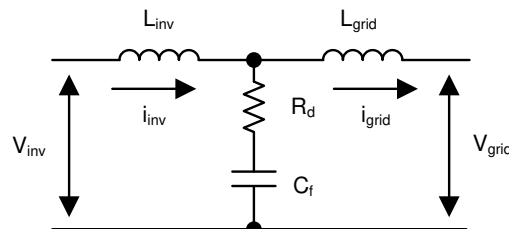


図 2-24. LCL フィルタのアーキテクチャ

このリファレンス デザインのように SiC MOSFET を使用する主な利点の 1 つは、従来の Si ベースのスイッチング素子と比較して、電力段のスイッチング周波数を大幅に高くできることです。スイッチング周波数を高くすると、インバータの出力フィルタの共振設計に直接影響するため、これを考慮する必要があります。このスイッチ周波数を中心にフィルタが正しく設計されていることを確認するため、この設計では既知の数学モデルを使用します。

主な部品はインバータ インダクタ (L_{inv}) で、式 1 を使用して求めることができます。

$$L_{inv} = \frac{V_{DC}}{8 \times f_{SW} \times I_{grid_rated} \times \%ripple} \quad (1)$$

ここで、

- f_{SW} は、PWM スイッチング周波数です
- I_{grid_rated} は、送電網の RMS 電流定格です
- $\% ripple$ は、定格グリッド電流のリプル電流パーセンテージです

1 次側の EMI フィルタ コンデンサ C_f のサイズは、式 2 で決定されます。

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

ここで、

- $\% Q_{rated}$ は、 C_f コンデンサを制限するための定格無効電力のパーセンテージです
- f_{grid} は、送電網の電気周波数です
- V_{grid} は、送電網の相電圧です

式 3 に計算例を示します。

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

$$24nF < C_f < 9.2\mu F$$

E7 では、 C_f に 4.7 μF を選択しました。これは、カットオフ周波数の約 6.4kHz に相当します。

2.3.1.3 インダクタの設計

注

E7 で使用されている昇圧インダクタは、カスタム Bourns インダクタです。昇圧インダクタの重要な仕様は、インダクタンス、電流定格、寸法、および動作温度です。TI では、カスタムインダクタの設計を支援するために、経験豊富な磁気メーカー (Bourns Inc. など) と連携することを推奨しています。

2.3.1.4 SiC MOSFET の選択

[アーキテクチャの概要](#)で示したように、メイン スイッチング デバイスはスイッチング電圧の全範囲をサポートする必要があります。この設計の 1000V DC リンク電圧をサポートするには、1200V の FET を使用します。ただし、この電圧では、次の要因により SiC への移行が必要になります。

- 1200V SiC MOSFET のスイッチング速度は、従来型の IGBT より大幅に高速なので、スイッチング損失が低減されます。
- SiC MOSFET の逆回復充電が非常に小さいため、電圧および電流のオーバーシュートが低減されます。
- 導通損失が減少するため、全負荷での温度依存性が低くなります。

中央のスイッチには、DC リンク電圧の半分 (この設計では 500V) しか印加されません。そのため、650V デバイスを使用できます。これらの同じ機能により、フル SiC 設計で最高の性能が得られます。この設計では、逆回復損失と電圧オーバーシュートにより、デバイスの選択が制限されます。そのため、1200V の SiC MOSFET + 650V の MOSFET 設計を使用します。

導通損失は主に、1200V の SiC MOSFET の $R_{DS(on)}$ と、650V の SiC MOSFET の $R_{DS(on)}$ により決定されます。75m Ω の SiC デバイスは高温性能に優れており、接合部温度 150°C での $R_{DS(on)}$ の増加はわずか 40% です。データシートの高温 I-V 曲線を使用して、デバイスの導通損失を計算します。

スイッチング損失は、スイッチング周波数と各スイッチング過渡のスイッチング エネルギーの関数であり、スイッチング エネルギーはスイッチング過渡時のデバイスの電流と電圧に関係します。データシートに掲載されているスイッチング エネルギー曲線を使用すると、合計スイッチング損失を推定できます。

同様に、すべてのデバイスの導通損失およびスイッチング損失と、効率を推定できます。熱システム設計の熱インピーダンス情報に基づいて、適切なデバイス定格を選択できます。1200V/75mΩ の SiC MOSFET と 650V/60mΩ の SiC MOSFET は、放熱、効率、コスト間の良いトレードオフとなります。

2.3.1.5 損失の推定

インバータで効率が低下する主な原因は、スイッチング デバイスで発生する損失です。これらの損失は、各デバイスで次の 3 つのカテゴリに分類されます。

- 導通損失: デバイスがオンになっており、正常に導通しているとき
- スwitching 損失: デバイスの状態が切り替わる時
- ダイオード導通損失: 導通時の電圧降下と電流に関係

これらにはそれぞれ固有の式があり、デバイスのデータシートとすでに設定されている設計パラメータから決定できます。導通損失は、FET のオン時間、スイッチングされる電流、およびオン抵抗を使用して計算されます。

$$P_{\text{cond_loss}} = \frac{1}{T} \int_0^T V_{\text{ce}}(t) \times I_{\text{c}}(t) \times D_{\text{Q}}(t) dt \quad (4)$$

ここで、

- V_{ce} : 導通電圧降下
- I_{c} : 導通電流
- D_{Q} : デューティ サイクル
- T : 1 変調サイクル

スイッチング損失は、デバイスのスイッチング エネルギーと、選択したテストポイントにおけるスイッチング電圧によって決定されます。設計した外部ゲート抵抗の値を使用して、デバイスのデータシートからスイッチング エネルギーの値を決定します。必要な残りの値は、設計フェーズの初期段階で決定されています。

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

図 2-25 に、C3M0060065D SiC MOSFET について、デバイスのデータシートからスイッチング エネルギーの値を抽出するために使用されたグラフの例を示します。

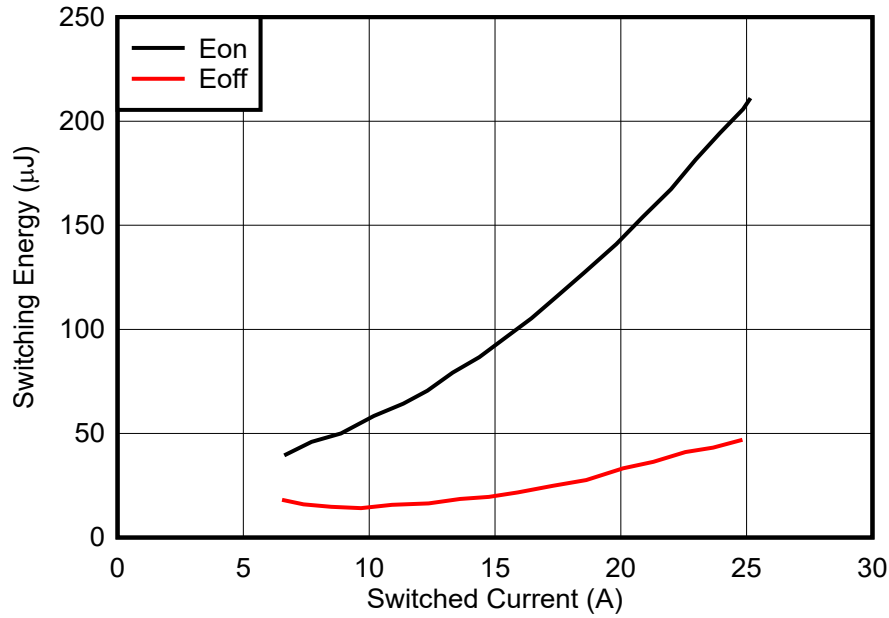


図 2-25. C3M0060065D のスイッチング エネルギーとスイッチ電流の関係

ダイオードの導通損失も同様に、既知の値を使用して計算します。

$$P_{sw_diode} = \frac{1}{T} \int_0^T V_f(t) \times I_f(t) \times D_D(t) dt \quad (6)$$

ここで、

- V_f : 電圧降下
- I_f : ダイオード電流
- D_D : デューティ サイクル
- T : 1 変調サイクル

これらの 3 つの式を使用して、両方の SiC MOSFET の設計で計算された推定される損失を表 2-1 に示します。

表 2-1. スwitchング デバイスで推定される損失

パラメータ	C3M0075120D (Q1)	C3M0060065D (Q3)
導通損失	5.76 W	4.5W
スイッチング損失	1.8 W	1.13W
ダイオード損失	0 W	0W
合計	7.56 W	5.63W

システム全体の損失を推定するための最後の要素は、インダクタの損失です。これらの損失は、インダクタの DC 抵抗と AC 抵抗の値、およびセクション 2.3.1.3 から予測されるインダクタ電流を使用して計算されます。

$$P_{ind_loss} = I_{ind_ac_rms}^2 \times R_{DC} + I_{ind_ripple_rms}^2 \times R_{AC} \quad (7)$$

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

そのため、この設計の主要エネルギー損失の合計は次のようになります。

$$P_{loss_total} = 6 \times (P_{Q1_total} + P_{Q3_total}) + 3 \times P_{int_loss} \quad (9)$$

$$P_{\text{loss_total}} = 6 \times (5.631 \text{ W} + 7.56 \text{ W}) + 3 \times 5.64 \text{ W} = 96.102 \text{ W} \quad (10)$$

式 10 を使用して、予想されるインバータの総効率を求めます。これは推定値ですが、推定値を使用すると、この時点までの設計を検証することができます。

$$\eta = \frac{P_{\text{out}}}{P_{\text{out}} + P_{\text{loss_total}}} \times 100 \quad (11)$$

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

2.3.2 電圧検出

電圧検出は、制御のため、インバータの信号パス内の 2 つのポイント (1 次側出力リレーの前後) で実行されます。リレーの両側で測定を有効にすることで、制御システムで接続前に送電網の電圧と周波数にロックし、不整合の問題を防止できます。

検出トポロジはどちらも類似しています。まず、抵抗ネットワークを使用して、V_PCC_N and V_SN_N を仮想中性点として使用します。高電圧信号は、一連の値の大きい抵抗を使用して減衰されます。減衰された位相電圧に 1.65V のオフセットを追加して、ADC の VREF (3.3V) に対して電圧を中心に配置します。

2.3.3 電流検出

閉ループ制御システムを実現するためには、インバータの電流を正確に測定することが重要です。この設計では、電流の測定は 2 つの場所で異なる検出技術を使用して行われます。1 つ目の場所は送電網出力で、ここではシャント抵抗を使用します。出力は高電圧であり、コントローラを絶縁状態に維持する必要があるため、AMC3306M05 強化絶縁型変調器を使用して抵抗での電圧降下を測定します。システム損失を低く抑えるため、AMC3306M05 の入力範囲は $\pm 50\text{mV}$ です。標準的な入力範囲 $\pm 250\text{mV}$ の他のデバイスと比較して、シャント抵抗の両端での合計電力損失は大幅に低減されます。

この設計でのシャント抵抗のサイズは、検出精度と消費電力とのトレードオフを考慮して決定します。0.002 Ω のシャント抵抗を使用すると、約 $\pm 25\text{A}$ の出力インバータで $\pm 50\text{mV}$ の出力信号が供給されると同時に、全負荷時の発熱はわずか 0.5W です。実際のデバイスを選択する際には、各センサパスを校正する必要性をなくすため、高精度の値を選択します。

シャント抵抗の両端の電圧が AMC3306M05 デルタシグマ変調器に供給され、デルタシグマストリームが生成されます。このストリームは、C2000™ MCU に搭載されている SDFM 復調器によってデコードされます。変調器のクロックは C2000 MCU 上の eCAP ペリフェラルで生成され、AMC3306M05 のデータは内蔵の SDFM 変調器を使用して決定されます。

2 つ目の場所はホール効果センサ TMCS1123 で、インダクタを流れる電流を検出するために使用されます。OPA4388 を使用して出力をフィルタリングし、すべての 3 相測定結果の中心を同じ 1.65V のオフセットリファレンス電圧に揃えます。

2.3.4 システムの補助電源

絶縁型バイアス電源、ファン、リレー、低電圧アナログ / デジタル回路、C2000 制御カードなどのサポート デバイスに電力を供給するには、12V、2A の補助電源が必要です。多数のオンボード電圧コンバータとレギュレータがあることで、12V 入力から、5V、3.3V、1.65V の各レールを供給できます。テスト中に保護機能を追加するため、過電圧および過電流保護機能を備えた 12V 電源の使用を推奨します。

2.3.5 ゲートドライバ

2.3.5.1 1200V の SiC MOSFET

VCC と GND は、UCC21710 デバイスの入力側の電源ピンです。VCC の電源電圧範囲は、GND に対して 3.0V ~ 5.5V です。VDD と COM は、UCC21710 デバイスの出力側の電源ピンです。VEE は出力ドライバの電源リターン、

COM はロジック回路の参照です。VDD の電源電圧範囲は、VEE に対して 15V~30V です。PWM は、ゲートドライバの IN+ ピンと IN- ピンの間に印加されます。

ゲートドライバの 2 次側では、ゲート抵抗 (R203 と R204 など) を使用してスイッチング デバイスのゲート電流を制御します。DESAT フォルト検出により、短絡フォルト時の過剰なコレクタ電流による破損が防止されます。スイッチング デバイスの損傷を防ぐため、フォルトが検出された場合、UCC21710 は SiC MOSFET をゆっくりとオフにします。ゆっくりオフにすることにより、フォルト状況が発生したときに過電流を制御された方法で低減できます。DESAT ダイオード (D10 など) はゲートドライバからバイアス電流を導通し、SiC MOSFET がオン状態のときに、MOSFET のコレクタとエミッタ間の飽和電圧を検出できます。

2.3.5.2 650V の SiC MOSFET

UCC5350 の 1 次側は、3.3V レールから電力を供給されます。ノイズ デカップリング用に 0.1 μ F のセラミック コンデンサを VCC1 ピンの近くに配置しています。電源の正方向 UVLO スレッショルドは 2.6V、負方向のスレッショルドは 2.5V です。

ゲートドライバに PWM 入力を供給するのは、コントローラの PWM 出力ペリフェラルです。ローサイドの PWM 信号とハイサイドの PWM 信号の間にデッド タイムを挿入し、両方のスイッチが同時にオンになる事態を防ぐ必要があります。この信号はシングルエンドで、RC ローパス フィルタでフィルタ処理した後、ゲートドライバの入力に接続します。このフィルタは高周波ノイズを減衰し、コントローラからゲートドライバまでの比較的長い配線に起因する PWM 入力のオーバーシュートやアンダーシュートを防止することができます。このリファレンス デザインは、反転型の PWM 入力 IN- を使用せず、1 次側グラウンドに接続しています。

MOSFET のターンオンおよびターンオフ制御には、3.3 Ω のゲート抵抗 (例: R258) を使用します。10k Ω の抵抗 (例: R85) を、MOSFET のゲートと、メイン電源基板の MOSFET に近接したコレクタ ピンとの間に接続します。この結果、異常が発生してゲートドライバが MOSFET から接続解除された場合に、MOSFET がオフ状態のままになります。

2.3.5.3 ゲートドライバのバイアス電源

高電圧バリアを通過するゲートを駆動するために、ゲートドライバは UCC14341B 絶縁バイアス電圧 (+15V および -4V) を使用しています。このアーキテクチャは合計 9 個のゲートドライバを使用し、絶縁型バイアス電源のデフォルトの数は 9 個です。絶縁型バイアス電源の数を 5 個に減らすことができます。9 電源オプションの場合:

1. REF_VDC-n を基準とする 3 つのローサイド電源 (n = 1, 2, 3)
2. REF_MIDn を基準とする 3 つのミドルサイド電源 (n = 1, 2, 3)
3. REF_SN_Ln を基準とする 3 つのスイッチング ノード電源 (n = 1, 2, 3)

5 電源オプションの場合:

1. REF_VDC-n を基準とする 1 つのローサイド電源 (n = 1)
2. REF_MIDn を基準とする 1 つのミドルサイド電源 (n = 1)
3. REF_SN_Ln を基準とする 3 つのスイッチング ノード電源 (n = 1, 2, 3)

5 電源オプションが動作するためには、3 相のすべてにわたってリファレンス ポイント間の電圧降下が無視できるほど小さいものと仮定します。

2.3.6 制御設計

用語:

V_{bus} または V_{dc} インバータのバス電圧

L_i および R_i インバータ側インダクタのインダクタンスおよび直列抵抗

L_g および R_g 送電網側インダクタのインダクタンスおよび直列抵抗

C_f および R_f 容量値および直列抵抗とダンピング抵抗

v_{i_a} , v_{i_b} , v_{i_c} 3相ブリッジの出力電圧。この電圧は、3相ブリッジのデューティ・サイクル制御を使用して制御されます。

そのため、制御目的では、2レベルと3レベルのインバータの間で変調のみを変更する必要があると想定されます (電力段と制御設計は変更しない)。 D_A , D_B , D_C は、インバータの出力電圧が式 13 で表されるように生成された制御変数です。

$$v_{i_a} = D_a \times \frac{V_{DC}}{2} \quad (13)$$

i_{i_a} , i_{i_b} , i_{i_c} インバータ側インダクタを流れる電流

v_{x_a} , v_{x_b} , v_{x_c} フィルタ・コンデンサの両端の電圧

i_{g_a} , i_{g_b} , i_{g_c} フィルタ・コンデンサの両端の電圧

v_{g_a} , v_{g_b} , v_{g_c} 送電網側の周波数

2.3.6.1 電流ループの設計

図 2-26 に示すインバータフィルタでは、KCL と KVL を使用して、式 14 を記述できます。

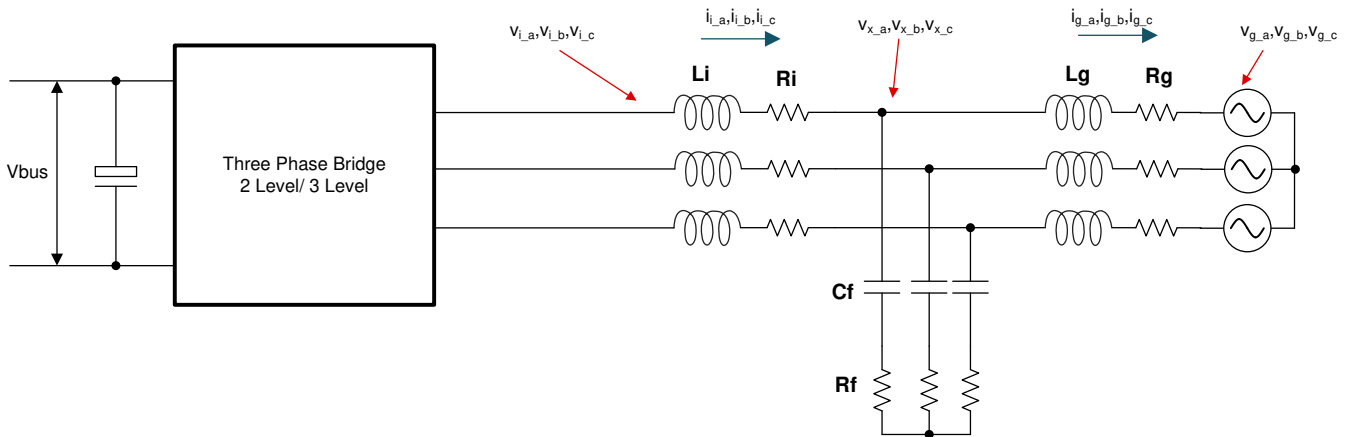


図 2-26. インバータ モデル

$$v_{i_a} - L_i \frac{di_{i_a}}{dt} - R_i i_{i_a} = v_{x_a} \quad (14)$$

式 14 は式 15 のようにも記述できます。

$$\frac{di_{i_a}}{dt} = \frac{1}{L_i} v_{i_a} - \frac{1}{L_i} (R_i i_{i_a} + v_{x_a}) \quad (15)$$

別のノードでも同様に、KCL と KVL を使用して、式 16 を式 16 のように記述できます。

$$\frac{di_{g_a}}{dt} = \frac{1}{L_2} v_{x_a} - \frac{1}{L_2} (R_2 i_{g_a} + v_{g_a}) \quad (16)$$

R_f は無視できると仮定すると、式 17 はコンデンサの電圧用に次のように記述できます。

$$\frac{dv_{x_a}}{dt} = \frac{1}{C_f} (i_{i_a} - i_{g_a}) \quad (17)$$

通常は、同期基準フレーム制御が設計されます。この設計では、 d_q は送電網側の周波数速度で回転する基準フレームで、 d 軸が送電網の電圧ベクトルに揃うように配置して使用します。基本的な三角関数の公式を使用すると、 i_d および i_q は式 18 および式 19 のように記述できます。

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

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

微分を取って偏微分定理を使用すると、式 20 は次のようになります。

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

状態式は次のようになります。

$$\frac{di_{i_d}}{dt} = \frac{1}{L_i} v_{i_d} + \omega i_{i_q} - \frac{1}{L_i} (R_i i_{i_d} + v_{x_d}) \quad (21)$$

$$\frac{di_{i_q}}{dt} = \frac{1}{L_i} v_{i_q} - \omega i_{i_d} - \frac{1}{L_i} (R_i i_{i_q} + v_{x_q}) \quad (22)$$

これらの式を使用すると、式 23 のようになります。

$$(sL_i) i_{i_d}(s) = v_{i_d}(s) + (\omega L_i) i_{i_q}(s) - (R_i i_{i_d}(s) + v_{x_d}(s)) \quad (23)$$

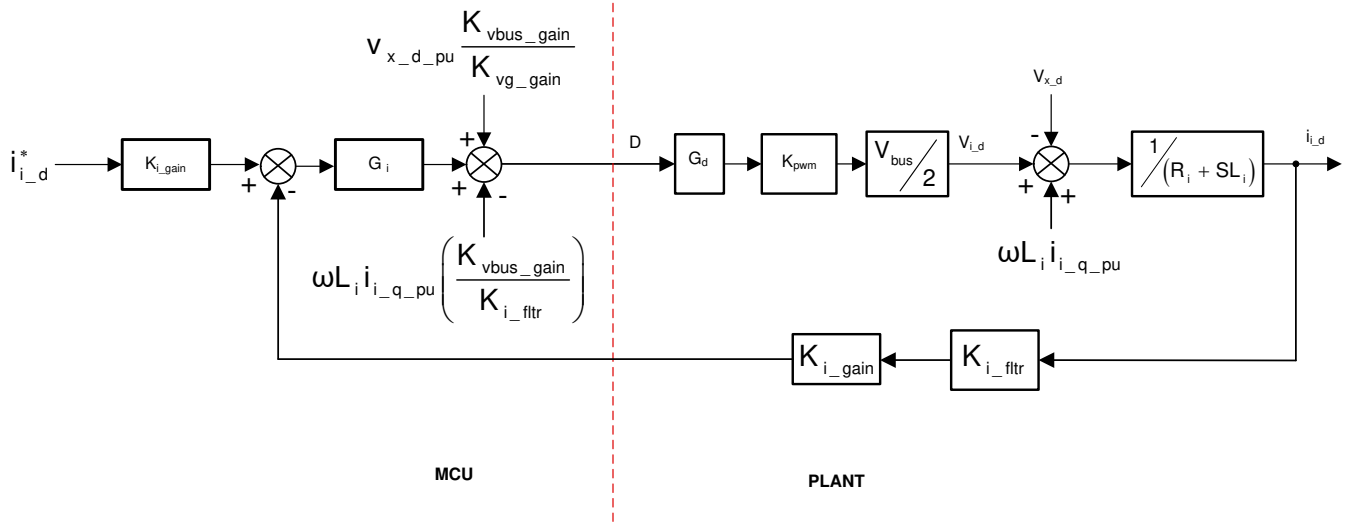
上記の式にラプラス関数を適用すると、次のようになります。

$$(sL_i) i_{i_q}(s) = v_{i_q}(s) - (\omega L_i) i_{i_d}(s) - (R_i i_{i_q}(s) + v_{x_q}(s)) \quad (24)$$

制御ダイアグラム形式で記述すると、次のようになります。フィードフォワード要素を追加して、モデルに存在する追加の外乱およびエラーの原因を除去します。次の 2 つのフィードフォワード要素が追加されます。

1. 同期フレーム内の他の軸からの結合項用
2. 出力送電網の電圧用

図 2-27 に示すような図になります。



ここで

- $i_{i_d}^*$: 電流基準
- K_{i_gain} : 電流検出スカラー値 (最大電流検出の逆数)
- K_{i_fltr} : 電流検出パスに接続されているフィルタ。電流検出スカラー値 (最大電流検出の逆数)
- K_{vbus_gain} : バスの電圧検出スカラー値 (検出された最大電圧の逆数)
- K_{vg_gain} : 送電網の電圧の電圧検出スカラー値 (検出された最大電圧の逆数)

図 2-27. Id 電流ループモデル

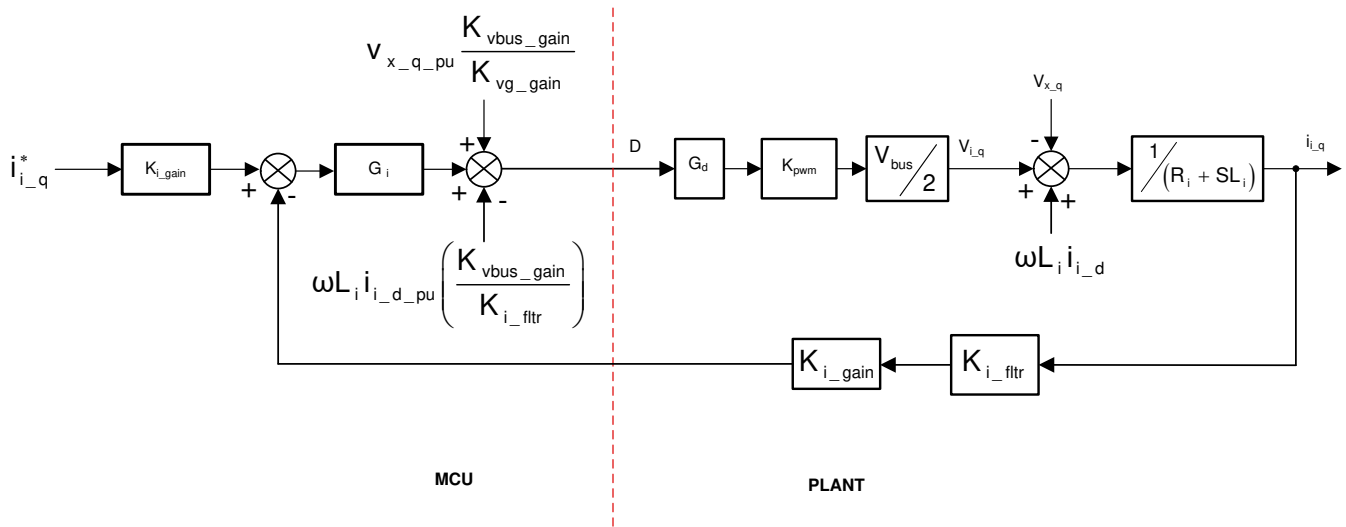


図 2-28. Iq 電流ループモデル

フィードフォワード要素を使用する場合、小信号モデルは式 25 のように記述できます (注: バス電圧と送電網の電圧の検出範囲は異なるので、別のスケーリング係数が適用されます)。

$$\frac{\hat{i}_{i_d_pu}}{\hat{d}} = G_d K_{pwm} \frac{1}{K_{vbus_gain}} K_{i_gain} K_{i_fltr} \frac{1}{(R_i + sL_i)} \quad (25)$$

LCL フィルタの場合、式 26 に示すような簡略化モデルを想定できます。

$$\frac{\hat{i}_{i_d_pu}}{\hat{d}} = G_d K_{pwm} \frac{1}{K_{vbus_gain}} K_{i_gain} K_{i_fltr} \frac{1}{\left(Z_i + \frac{Z_c Z_f}{Z_c + Z_f} \right)} \quad (26)$$

電流ループプラントは、[図 2-29](#) に示すように、電流ループのソフトウェア周波数応答アラート (SFRA) 測定データと比較されます。

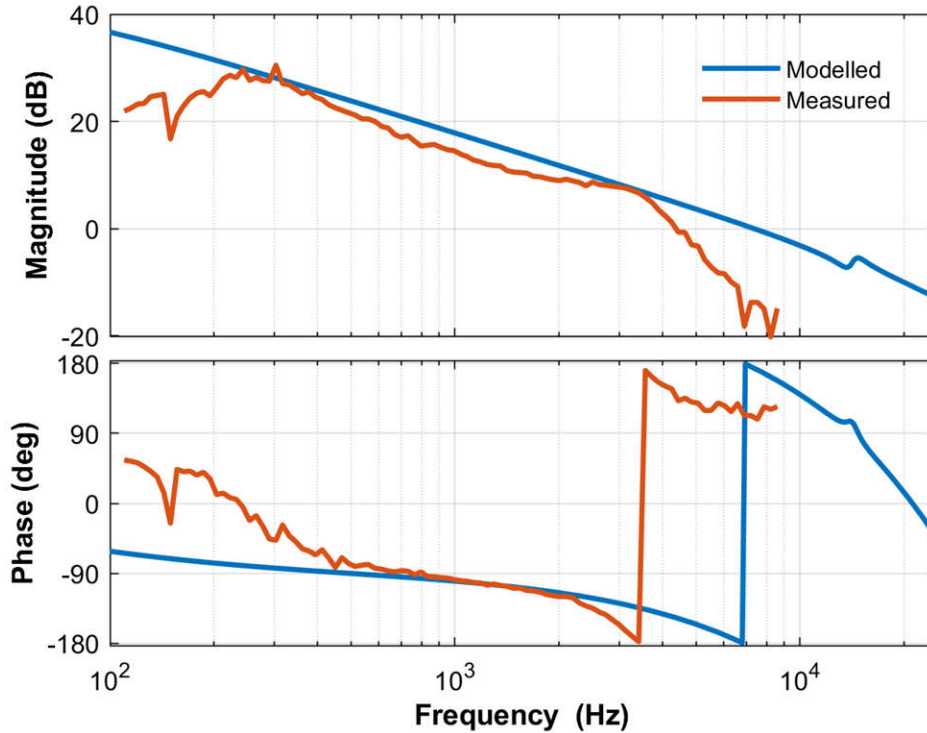


図 2-29. 電流ループプラントの周波数応答のモデルと測定値の比較

[式 27](#) は、閉ループ動作に設計された補償回路を表します。

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

[図 2-30](#) の開ループプロットを達成すると、 I_d および I_q ループで約 1kHz を超える帯域幅が得られます。

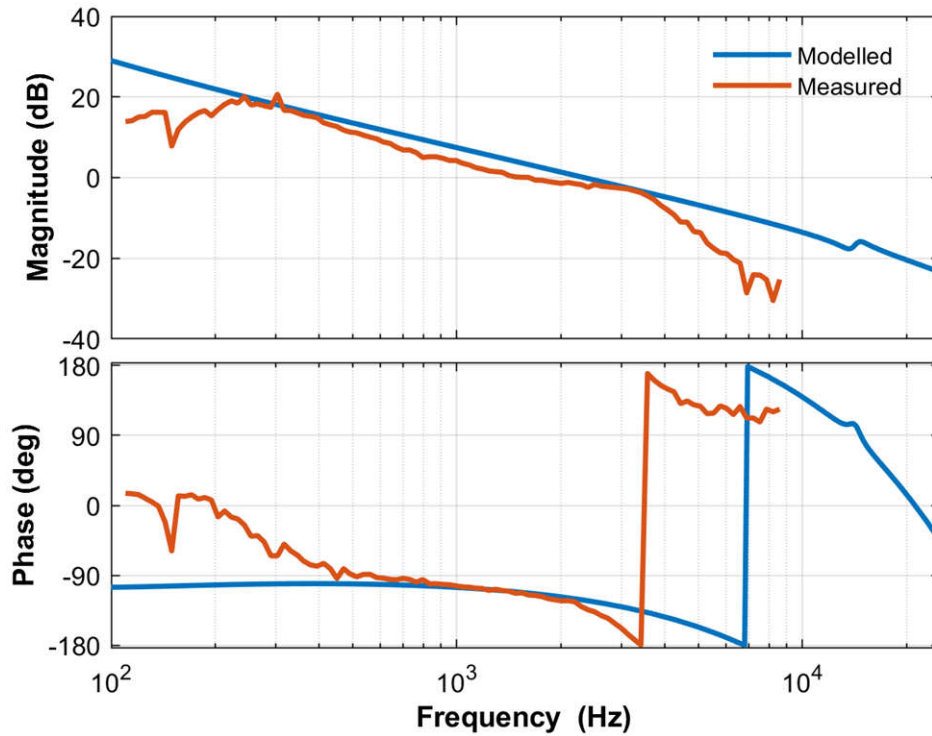


図 2-30. 電流ループ、開ループ応答のモデルと測定値の比較

2.3.6.2 PFC の DC バス電圧レギュレーション ループの設計

電圧ループ モデルをしてみる前に、DQ ドメインからの電力測定値は次のように記述できます。

$$P = v_a i_a + v_b i_b + v_c i_c = \begin{bmatrix} v_a & v_b & v_c \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \left(T_{abc \rightarrow dq0}^{-1} \vec{v}_{dq0} \right)^T \left(T_{dq0 \rightarrow abc}^{-1} \vec{i}_{dq0} \right) = \vec{v}_{dq0} \left(T_{abc \rightarrow dq0}^{-1} \right)^T T_{abc \rightarrow dq0}^{-1} \vec{i}_{dq0} = \frac{3}{2} (v_{gd} i_d + v_{gq} i_q) \quad (28)$$

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

ここで、

したがって、

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

$$Q = -v_{gd} i_d + v_{gq} i_q \quad (30)$$

DC バス電圧レギュレーション ループは、基準電力を供給すると想定されます。基準電力をライン電圧 RMS の 2 乗で除算すると、コンダクタンスが得られます。これにライン電圧を掛けると、瞬間電流が得られます。

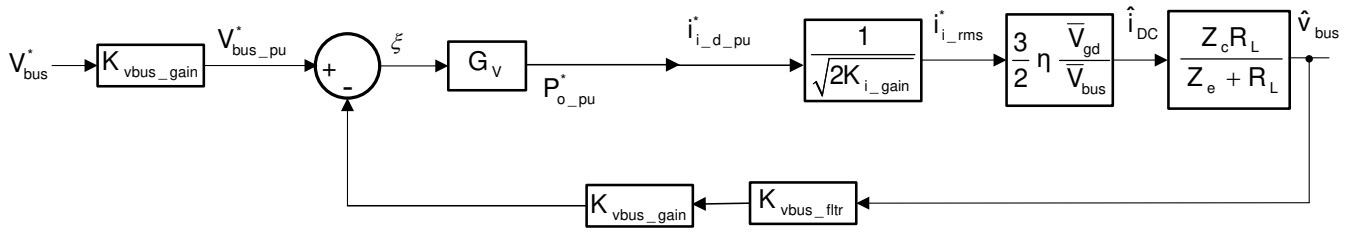


図 2-31. 電圧ループモデル

DC バス電圧レギュレーションループの小信号モデルは、式 31 を動作点近傍で線形化して作成します。

$$i_{DC} V_{bus} = 3\eta v_{g_rms} i_{g_rms} \Rightarrow \hat{i}_{DC} = 3\eta \frac{\bar{V}_{g_rms}}{\bar{V}_{bus}} \hat{i}_{g_rms} \quad (31)$$

変換は振幅不変なので、 $\hat{i}_{g_rms} = \frac{1}{\sqrt{2}} \hat{i}_{gd}$ および $\hat{v}_{g_rms} = \frac{1}{\sqrt{2}} \hat{v}_{gd}$ を使用して RMS からピーク量に変換することで、式 32 を求めることができます。

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

DC バスの抵抗性負荷は次のようになります。
$$\hat{v}_{bus} = \frac{Z_c R_L}{Z_c + R_L} \hat{i}_{DC}$$

したがって、電圧ループプラントは式 33 のように記述できます。

$$G_{v_bus} = \frac{Z_c R_L}{Z_c + R_L} \times p \times \frac{K_{vbus_fitr} \times K_{vbus_gain}}{K_{i_gain}} \times \frac{1}{(V_{bus} \times K_{vg_gain})} \times \frac{3}{2} \times \left(\frac{V_{g_peak}}{V_{g_Sense_max}} \right) \quad (33)$$

以前のモデルを使用すると、電圧ループ用に式 34 で表される補償器が設計されます。

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

SFRA を使用して電圧ループ帯域幅を測定し、モデルと比較すると、モデルと良好な相関が示されます。図 2-32 にプラントの周波数応答のモデルと測定値の比較を示し、図 2-33 に開ループ周波数応答のモデルと測定値の比較を示します。

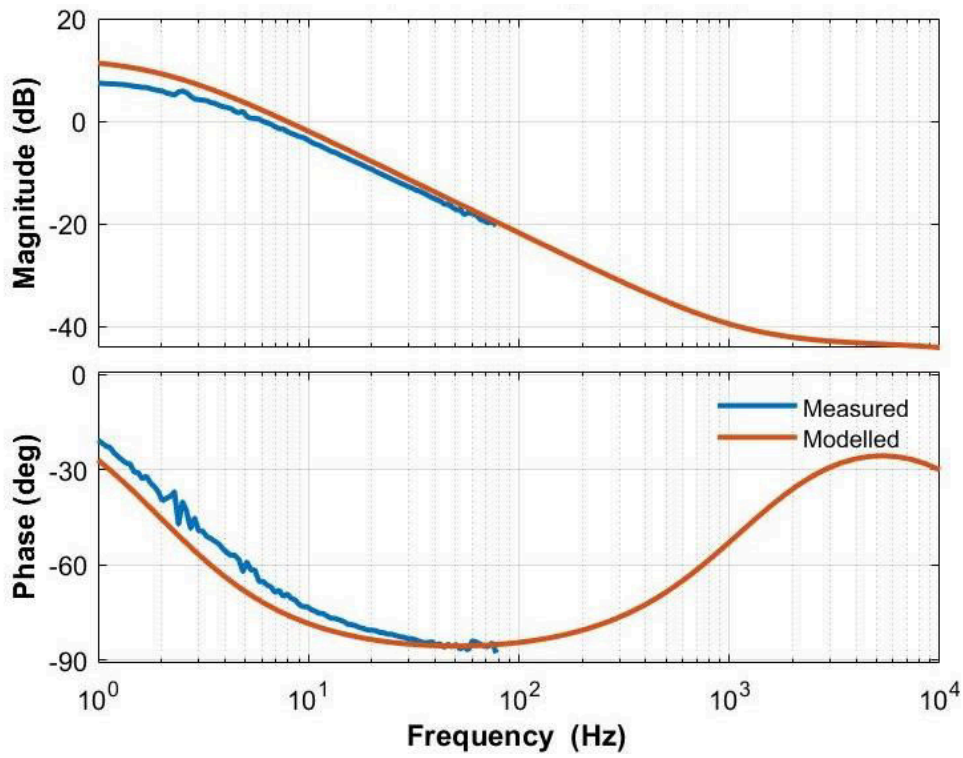


図 2-32. 電圧ループ プラントの周波数応答の測定値とモデルの比較

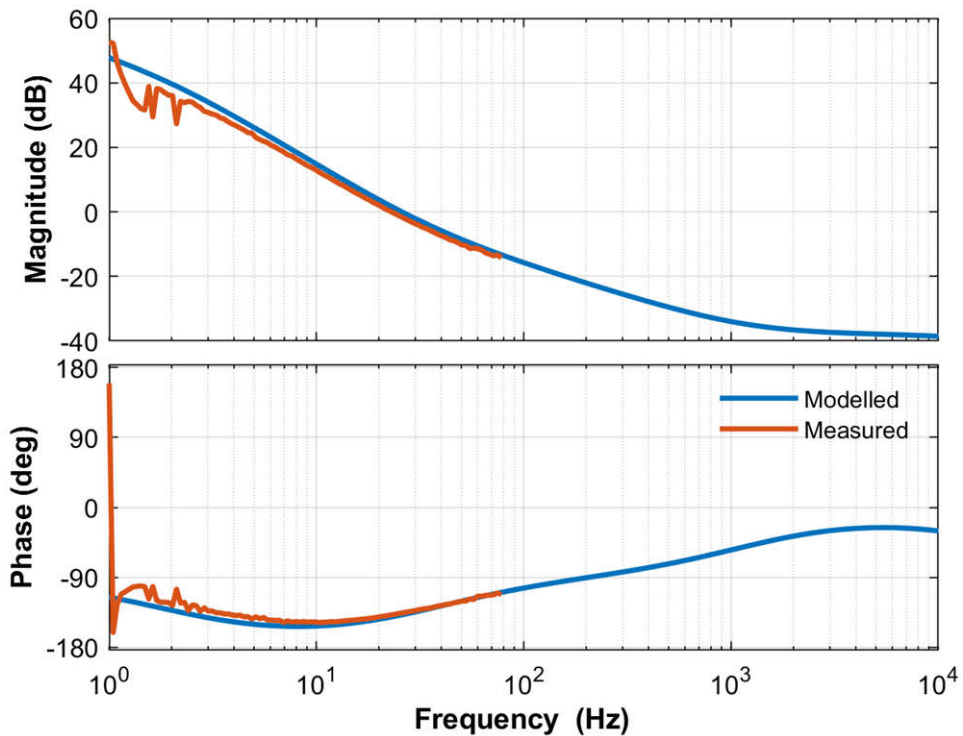


図 2-33. 電圧ループと開ループの周波数応答のモデルと測定値の比較

3 ハードウェア、ソフトウェア、テスト要件、テスト結果

3.1 必要なハードウェアとソフトウェア

このセクションでは、ハードウェアの詳細について解説し、ボード上のさまざまなセクションと、この設計ガイドで説明しているように、実験用にそれらを設定する方法について説明します。

3.1.1 ハードウェア

3.1.1.1 必要なテスト ハードウェア

この設計の Device Under Test (DUT) は、以下に示す複数のコンポーネントで設定され、動作します。

- E7 TIDA-01606 マザー ボード 1 枚
- [TMS320F28379D](#) 制御カードまたは [TMS320F280039C](#) 制御カード
- スナップ フェライト付き Mini USB ケーブル (ZCAT3035-1330-BK など、ノイズ耐性のために推奨)
- USB アイソレータ 1 個 (安全性とノイズ耐性のため)
- ラップトップまたはその他のコンピュータ (Microsoft® Windows® 11 オペレーティング システムなど)

この設計への電力供給と評価に必要な試験装置は、以下のとおりです。

1. 1 次側ボード電源用の 12V、2A ベンチ型電源。TI は、過電圧保護 (OVP) 機能と過電流保護 (OCP) 機能を備えた電源を推奨します。
2. PFC モードの場合:
 - 400V_{L-L} 対応の 3 相 AC 電源
 - 11kW 等価電子負荷を DC 出力に接続
3. インバータ モードの場合:
 - スター型接続の抵抗性負荷ネットワーク (11kW の負荷バンクなど)
 - DC リンク入力用の 800V、12A 電源
4. 4 チャネル、電源品質アナライザ (WT5000 など)
5. オシロスコープ、電圧計、差動プローブ (P5200A など)、電流プローブ (TCP0030 など)

この設計は、高速エッジカード (HSEC) の概念を採用しています。この設計は、互換性のある HSEC 制御を使用して、C2000™ マイコン製品ファミリの複数のデバイスに拡張できます。マイコンの電力段の制御に使用される主要なリソースを表 3-1 に示します。

3.1.1.2 設計に使用されているマイクロコントローラ リソース (TMS320F28379D)

表 3-1 にボード上の電力段の制御に使用される主要なコントローラ パリフェラルの詳細を示し、表 3-2 に主要なコネクタと機能の一覧を示します。

表 3-1. 基板の電力段の制御に使用される主要なコントローラ パリフェラル

ピン番号	説明	ソフトウェア名
15, 31, 28	送電網の電圧検出相 A、B、C	TINV_VGRID_A、B、C
21, 33, 30	インバータ側の電圧相 A、B、C	TINV_VINV_A、B、C
25, 37, 34	インバータ側の電流相 A	TINV_IINV_A、B、C
42	バス電圧検出	TINV_VBUS
40	バス電圧の中間点検出	TINV_VBUS_MID
12, 14, 18, 20	温度 A、B、C、周囲	TINV_TEMP_A、B、C、AMB
49, 50, 58	PWM1 相 A、B、C	TINV_Q1_A、B、C
51, 52, 60	PWM3 相 A、B、C	TINV_Q3_A、B、C
53, 54, 62	PWM2 相 A、B、C	TINV_Q2_A、B、C
99, 103, 107	SDFM データ IG A、B、C	TINV_IGRID_A、B、C
101, 105, 109	SDFM クロック IG A、B、C	
57, 75	SDFM クロック ソース	
89, 87, 85	SiC フォルト信号 A、B、C (アクティブ Low)	TINV_FAULT_A、B、C

表 3-1. 基板の電力段の制御に使用される主要なコントローラ パリフェラル (続き)

ピン番号	説明	ソフトウェア名
92	制御リレー A、B、C。 E7 は、1 つの GPIO を使用してすべてのリレーを制御します。	TINV_RELAY_N
61、63	ゲートドライバ電源 PWM	TINV_GATE_DRIVE
71	ファンの制御 GPIO。従来、E6 では、ファンの制御 GPIO はピン 59 でした。	TINV_FAN
108、110	ファームウェアのデバッグ開始時にドッキングステーションでの ISR のネスティングなどを確認するために使用	TINV_PROFILING1、2 年
95	ゲートドライバのイネーブル	TINV_PWM_EN
81	ゲートドライバのリセット	TINV_R

注

C2000Ware_DigitalPower_SDK で E7 ハードウェアをサポートするため、ソフトウェアに小規模な変更が加えられました。新しいファン制御 GPIO をサポートするために、`tinv_user_settings.h` のコードを必ず変更してください。

```

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

#define TINV_FAN_GPIO          18
#define TINV_FAN_GPIO_PIN_CONFIG GPIO_18_GPIO18

```

CCS WATCH ウィンドウでリレー ユーザー制御を行うには、`TINV_neutralRelaySet` 関数を使用してリレーを制御します (E7 ハードウェアの変更)。また、絶縁バイアス電源イネーブルピンはアクティブ Low です。ユーザーが CCS デバッグを起動すると、デフォルトで電源が有効になります。`TINV_allRelaySet` 関数はバイアス電源のイネーブルピンに影響を及ぼすため、使用しないでください。FET が過熱する可能性があるため、高電力をプッシュする前に、`TINV_fanSet` 関数でファンを有効にしてください。

表 3-2. 主要なコネクタと機能

コネクタ名	機能
J13、J15、J18	VDC+、VDC_MID、VDC 端子
J30、J14、J16、J17	PE (保護アース)、L1、L2、L3 端子
J3	12V 補助電源
J1	補助電源用ジャンパ
J26、J29B	HSEC 制御カードのコネクタ スロット
J4、J5、J6	MCU GND リファレンス オプション。一度に 1 つのオプションを選択してください。デフォルトは J6 (GRID_GND / PE) です
J19、J20、J21	ファン コネクタ
S1–S5	5 つのすべてのスイッチを、379D の場合はポジション 1、039C の場合はポジション 3 のいずれかに設定します。PCB には、ユーザーが設定を構成するのに役立つシルクスクリーン ラベルがあります。

注

デフォルトの MCU GND リファレンスは GLID_GND / PE です。R10 の代わりにフェライトを使うと、同相ノイズ除去に役立ちます。マイコンのリファレンスとして J4 と J5 のどちらかを選択する場合は、高電圧入力ソース (例: 3-P Chroma) を使用してテストする際に、絶縁型 12V ベンチ電源を使用してください。スイッチング ノイズが伝導エミッションと放射エミッションによってファン機能を損なう場合があります。スナップ フェライト チョークでファンのノイズの問題を緩和できない場合、大電力テスト中もファンの正常な動作を維持するために、別の 12V 電源を使用してください。つまり、J19、J20、J21 からファンを取り外し、別の 12V (1A) 電源に接続してください。

大電力テスト時には、ほとんどのシングルエンド プローブのリターン クリップがアース GND に接続されているため、差動プローブを使用してマイコンの低電圧側でも信号を測定することを推奨します。これにより、同相モードのノイズに対する低インピーダンスの経路を形成し、オシロスコープに表示できます。差動プローブを使用すると、GND がアースされるのに十分なインピーダンスが得られ、ハイサイドからローサイドに同相モードのノイズがデカップリングします。これは、R10 の代わりにフェライトを使用している場合に適切に機能します。

3.1.1.3 F28377D、F28379D 制御カードの設定

デバイス制御カードの設定には、JTAG 経由の通信が必要であり、絶縁 UART ポートを使用します。また、適切な ADC 基準電圧も提供する必要があります。以下に、F28379D 制御カードのリビジョン 1.1 に必要な設定を示します。

C2000Ware の <sdk_install_path>c2000ware\boards\controlCARDs\TMDSCNCD28379D にある情報シートも参照してください。

1. デバイスへの JTAG 接続と SFRA GUI 用の UART 接続を確立するには、制御カードの A:SW1 を両端で ON (上) に設定する必要があります。このスイッチが OFF (下) になっている場合、制御カードに内蔵されている絶縁 JTAG を使用できず、SFRA GUI はデバイスと通信できません。
2. A:J1 は、USB ケーブル用のコネクタで、Code Composer Studio (CCS) が実行されているホスト PC からデバイスに通信するために使用されます。前述したように、USB アイソレータを使用してラップトップに接続します。
3. この設計の制御ループの調整には 3.3V の基準電圧が適切です。そのため、オンチップ ADC に 3.3V の基準電圧を外部から供給するように適切なジャンパを設定します。F28379D 制御カードのバージョン 1.3 では、SW3 と SW2 を「」側の端 (定義通り左側) に移動します。これにより、3.3V VDDA が ADC の基準電圧となります。詳細については、「情報シート」を参照してください。
4. ジャンパコネクタ J3 で、ジャンパを使用してピン 2 とピン 3 を接続します。

3.1.1.4 設計に使用されているマイクロコントローラ リソース (TMS320F280039C)

注

TMDSCNCD280039C の評価基板と開発基板は将来的にサポートされています。対象の制御カードに従って、メイン マザー ボードで S1 ~ S5 を構成します。

3.1.2 ソフトウェア

関連するソフトウェア情報については、www.ti.com/tool/ja-jp/C2000WARE-DIGITALPOWER-SDK を参照してください。

3.1.2.1 ファームウェアの概要

3.1.2.1.1 CCS プロジェクトを開く

このリファレンス デザインのソフトウェアは C2000Ware_DigitalPower_SDK で利用可能であり、powerSUITE フレームワーク内でサポートされています。プロジェクトを開くには、次の手順に従います。

1. CCS (バージョン 10.1 以降) をインストールします。
2. ツール ページから C2000Ware DigitalPower SDK をインストールします
3. CCS を開き、新しいワークスペースを作成します。

4. CCS 内で、[View] → [Resource Explorer] を選択します。Resource Explorer で、[Software] → [C2000Ware DigitalPower SDK - <version>] → [solutions] の順に選択し、ソリューション **TIDA-01606** を選択して [import project] をクリックします。このコードは、F28379D と F280039C の両方で利用できます。

注

CCS は、インポートされたプロジェクトに関連する特定バージョンのコンパイラをインストールすることを推奨します。必要に応じて、TI.com でコンパイラを見つけて、ダウンロードとインストールを行ってください。コンパイラのインストール後、プロジェクトのプロパティメニューでコンパイラのバージョンを構成します。CCS プロジェクトツールの検出パスに、コンパイラのインストールのパスが含まれていることを確認してください。

3.1.2.1.2 デジタル電源 SDK のソフトウェアアーキテクチャ

プロジェクトの一般構造を <> に示します。プロジェクトがインポートされると、CCS 内に Project Explorer が表示されます。

コア アルゴリズム コードで構成されるソリューション固有でデバイスに依存しないファイルは、<solution>.c および <solution>.h にあります。

基板固有およびデバイス固有のファイルは、<solution>_hal.c および <solution>_hal.h にあります。このファイルは、シナリオを実行するデバイス固有のドライバで構成されています。別の変調方式やデバイスを使用する場合、プロジェクト内のデバイス サポート ファイルを変更する以外に変更を加える必要があるのは、これらのファイルのみです。

<solution>-main.c ファイルは、プロジェクトのメイン フレームワークで構成されています。このファイルは、システム フレームワークの作成に役立つボード ファイルとソリューション ファイルの呼び出し、割り込みサービス ルーチン (ISR)、低速なバックグラウンド タスクで構成されています。

この設計では、<solution> は *tin* で、これはモジュール名とも呼ばれます。

powerSUITE ページは、Project Explorer に表示される main.syscfg ファイルをクリックして開くことができます。

powerSUITE ページでは <solution>_settings.h ファイルが生成されます。このファイルは、powerSUITE ページで生成されたプロジェクトのコンパイル時に使用される唯一の C ベース ファイルです。プロジェクトが保存されるたびに powerSUITE によって変更内容が上書きされるため、このファイルを手動で変更しないでください。

<solution>_user_settings.h は <solution>_settings.h に含まれており、ADC マッピングの #defines や GPIO など、powerSUITE ツールの範囲外の設定を保持するために使用できます。

Kit.json ファイルと solution.js ファイルも powerSUITE 内で使用されるため、ユーザーが変更することはできません。これらのファイルを変更すると、プロジェクトが正常に機能しなくなります。

設計名は、設計で使用されるすべての変数と定義のモジュール名としても使用されます。

したがって、すべての変数および関数呼び出しの前に *TINV* という名称が追加されます (TINV_vSecSensed_pu など)。この命名規則により、名前の競合を回避しながら、異なるシナリオを組み合わせることができます。

3.1.2.1.3 割り込みとラボの構造

このプロジェクトは、2 つの ISR (ISR1 および ISR2) で構成されています。ISR1 のほうが高速で、ネスティング不可能です。ISR1 は、制御ループおよび PWM 更新用に予約されています。ISR1 は、PRIM_LEG1_PWM_BASE → EPWM_INT_TBCTR_U_CMPC イベントでトリガされます。

ISR2 は、CPU タイマのオーバーフローにより開始する CPU タイマ INT でトリガされます。電流信号と電圧信号の移動平均を実行してノイズを除去したり、指定された基準のスルーレート関数を実行したりするなどのハウスキーピング機能を実行するために使用されます。

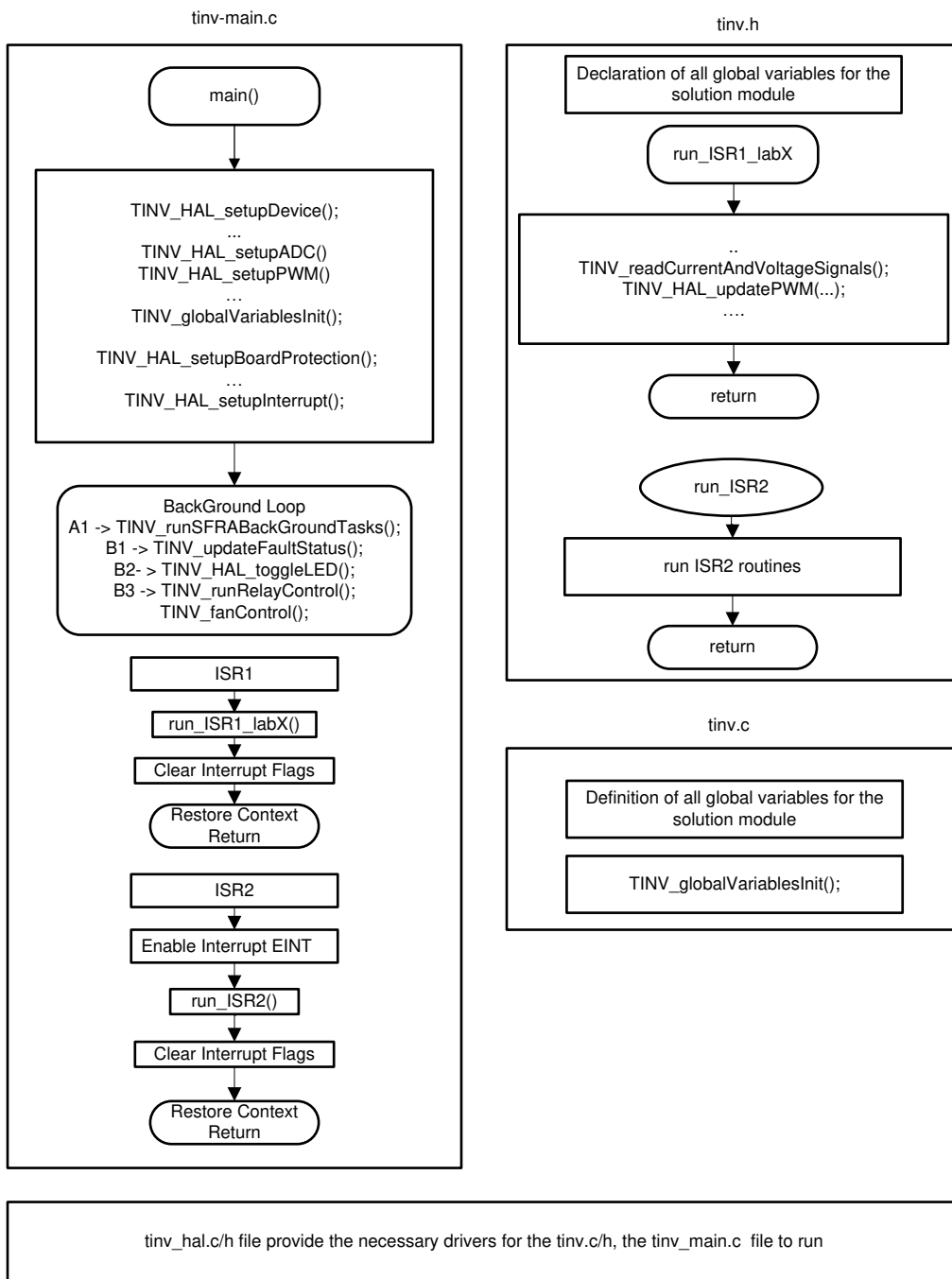


図 3-1. ソフトウェア フロー図

このリファレンス デザインのソフトウェアは 7 つのラボで構成されています。表 3-3 に、ラボとテスト方法を示します。すべてのラボは、C28x メイン CPU または制御補償器アクセラレータで実行できます。

表 3-3. リファレンス デザインをテストするラボの概要

ラボ番号	説明	備考	テスト環境
1	INV: PWM と ADC のチェック	PWM ドライバ、ISR 構造、実行レートをテスト、制御カードで実行可能。ユニット テストの保護メカニズム。ADC マッピングと変換データの読み取りをテスト。	制御カード
2	INV: 開ループのチェック	PWM チェック、ADC チェック、保護チェック、インバータ モード DC バス接続、負荷としての抵抗性スター型ネットワーク	制御カード + 電力段ハードウェア
3	INV: 閉電流ループ、AC に接続された抵抗性負荷		制御カード + 電力段ハードウェア
4	INV: 閉電流ループ、送電網に接続されたテストインバータ モード		制御カード + ハードウェア インザループでエミュレートされた電力段
5	PFC: 3 相 AC ソース、DC での抵抗負荷、開ループ チェック	1. vGridRms、iGridRms、vBus の測定値が正しいかどうかをチェック 2. PLL がロックされているかどうかをチェック。	制御カード + 電力段ハードウェア
6	PFC: 閉電流ループ、DC に接続された抵抗性負荷、3 相 AC ?		制御カード + 電力段ハードウェア
7	PFC: 閉電圧ループ + 電流ループ、DC に接続された抵抗性負荷、3 相 AC ?		制御カード + 電力段ハードウェア

3.1.2.1.4 ファームウェアのビルド、ロード、デバッグ

プロジェクトをビルドするには、プロジェクト名を右クリックし、[Rebuild Project] をクリックします。プロジェクトが正常にビルドされます。

プロジェクトをロードするには、まず Project Explorer の targetConfigs (*.ccxml ファイル) の下で正しいターゲット構成ファイルがアクティブに設定されていることを確認します。その後、[Run] → [Debug] をクリックしてデバッグ セッションを開始します。デュアル CPU デバイスの場合、デバッグを実行する CPU を選択するウィンドウが表示されます。ここでは、CPU1 を選択します。するとプロジェクトがデバイスにロードされ、CCS デバッグ ビューが有効になります。メイン ルーチンの開始時にコードは停止します。

システムをデバッグするには、[Watch]/[Expressions] ウィンドウで変数を監視します。このウィンドウに正しい変数を入力するには、[View] → [Scripting Console] をクリックして、[Scripting Console] ダイアログボックスを開きます。このコンソールの右上隅で [Open] をクリックし、プロジェクト フォルダ内にある setupdebugenv_lab<Number>.js スクリプト ファイルを参照します。これにより、[Watch] ウィンドウに、システムをデバッグするのに必要な適切な変数が入力されます。[Watch] ウィンドウで [Continuous Refresh] ボタンをオンにして、コントローラからの値が連続的に更新されるようにします。

リアルタイム エミュレーションは、MCU 動作中に Code Composer Studio 内のウィンドウを更新できる特別なエミュレーション機能です。この機能により、プロセッサを停止することなく、グラフおよび [Watch] ビューが更新されるようになるだけでなく、ユーザーが [Watch] ウィンドウや [Memory] ウィンドウの値を変更して、その変更をシステムに反映できるようになります。リアルタイム モードを有効にするには、CCS のトップ バーにあるこのボタンをクリックします。メッセージ ボックスが表示されたら、[YES] を選択して、デバッグ イベントを有効にします。これにより、ステータス レジスタ 1 (ST1) のビット 1 (DGBM ビット) が 0 に設定されます。DGBM は、デバッグ イネーブル マスク ビットです。DGBM ビットが「0」に設定されると、メモリ値とレジスタ値がホスト プロセッサに渡されて、デバッグのウィンドウが更新できるようになります。

ラボによっては、グラフ ウィンドウにデータを表示して、測定された電流と電圧または制御変数を検証する必要がある場合があります。このグラフ ウィンドウは、コントローラ上で実行されるコードと組み合わせて使用して、コントローラによって値がどのように検出されるかを示すスナップショットを表示できます。これらの値は通常、低速 ISR でデータロガーにより記録さ

れます。グラフを CCS ビューにインポートするには、[Tools] → [Graph] → [DualTime] を選択し、[Import] をクリックして、プロジェクトフォルダ内にある graph1.GraphProp ファイルを指定します。CCS に 2 つのグラフが表示されます。これらのグラフで [Continuous Refresh] をクリックします。graph2.GraphProp ファイルをインポートして、グラフの 2 つ目のセットを追加することもできます。

3.1.2.1.5 CPU ローディング

ラボ 3 とラボ 7 のメイン制御 ISR は、CPU1 から動作する場合、50kHz のレートで約 54MIPS になります。これは、200MHz の F2837x プロセッサから動作する場合、CPU の約 27% に相当します。これには、ADC ドライバ、abc-dq0 および dq0-abc 変換、変換、PWM 生成、2 つの電流制御ループ、1 つの電圧ループ、SFRA 呼び出しが含まれます。

3.1.2.2 保護方式

図 3-2 に、この設計のトリップ動作をセットアップするために使用するソフトウェア機能を示します。

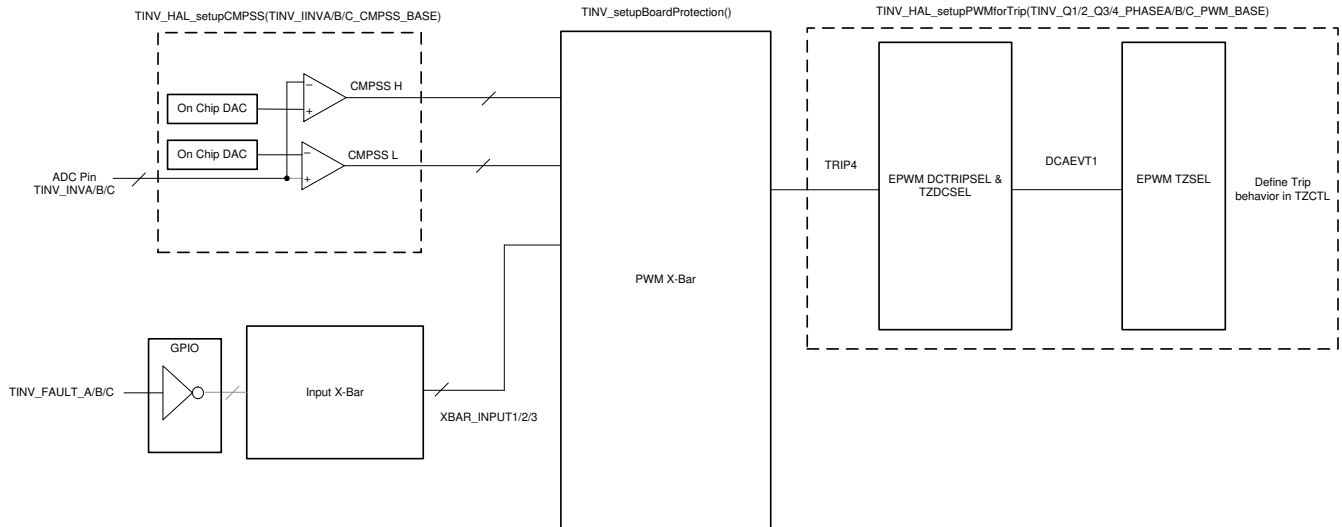


図 3-2. トリップ セットアップのソフトウェア図

トリップ フラグを更新し、必要に応じてラッチをリセットするため、TINV_updateFaultStaus() 関数が低速なバックグラウンド タスクで定期的呼び出されます。

トリップ イベントが発生した場合、PWM でトリップ フラグを個別にクリアする必要があります。この部分は通常、ISR で TINV_clearPWMTrips() を呼び出すことにより処理されます。

3.1.2.3 PWM スイッチング方式

図 3-3 に使用される PWM 構成を示します。相 A PWM モジュールのタイムベースのみを示しています。他も同一です。EPWM11 は、F28377 の SDFM 同期に使用されます。

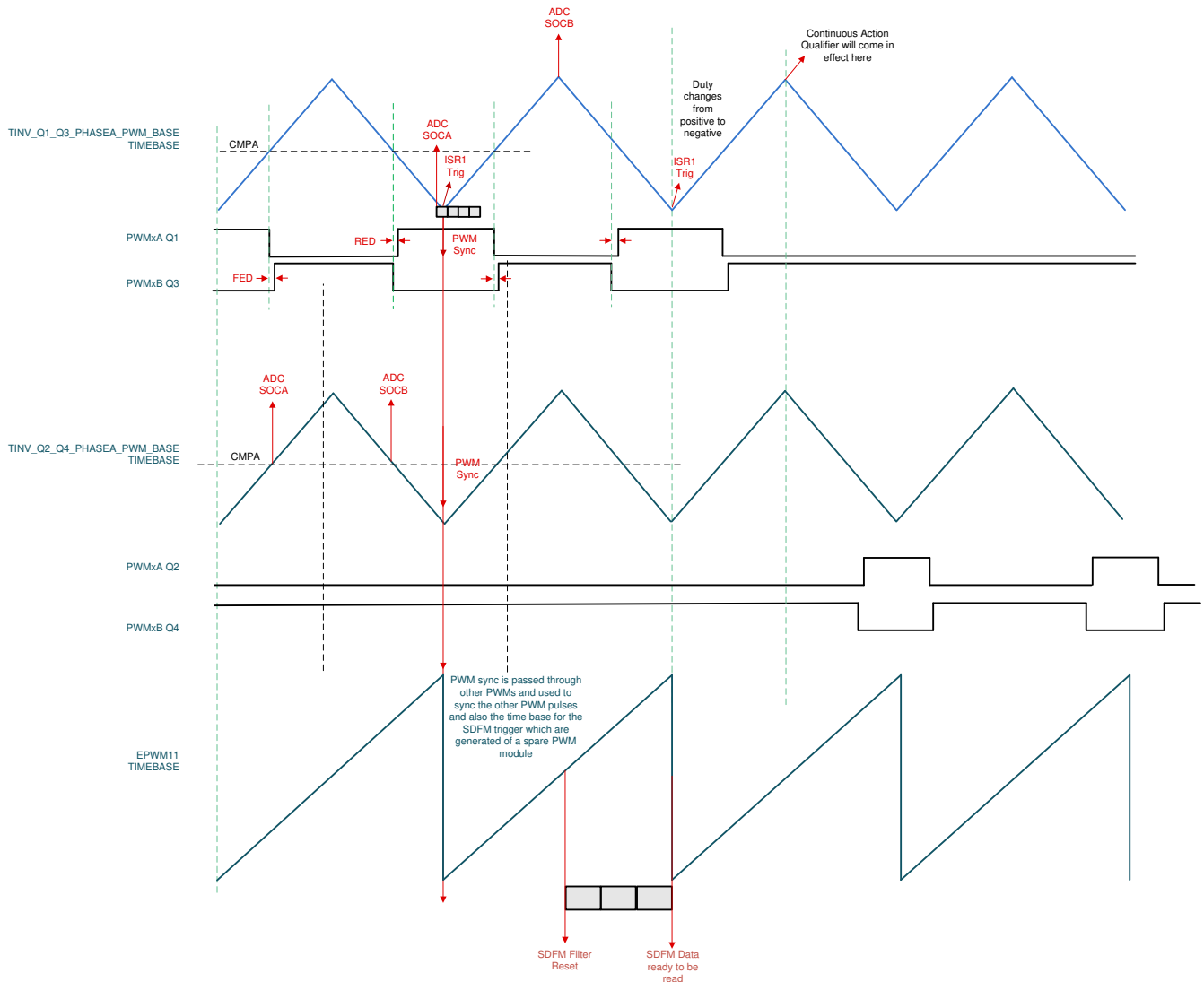


図 3-3. PWM 構成

3.1.2.4 ADC ローディング

同期動作を維持するため、すべての変換は次のようにトリガされます。

- TINV_Q1_Q3_A_PWM_BASE:EPWM1 TBCTR_D_CMPB → EPWM1_SOCA (緑)、各サイクルでトリガ
- TINV_Q1_Q3_A_PWM_BASE:EPWM1 TBCTR_D_CMPB → EPWM1_SOCA (青)、10 サイクルごとにトリガ
- TINV_Q2_Q4_A_PWM_BASE:EPWM2 TBCTR_U_CMPB → EPWM2_SOCA、各サイクルでトリガ
- TINV_Q2_Q4_A_PWM_BASE:EPWM2 TBCTR_D_CMPB → EPWM2_SOCA、各サイクルでトリガ
- TINV_Q1_Q3_A_PWM_BASE:EPWM3 TBCTR_PERIOD → EPWM3_SOCA、各サイクルでトリガ

表 3-4 に、TIDA-01606 ハードウェアでの F2837xD とのマッピングを示します。

表 3-4. ADC ローディングのアーキテクチャ F28379D

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

表 3-5 に、TIDA-01606 ハードウェアでの F280039C とのマッピングを示します。

表 3-5. ADC ローディングのアーキテクチャ F280039C

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

注

閉ループ動作では、ADC 電流読み取り値はレイアウトのノイズのため使用されません。代わりに、SDFM ベースの検出を使用してループを閉じます。そのため、送電網の電流を使用して電流ループを閉じるので、この変更を考慮して図を解釈する必要があります。

3.2 テストと結果

このセクションのすべてのラボは、両方の制御カードで実行できます。ただし、TMS320F280039C 制御カードを使用する場合は、以下の点に注意してください。

- GUI インターフェイスは使用できません。パラメータおよびラボを変更するには、`user_settings.h` ファイル内のパラメータを直接変更する必要があります。
- SFRA は使用できません。
- 空間ベクトル変調と、DC リンク コンデンサの中間点の制御を組み合わせることができます。送電網の中性点は、送電網から切断する必要があることに注意してください。次のユーザー設定パラメータを変更すると、2 つの新しい制御方式を動作させることができます。

- #define TINV_THIRD_HARMONIC_INJECTION_STATUS
TINV_THIRD_HARMONIC_INJECTION_ENABLE
- #define TINV_MIDDLE_POINT_CONTROL_STATUS TINV_MIDDLE_POINT_CONTROL_ENABLE

以下のパラメータは E7 ハードウェアのテストに使用されており、main.syscfg または tinv_settings.h のいずれかで構成できます。E6 を使用する場合は、default main.syscfg 値をそのまま使用します。E7 パラメータは次のコードに示されています：

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

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

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

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

注

新しい E7 ハードウェアをサポートするには、C2000Ware_DigitalPower_SDK のソフトウェアの一部の変更が必要です。最初の変更は、[セクション 3.1.1.2](#) ですでに説明したように、ファン GPIO の変更です (便宜上、ここに表示)。2 つ目のソフトウェア変更は、最新の E7 が V_PCC と V_SN について非反転電圧検出を使用しているため、デフォルトの SDK ソフトウェアでの電圧検出極性が反転することです。

tinv_user_settings.h でファン GPIO を変更する場合：

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

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

注

高電力テストの際には、デバッグ セッション中に CCS WATCH ウィンドウの TINV_fanSet 機能を使用して、ファンを有効にしてください。

tin.v.h の E7 電圧検出の極性変更には、以下を使用：

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

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

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

TINV_vGrid_A_sensed_prev_pu = TINV_vGrid_A_sensed_pu;

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

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

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

E6 の電圧検出の場合、tin.v.h から (デフォルトの SDK コードは許容可能)：

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

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

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

TINV_vGrid_A_sensed_prev_pu = TINV_vGrid_A_sensed_pu;

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

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

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

注

E7 ハードウェアの場合、(プリチャージ後に) 3 相リレーをすべてオンにするには、CCS WATCH ウィンドウの TINV_neutralRelaySet 機能を使用します。TINV_allRelaySet は E6 ハードウェアにのみ使用してください。

3.2.1 ラボ 1

このラボは、制御カードとドッキングステーションで実行することを意図しています。

<tinvs_settings.h> または main.syscfg ファイルのラボ番号を変更してプロジェクトをラボ 1 に設定します (powerSUITE プロジェクトを使用する場合、これは powerSUITE GUI で変更されます)。ほとんどのユーザーは、ラボ 2 にスキップしてテストフローを開始できます。

```
#define TINV_LAB 1
```

ここでは、user_settings.h ファイルの他のすべてのオプションはデフォルトのままにします。

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

図 3-4 に、コード実行時のソフトウェアの図を示します。

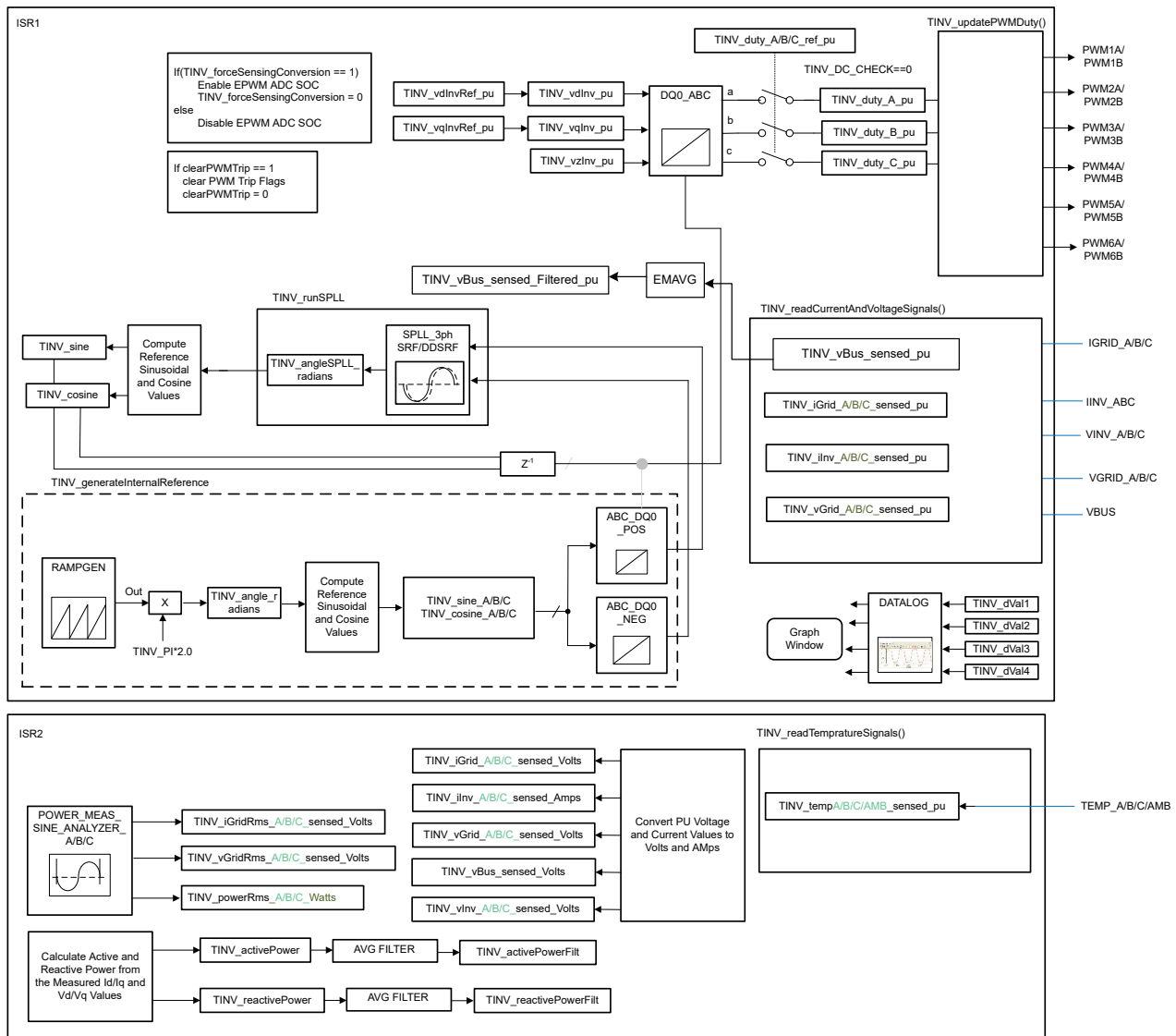


図 3-4. ラボ 1 のソフトウェア構成図

3.2.2 インバータ動作のテスト

ラボ 2、ラボ 3、ラボ 4 では、インバータ モードで電力段を動作させる手順を詳しく説明します。ラボ 2 は開ループでのインバータ動作モードです。ラボ 3 は閉電流ループでのインバータ動作モードです。ラボ 4 は、送電網に接続されたインバータの動作モードであり、これはハードウェア インザ ループ (HIL) プラットフォームでのみチェックされ、ハードウェアではチェックされません。端子 J13 と J18 の間に高電圧 (800VDC) を印加します。12 V の補助電源を端子 J3 に接続します。3 相スター型接続の抵抗性負荷は、端子 J14、J16、J17 の間に接続します。J30 は、高電圧の電源アースに接続されている保護接地端子です。

DC バス過電圧のチェックをすべてのインバータ ラボ (ラボ 1 ~ ラボ 5) に DC バス電圧のフィルタされた値を使用して追加します。TINV_filterAndCheckForBusOverVoltage() 関数は ISR1 から実行し、DC バスの過電圧状態をチェックします。過電圧状態では、この関数によりすべての PWM 出力がオフになり、システム動作状態が「バス過電圧状態」として登録されます。フィルタされた DC バス電圧は、瞬間検出された DC バス電圧から平均化関数 EMAVG を使用して計算されます。これらはすべて ISR1 内で計算されます。ユーザーは、tinv_user_settings.h で TINV_VBUS_OVERVOLT_LIMIT を設定できます。

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

フィードフォワードおよびデカップリング機能は ISR1 内に実装されており、電流ループを使用するすべてのインバータ ラボに追加されています。そのため、インバータ モードでは、ラボ 3 とラボ 4 でこれ (フィードフォワードとデカップリング) を行います。このフィードフォワードおよびデカップリング機能では、フィルタ処理された DC バス電圧を、ユーザー定義の最小バス電圧と比較して、クランプされフィルタ処理された DC バス電圧を計算します。これも ISR1 内で実行されます。このクランプされフィルタ処理された DC バス電圧と電流コントローラ出力を使用して、最終的にフィードフォワードおよびデカップリング機能を実装します。

SDFM ベースの電流検出では、すべてのインバータ ラボに過電流保護 (OCP) も追加されています。

3.2.2.1 ラボ 2

このラボでは、ハードウェアまたは HIL プラットフォーム上で、電力段を開ループで動作させます。図 3-5 に、実際のハードウェアのラボ設定を示します。

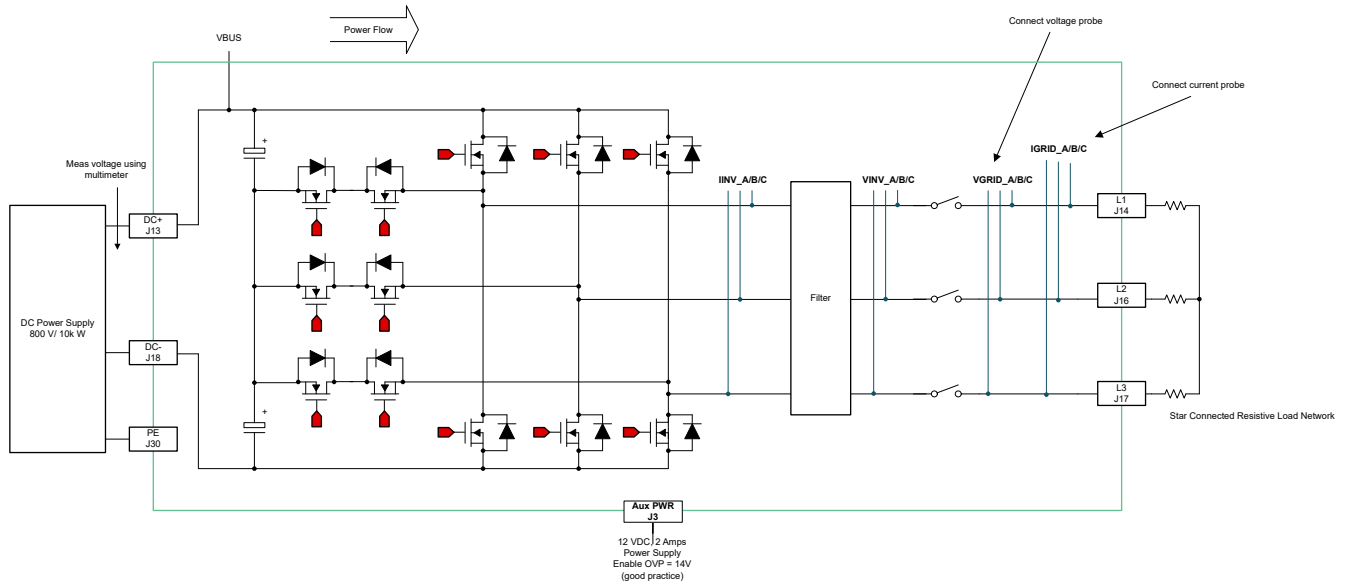


図 3-5. 抵抗性負荷を使用するインバータ モードのラボのセットアップ

図 3-6 に、ソフトウェア図を示します。

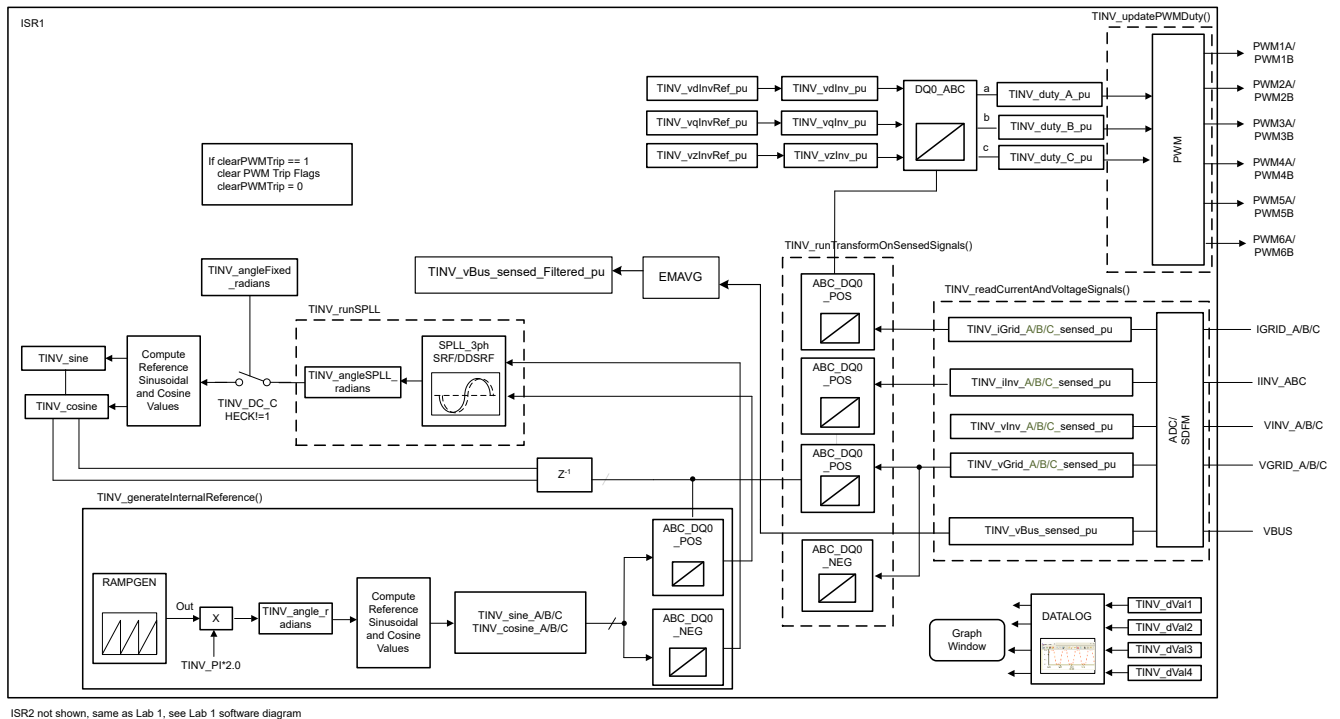


図 3-6. ラボ 2 のソフトウェア構成図

テストの構成に使用した機器の詳細については、ハードウェア テスト設定のセクションも参照してください。

<tin_v_settings.h> または main.syscfg ファイルのラボ番号を変更してプロジェクトをラボ 2 に設定します (powerSUITE プロジェクトを使用する場合、これは powerSUITE GUI で変更されます)。

`user-settings.h` ファイルには他にもいくつかの追加オプションがありますが、このユーザー ガイドに記載されているテストでは以下のオプションを使用します。

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

このチェックでは、ソフトウェアはハードウェアまたは HIL プラットフォームのいずれか、あるいは両方で実行されます。

まず、スター型接続負荷に適切な約 500Ω の抵抗性負荷を設定します (テストに適した電力定格を確認)。ただし、インバータ モードは無負荷でも開始できます。12V 補助電源を有効にします。コードをビルドしてロードし、`lab2.js` ファイルを使用して、CCS ウィンドウでウォッチ対象の変数を入力します。

- **CCS WATCH** ウィンドウをデバッグ モードで起動したら、連続的にリフレッシュしてテストを開始します。
- 高電力テストの際には、デバッグ セッション中に **CCS WATCH** ウィンドウの `TINV_fanSet` 機能を使用して、ファンを有効にしてください。
- `TINV_neutralRelaySet` に 1 を書き込んでリレーをオンにします。補助電源の消費電流は約 600mA になります。
- DC バス電圧 V_{bus} を徐々に 800V まで上昇させます。
- `TINV_clearPWMTrip` を 1 に設定して PWM トリップ信号をクリアします。これでスイッチング動作が開始し、電圧の正弦波が出力に表示され始めます。この時点で、補助電源の消費電流は約 800mA になります。ファンが有効なとき、補助電源の合計電流は約 1.4A です。
- `TINV_vdInvRef_pu` (デフォルト値は 0.835) は、インバータの AC 出力を開ループ形式で変化させるために使用可能な変調インデックスです。
- ラボ 3 の閉電流ループに進む前に、グラフ ウィンドウで検出された電圧および電流の測定データを確認します。☒ [3-7](#) に、C2000SDFM モジュールを使用して検出した送電網側の電流のグラフ ウィンドウを示します。目盛りは単位あたり (pu) で示されます。

```
#ifndef __TMS320C28XX_CLA__
TINV_dval1 = TINV_iGrid_A_sensed_pu;
TINV_dval2 = TINV_iGrid_B_sensed_pu;
TINV_dval3 = TINV_iGrid_C_sensed_pu;
TINV_dval4 = TINV_rgen.out;
DLOG_4CH_run(&TINV_dLog1);
#endif
```

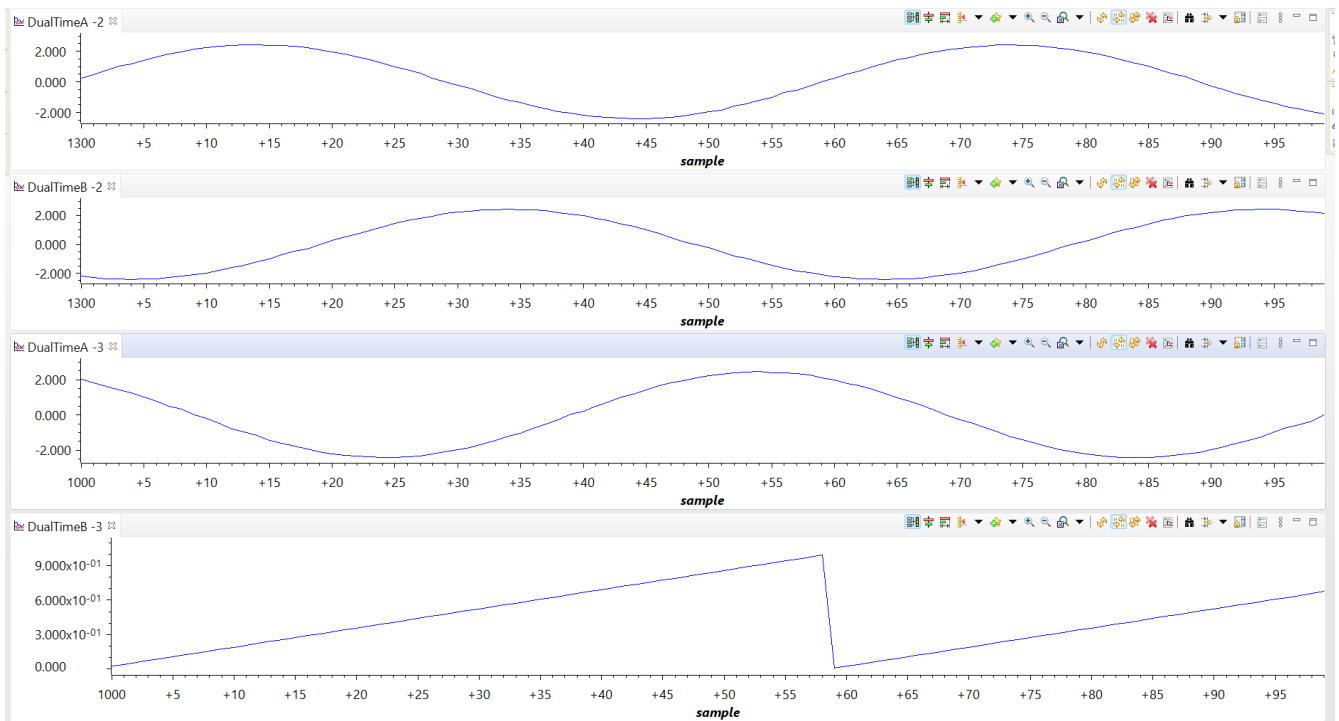


図 3-7. 検出された送電網電流

図 3-8 に、CCS のグラフ ウィンドウで監視される 3 つの送電網電圧を示します。目盛りは単位あたり (pu) で示されます。

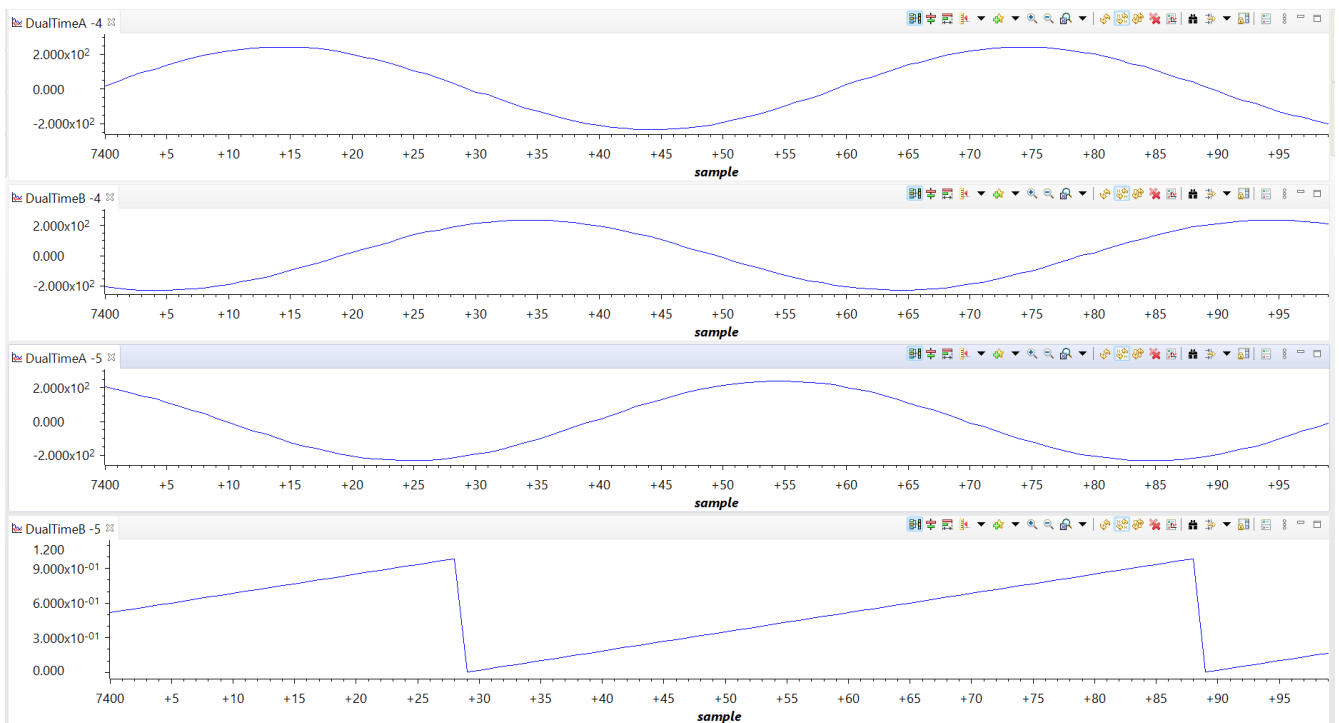
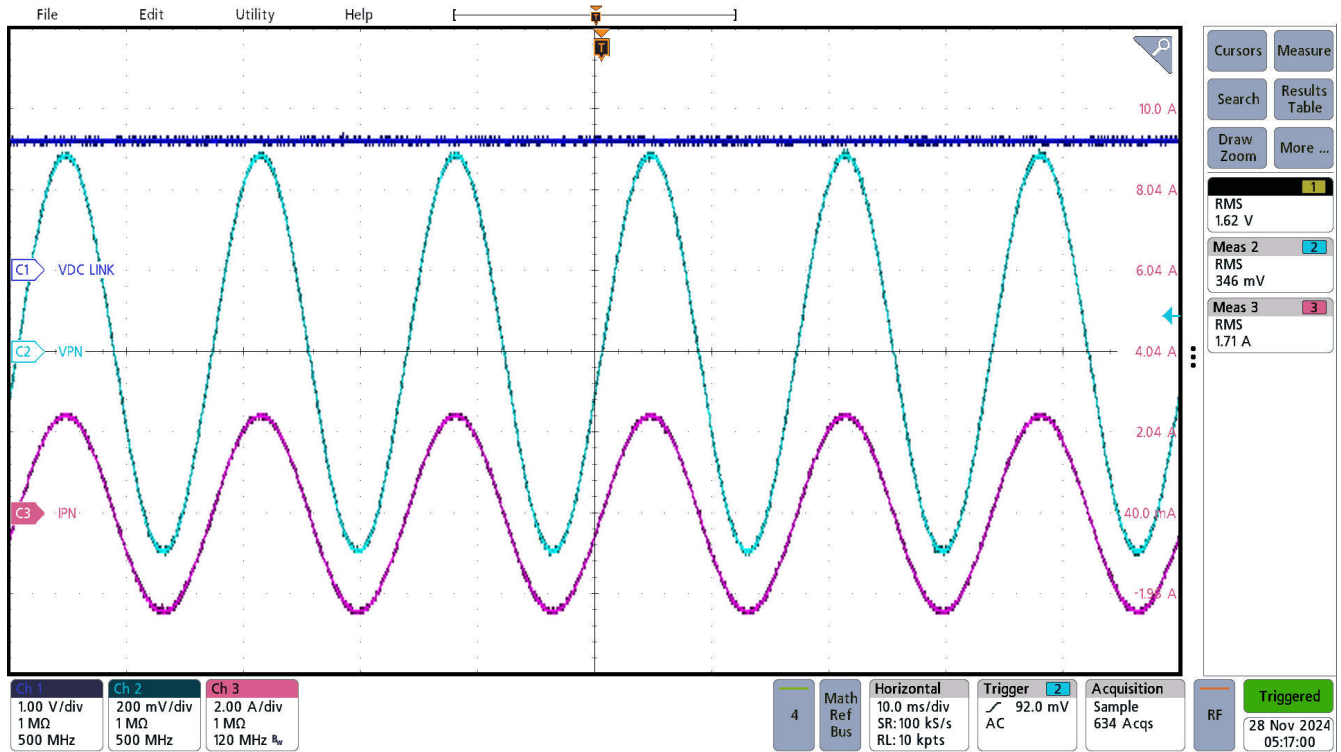


図 3-8. 検出された送電網電圧

図 3-9 に、173VAC および 0.88kW で開ループで動作するインバータの電圧および電流の波形を示します。



スコープの信号:チャンネル 1 - DC リンク電圧 (青)、チャンネル 2 - VPN AC 電圧 (ターコイズ)、チャンネル 3 - IPN AC 電流 (赤)。電圧プローブは 500:1 に縮小されます。

図 3-9. 開ループ インバータの電圧と電流の波形

3.2.2.2 ラボ 3

このラボでは、実際のハードウェアまたは HIL プラットフォーム上で、電力段を閉ループで動作させます。図 3-10 に、ソフトウェア図を示します。

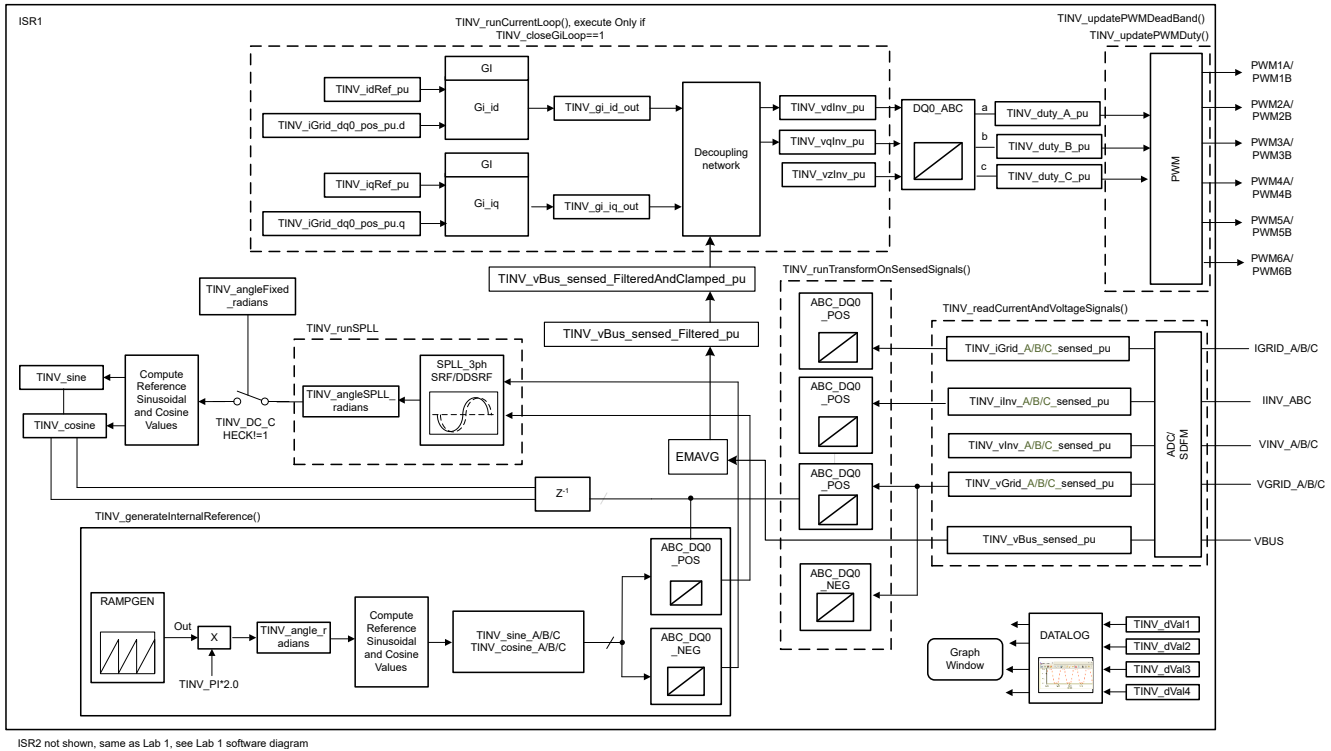


図 3-10. ラボ 3 のソフトウェア構成図

<tinvs_settings.h> または main.syscfg ファイルのラボ番号を変更してプロジェクトをラボ 3 に設定します (powerSUITE プロジェクトを使用する場合、これは powerSUITE GUI で変更されます)。

ユーザーの settings.h ファイルには他にもいくつかの追加オプションがありますが、このユーザー ガイドに記載されているテストでは以下のオプションを使用します。

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

このチェックでは、ソフトウェアはハードウェアまたは HIL プラットフォームのいずれか、あるいは両方で実行されます。

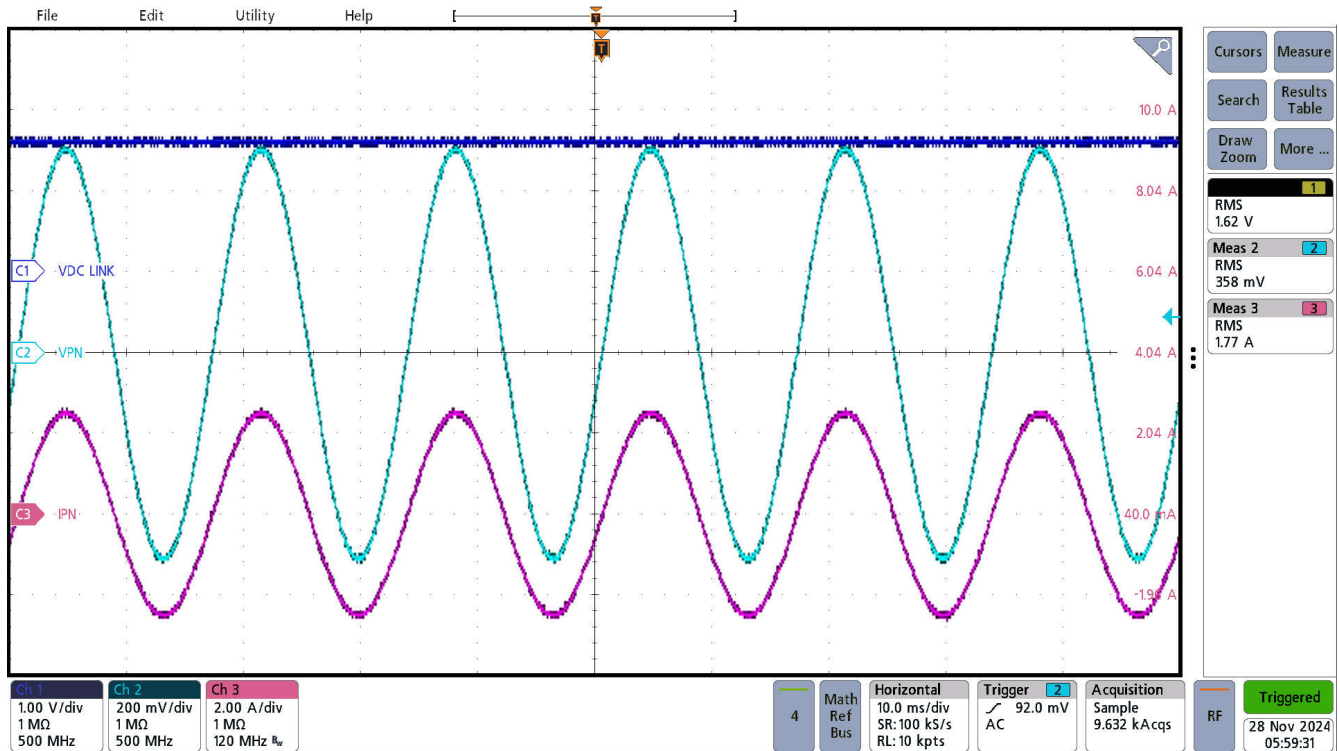
テストの構成に使用した機器の詳細については、ハードウェア テスト設定のセクションを参照してください。この時点では、高圧電力を基板に供給しないでください。

- まず main.cfg を起動し、プロジェクト オプションでラボ 3 を選択します。補償器スタイル (PI 補償器) と調整ループ (電流ループ) が自動的に設定されます。ここで [Run Compensation Designer] アイコンをクリックすると、Compensation Designer ツールが起動し、powerSUITE ページで指定したパラメータを持つ電流ループプラントモデルが表示されます。
- 制御ループの実行に使用する電流補償係数を次のコードに示します。これらの係数は、必要なループ帯域幅と位相マージンを満たすように変更できます。抵抗性負荷を使用する場合の理想的な係数は、送電網接続に使用する係数とはわずかに異なります。これは、送電網のインピーダンスが非常に小さいためです。補償器設計の伝達関数と応答は、図 3-11 に示すようになります。

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

- 比例および積分利得値が条件を満たしているなら、[Save COMP] をクリックします。これにより、補償器の値がプロジェクトに保存されます。**Compensation Designer** を閉じて、**powerSUITE** ページに戻ります。
- コードをビルドしてロードし、**lab3.js** ファイルを使用して、**CCS** ウィンドウでウォッチ対象の変数を入力します。
- 高電力テストの際には、デバッグ セッション中に **CCS WATCH** ウィンドウの **TINV_fanSet** 機能を使用して、ファンを有効にしてください。
- **TINV_neutralRelaySet** に「1」を書き込んでリレーをオンにします。補助電源の消費電流は約 **600mA** になります。
- まず、適切な約 **500Ω** の抵抗性負荷を設定します。ただし、インバータ モードは無負荷でも開始できます。
- **DC** バス電圧 **Vbus** を徐々に **800V** まで上昇させます。
- **TINV_clearPWMTrip** を 1 に設定して **PWM** トリップ信号をクリアします。これでスイッチング動作が開始し、電圧の正弦波が出力に表示され始めます。この時点で、補助電源の消費電流は約 **800mA** になります。
- **TINV_clearPWMTrip** が設定されるとすぐに、**TINV_closeGiLoop** 変数がイネーブルになり、閉電流ループ動作が開始します。
- **TINV_idRef_pu** はコマンドの電流基準で、デフォルトではスタートアップ時に **0.005pu** に設定されます。これを徐々に変化させて出力 **AC** 電圧を上げ、測定される電流がコマンドで指定された値に追従することを確認します。
- ラボ 3 の電流ループを閉じる前に、**WATCH** ウィンドウの **TINV_idRef_pu** が低い設定 (**0.005pu**) になっていることを確認します。
- **800V** 入力電圧で **id_ref** を **0.06pu** まで徐々に上げ、出力電力を **0.9kW** (各相約 **300W**) まで向上させます。図 **3-11** に、電力アナライザとオシロスコープの波形を示します。

- 図 3-11 に、0.9kW で開ループで動作するインバータの電圧および電流の波形を示します。



スコープの信号:チャンネル 1 - DC リンク電圧 (青)、チャンネル 2 - AC 電圧 (ターコイズ)、チャンネル 3 - AC 電流 (赤)。電圧プローブは 500:1 に縮小されます。

図 3-11. インバータの開ループ動作

- このラボのソフトウェアには SFRA が統合されているため、ハードウェアを測定して、設計した補償器が十分なゲインマージンと位相マージンを提供していることを検証できます。SFRA を実行するには、プロジェクトを実行している状態で、.cfg ページから SFRA アイコンをクリックします。SFRA GUI がポップアップ表示されます。
- SFRA GUI でデバイスのオプションを選択します。例として、F28379D の場合には浮動小数点を選択します。[Setup Connection] をクリックします。ポップアップ ウィンドウで [Boot on Connect] オプションのチェックを外し、適切な COM ポートを選択します。[OK] をクリックします。SFRA GUI に戻り、[Connect] をクリックします。
- SFRA GUI がデバイスに接続します。これで [Start Sweep] をクリックして、SFRA 掃引を開始できます。SFRA 掃引が完了するまでには数分かかります。SFRA GUI のプログレス バーを確認したり、UART の動作を示す制御カード裏面の青色 LED の点滅をチェックすることで、動作を監視できます。終了すると、図 3-12 のように開ループプロットによるグラフが表示されます。これにより、設計した補償器が確かに安定していることを検証できます。上記の係数を持つプラントおよび開ループの SFRA を、それぞれ図 3-12 および図 3-13 に示します。この操作により、電流補償器設計を検証できます。システムを安全に停止させるには、入力 DC 電圧をゼロまで下げます。
- Compensation Designer の上記の係数セットは、堅牢で安定しています。コマンドで指定された基準に対して電流のトラッキング性能が変動しているように見える場合は、以下の係数セットを使用できます。係数を変更するには、powerSUITE ページから Compensation Designer ツールを再起動する必要があります。

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

- 比例および積分利得値が条件を満たしているなら、[Save COMP] をクリックします。これにより、補償器の値がプロジェクトに保存されます。

- 新しい係数セットを使用した電流モードのインバータに対するプラントおよび開ループの SFRA 応答を、それぞれ図 3-12 および図 3-13 に示します。

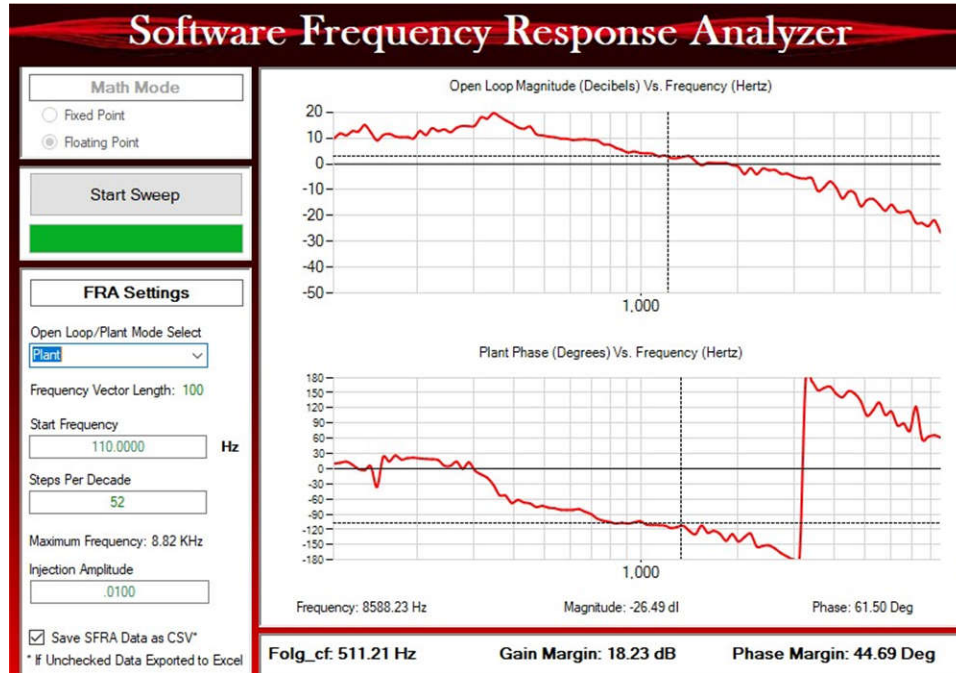


図 3-12. 電流ループに対するインバータ SFRA プラントの応答

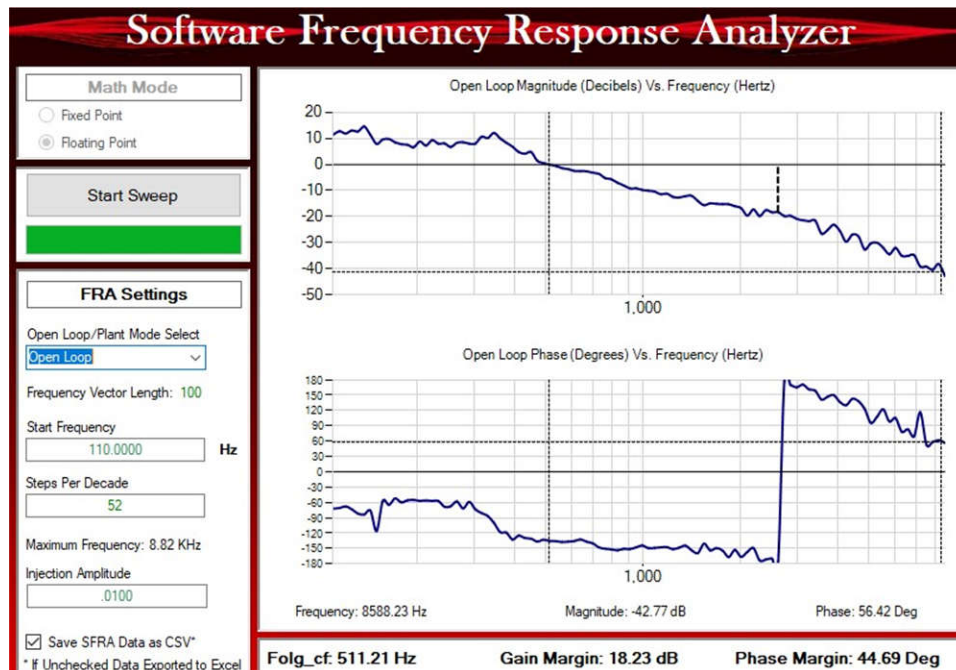


図 3-13. 電流ループに対するインバータ SFRA ループの応答

3.2.2.3 ラボ 4

このラボでは、HIL プラットフォーム上で、送電網に接続された閉ループで電力段を動作させます。図 3-14 に、ソフトウェア図を示します。

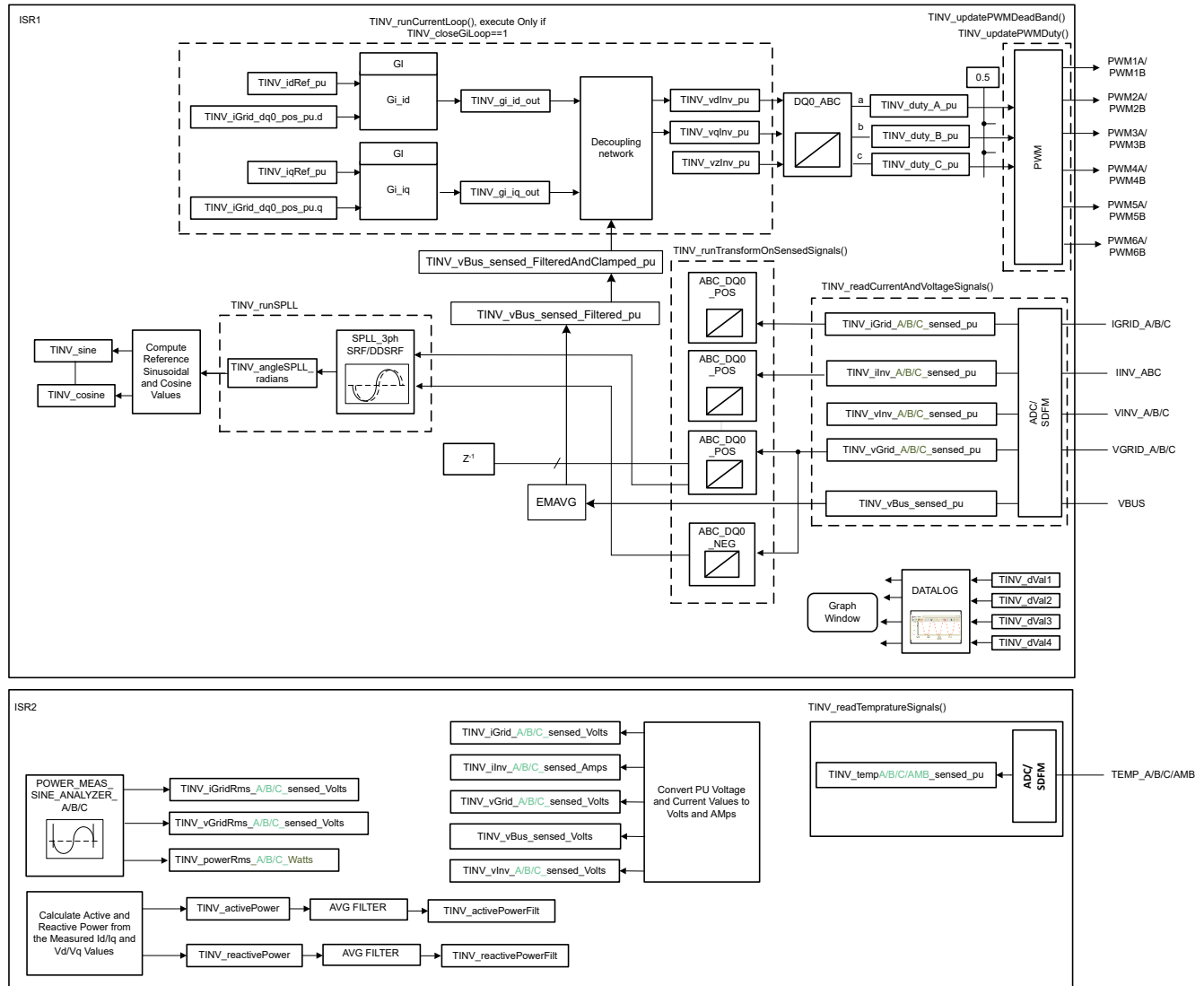


図 3-14. ラボ 4 のソフトウェア構成図

注

このラボは、HIL セットアップでのみ検証されます。

<tinvs_settings.h> または main.syscfg ファイルのラボ番号を変更してプロジェクトをラボ 4 に設定します (powerSUITE プロジェクトを使用する場合、これは powerSUITE GUI で変更されます)。

ユーザーの `settings.h` ファイルには他にもいくつかの追加オプションがありますが、このユーザー ガイドに記載されているテストでは以下のオプションを使用します。

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

このチェックでは、ソフトウェアは HIL プラットフォームで実行されます。

コードをビルドしてロードし、`lab4.js` ファイルを使用して、CCS ウィンドウでウォッチ対象の変数を入力します。

- 高電力テストの際には、デバッグ セッション中に CCS WATCH ウィンドウの `TINV_fanSet` 機能を使用して、ファンを有効にしてください。
- DC バス電圧 V_{bus} を徐々に 800V まで上昇させます。
- `TINV_startPowerStage` 変数に「1」を入力し、送電網の AC 電圧を 230V_{RMS}、L-N (400V_{L-L}) に上昇させます。
- 突入電流制限抵抗が加熱してバーンアウトする可能性があるため、電圧が 230V_{RMS} に達したらずぐに `TINV_neutralRelaySet` に「1」を書き込み、適切な送電網接続を設定してリレーをオンにします。これで、電流が送電網に供給されます。
- この `TINV_idRef_pu` 変数を徐々に 0.6pu まで上昇させます。この時点で、相ごとの電力は約 1.9kW になります。

SFRA を使用して電流ループ帯域幅を測定します。図 3-15 および 図 3-16 に、HIL プラットフォームで測定された電流モードで動作するインバータのプラント応答とループ応答を示します。

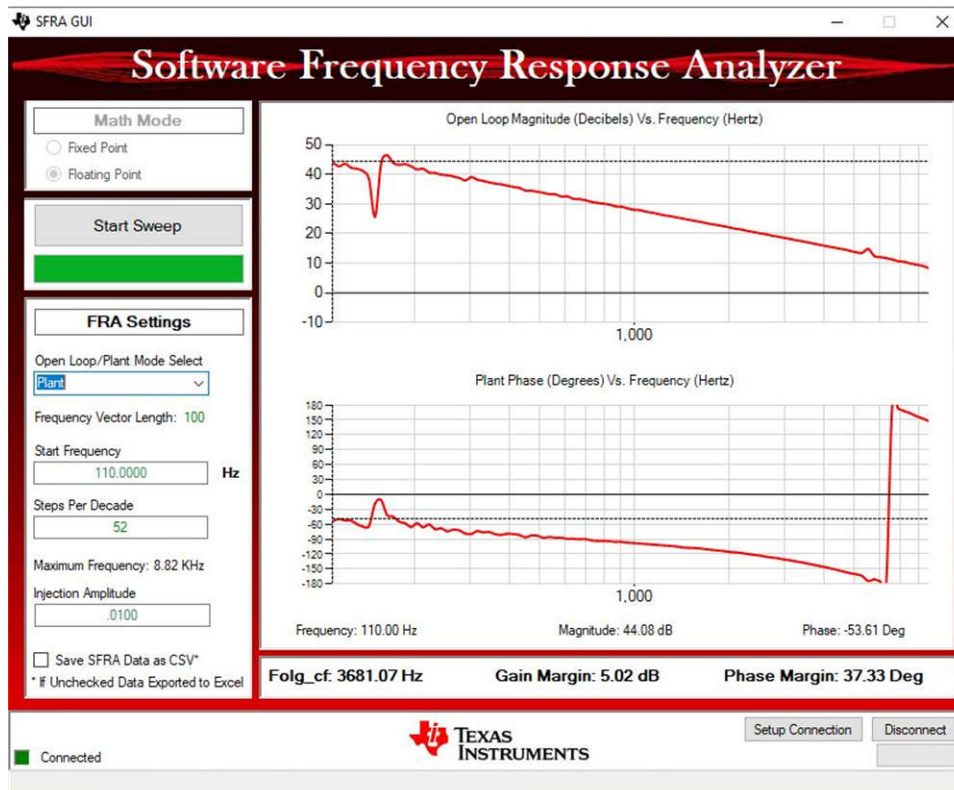


図 3-15. HIL で測定したインバータ SFRA プラントの応答

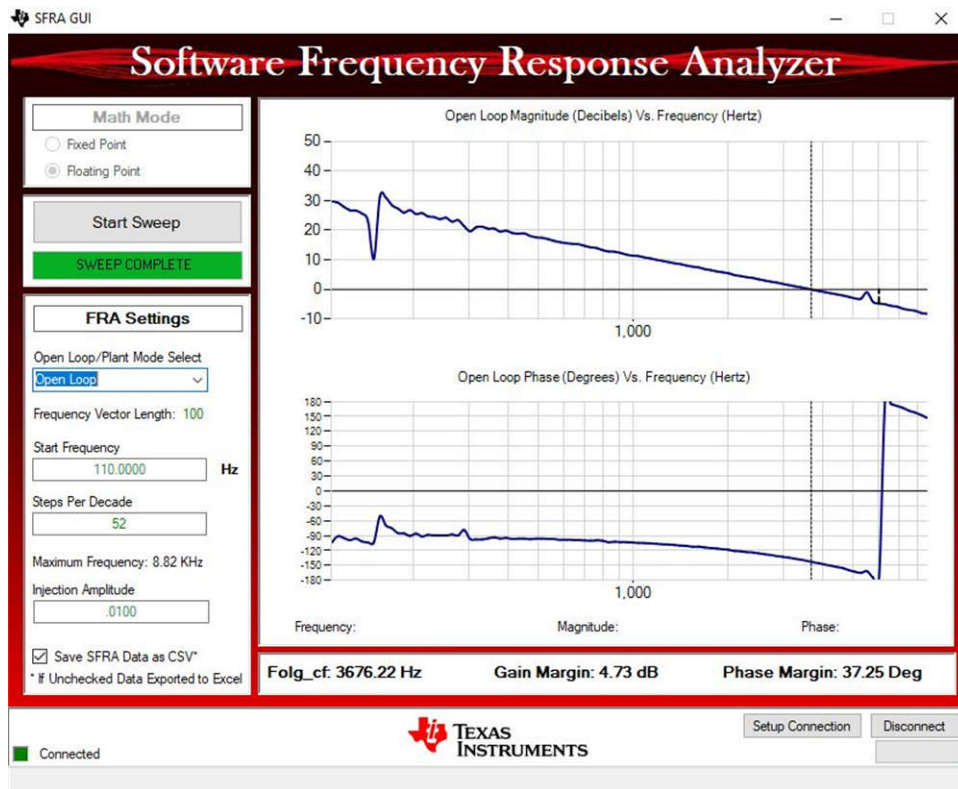


図 3-16. HIL で測定したインバータ SFRA ループの応答

3.2.3 PFC 動作のテスト

ラボ5、ラボ6、ラボ7では、PFC モードで電力段を動作させる手順を詳しく説明します。ラボ5は開ループでの PFC 動作モードです。ラボ6は閉電流ループでの PFC 動作モードです。ラボ7は、閉電圧および電流ループでの PFC 動作モードであり、HIL プラットフォームと実際のハードウェアの両方を使用してこれをチェックします。

DC バス過電圧のチェックをすべての PFC ラボ(ラボ5～ラボ7)に DC バス電圧のフィルタされた値を使用して追加します。TINV_filterAndCheckForBusOverVoltage() 関数は ISR1 から実行し、DC バスの過電圧状態をチェックします。過電圧状態では、この関数によりすべての PWM 出力がオフになり、システム動作状態が「バス過電圧状態」として登録されます。フィルタされた DC バス電圧は、瞬間検出された DC バス電圧から平均化関数 EMAVG を使用して計算されます。これらはすべて ISR1 内で計算されます。

フィードフォワードおよびデカップリング機能は ISR1 内に実装されており、電流ループを使用するすべての PFC ラボに追加されています。そのため、PFC モードでは、ラボ6とラボ7でこれを行います。このフィードフォワードおよびデカップリング機能では、フィルタ処理された DC バス電圧を、ユーザー定義の最小バス電圧と比較して、クランプされフィルタ処理された DC バス電圧を計算します。これも ISR1 内で実行されます。このクランプされフィルタ処理された DC バス電圧と電流コントローラ出力を使用して、最終的にフィードフォワードおよびデカップリング機能を実装します。

SDFM ベースの電流検出では、すべての PFC ラボに過電流保護 (OCP) も追加されています。

図 3-17 に、ハードウェア設定を示します。DC 端子 J13 および J18 は、電子負荷に接続されています。12V の補助電源を端子 J3 に接続します。3 相 AC ソースは、端子 J14、J16、J17 (A、B、C) に接続されています。J30 は、ソース PE に接続されている PE 端子です。テストの構成に使用した機器の詳細については、ハードウェア テスト設定のセクションを参照してください。

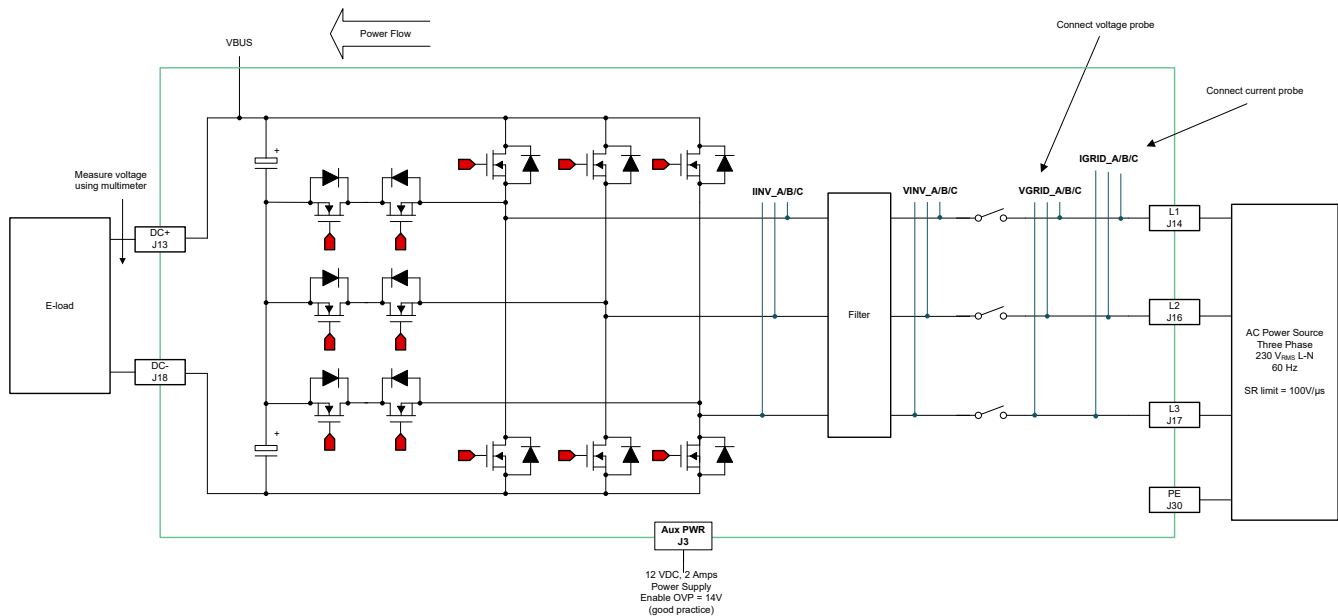


図 3-17. PFC モードのテスト構成

3.2.3.1 ラボ 5

これは最初の PFC ラボです。このラボでは検出がチェックされ、clearPWMTrip が 1 に設定されるまでスイッチング動作は発生しません。

PFC モードのハードウェア設定は、図 3-17 で説明されています。PFC は 30V_{RMS} などの低電圧で起動し、2kΩ の抵抗を接続することを推奨します。

<tinv_settings.h> または main.syscfg ファイルのラボ番号を変更してプロジェクトをラボ 5 に設定します (powerSUITE プロジェクトを使用する場合、これは powerSUITE GUI で変更されます)。

この条件では、コンバータが整流器として動作し、力率補正なしで整流された電流が流れます。このビルドでは、ソフトウェアのフェーズ ロック ループ (SPLL) のロックも安全に検証できます。

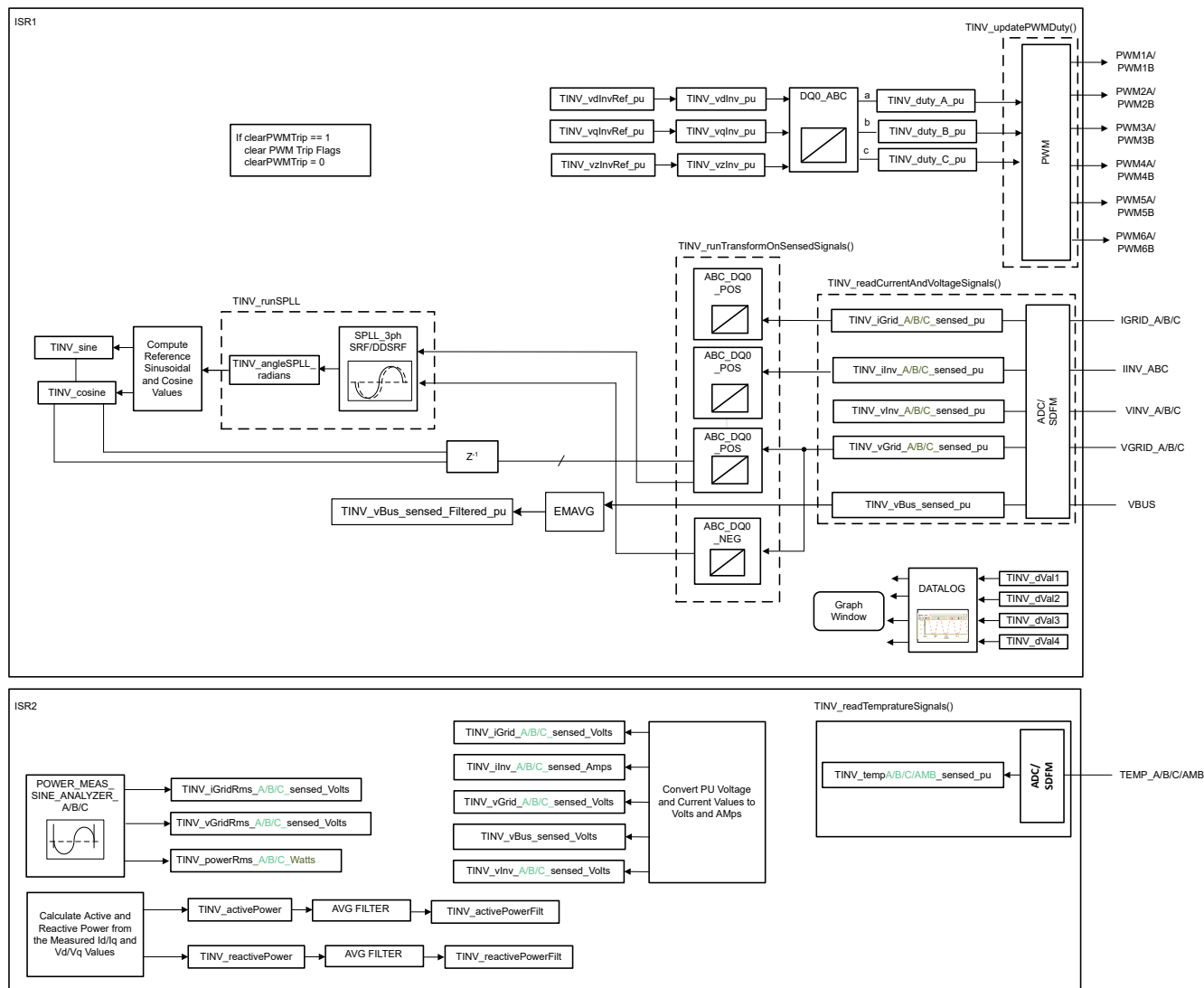


図 3-18. ラボ 5 のソフトウェア構成図

データロガーに次の変数が入力されます。

```

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

```

送電網の周波数が正しく指定されていることを確認します。送電網の周波数は、powerSUITE ベースのプロジェクトの sysconfig ページで変更できます。powerSUITE ベースのプロジェクトを使用していない場合は、tin_v_settings.h ファイルを変更してください。

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

コードをビルドしてロードし、lab5.js ファイルを使用して、CCS ウィンドウでウォッチ対象の変数を入力します。

PLL ロックは、バッファをプロットすることによってチェックできます。graph1.graphprop を使用して、[Tools] → [Graph] → [Dual Time] でバッファを確認します。

コサイン変換が使用されるため、Vgrid がピークになると、角度は 0 になります。

TINV_neutralRelaySet に 1 を書き込んでリレー閉じます。

最初は、安全のため $30V_{RMS}$ を使用してテストを実行できるので、AC 電源を $30V_{RMS}$ まで安全に上昇させ、CCS デバッグ ウィンドウのグラフを観察して PLL がロックされることを確認します。図 3-19 に、WATCH ウィンドウからの低電圧フェーズ ロック ループのチェックを示します。

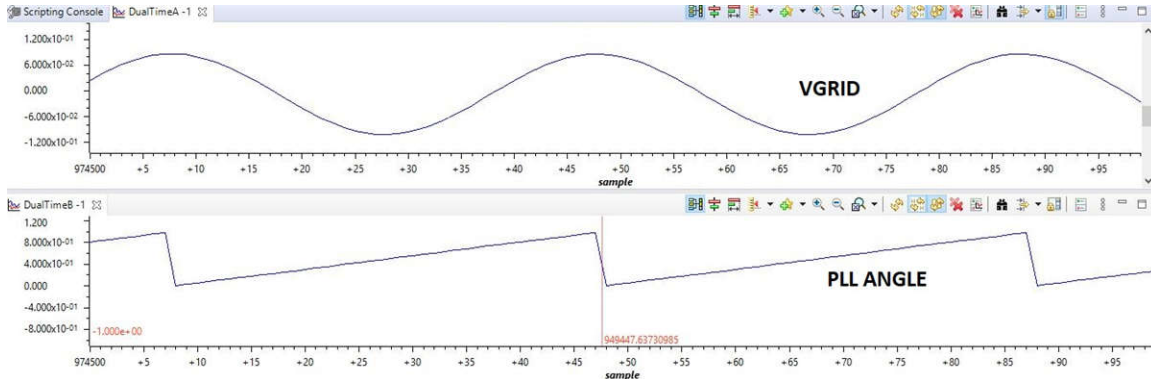


図 3-19. PLL - 送電網電圧の同期

PLL がロックされない場合、コマンドを「1」に設定して tinv_reset_PLL コマンドを発行できます。これにより、モジュール内のエラーをゼロにし、すべてのメモリ素子をゼロにするタスクが開始されます。

同様に、CCS のグラフ ウォッチ ウィンドウを使用して、送電網からすべての相に流れる電流を確認できます。図 3-20 に、グラフ windowCheck から検出された送電網電流を示します。WATCH ウィンドウで 3 相の送電網電流を確認してください。



図 3-20. 検出された送電網電流 - PFC モード

ラボ 5 で昇圧動作を確認するには、次の手順を実行します。

- 補助電源をオンにして $12V$ に設定し、コードを実行してデバッグします。
- J13 端子と J18 端子に許容可能な負荷を接続します。高い負荷抵抗 (約 $2k\Omega$) を使用してください。そうしないと、大きな突入電流が発生して過電流フラグがトリガされる可能性があります。電子負荷は、定電圧 (CV) モードでも使用できます。クランプ電圧を、入力 AC 位相電圧の 2.6 倍より高く設定します。たとえば、 $30V_{RMS}$ の AC 電圧を使用する場合は、電子負荷の CV 電圧を少なくとも $78V$ (DC リンクの昇圧されていない整流電圧より高い値) に設定します。

- 高電力テストの際には、デバッグ セッション中に CCS WATCH ウィンドウの TINV_fanSet 機能を使用して、ファンを有効にしてください。
- 3 相端子に 30V_{RMS} の AC 電圧を印加します。
- すぐに TINV_neutralRelaySet に 1 を書き込んでリレーをオンにします。DC 端子の両端に電圧が印加され始めます。
- TINV_clearPwmTrip を 1 に設定して PWM トリップをクリアし、DC 電圧がわずかに昇圧されることを確認します。

PFC 動作が開始する前に、Vbus の負荷によって整流された電流が引き込まれます。clearPWMTrip を 1 に設定するとすぐに、DC 電圧がわずかに昇圧されることが確認できます。

注

PFC のラボ (ラボ 5、ラボ 6、ラボ 7) では、コンバータが整流器として動作し、力率補正を行わずに整流された電流が流れる場合があります。ただし、TINV_clearPwmTrip を 1 に設定すると、スイッチング動作が発生しなくなり、ゲート信号はオフのままになります。

これは、3 相のいずれかで設定される過電流フラグまたは DSAT フラグ (InvA_overcurrent、InvB_overcurrent、DSATA、DSATB) があるからです。これは、次の 3 つの状況で発生します。

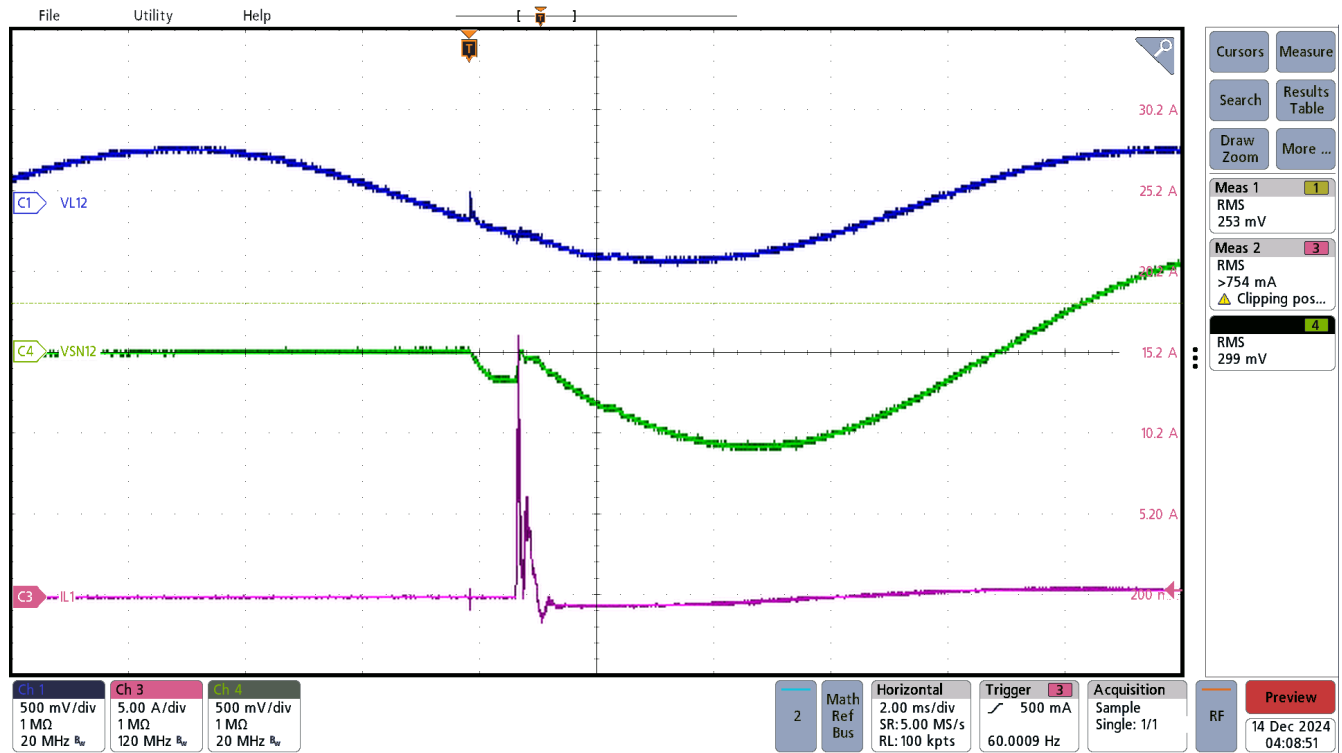
- リレーを閉じたときに突入電流が発生し、3 相のいずれかで過電流トリップが発生する。
- TINV_clearPwmTrip を 1 に設定したときに、スイッチング動作によってフラグの 1 つが設定されます。
- 電流および電圧ループを閉じるために TINV_StartpowerStage を 1 に設定します。

EPWM の TZFLG は 0X000C に設定され、この状況ではスイッチングは発生しません。そのため、負荷抵抗を大きくして突入電流によってトリップ条件が発生しないようにし、EPWM TZFLG が 0x0004 から 0x0000 に変化してスイッチングが発生するようにします。

FLG が 0x000C に設定されると、TINV_reset_fault_status を実行してフォルトをリセットしても、フォルトはクリアされますが、PWM 動作は観測されません。

ここでの目標は、補助電源が起動して、コードがデバッグされたらすぐに、コントローラがトリップ状態にならないように、すべてのフォルト (InvA_overcurrent、InvB_overcurrent、DSATA、DSATB など) をゼロに設定することです。

可能であれば、AC ソースのスルーレートを 100V/μs に制限します。これにより、突入電流で AC ソースの OCP がトリップするのを防げます。これは、[図 3-21](#) に示すように、リレーを長時間オープンのままにしている場合に、EMI フィルタ コンデンサの電圧が 0V 近くまで減衰して発生する可能性があります。その後、リレーが閉じられると、大きな突入電流がリアクティブな負荷に入り、AC ソースの OCP 機能をトリップする可能性があります。



スコープの信号:チャンネル 1 - 入力時 V_{L_L} (青)、チャンネル 4 - EMI コンデンサ C68 VSN 間の電圧 (薄緑)、チャンネル 3 - AC 入力電流 (赤)。電圧プローブは 500:1 に縮小されます。

図 3-21. プリチャージ期間後のリアクティブな負荷電圧減衰に起因する突入電流

3.2.3.2 ラボ 6

このビルドレベルで PFC の電流ループをチェックします。ラボ 6 は主に、電流ループの調整と性能の最適化を目的としています。スーパーバイザ電圧ループを使用せずに大電力から開始すると、過電流イベントによって電圧が上昇してスイッチが破損する可能性があるため、このラボは低電圧かつローパワーで開始するのが安全です。DC 端子での過電流トリップおよび高電圧を避けるため、TINV_idRef_pu 変数を注意深く設定してください。また、TINV_idRef_pu は、PFC 動作モードでは負の符号で定義され、インバータ動作モードでは正の符号で定義されます。

図 3-22 に、ラボ 6 を実行するためのソフトウェアフローを示します。

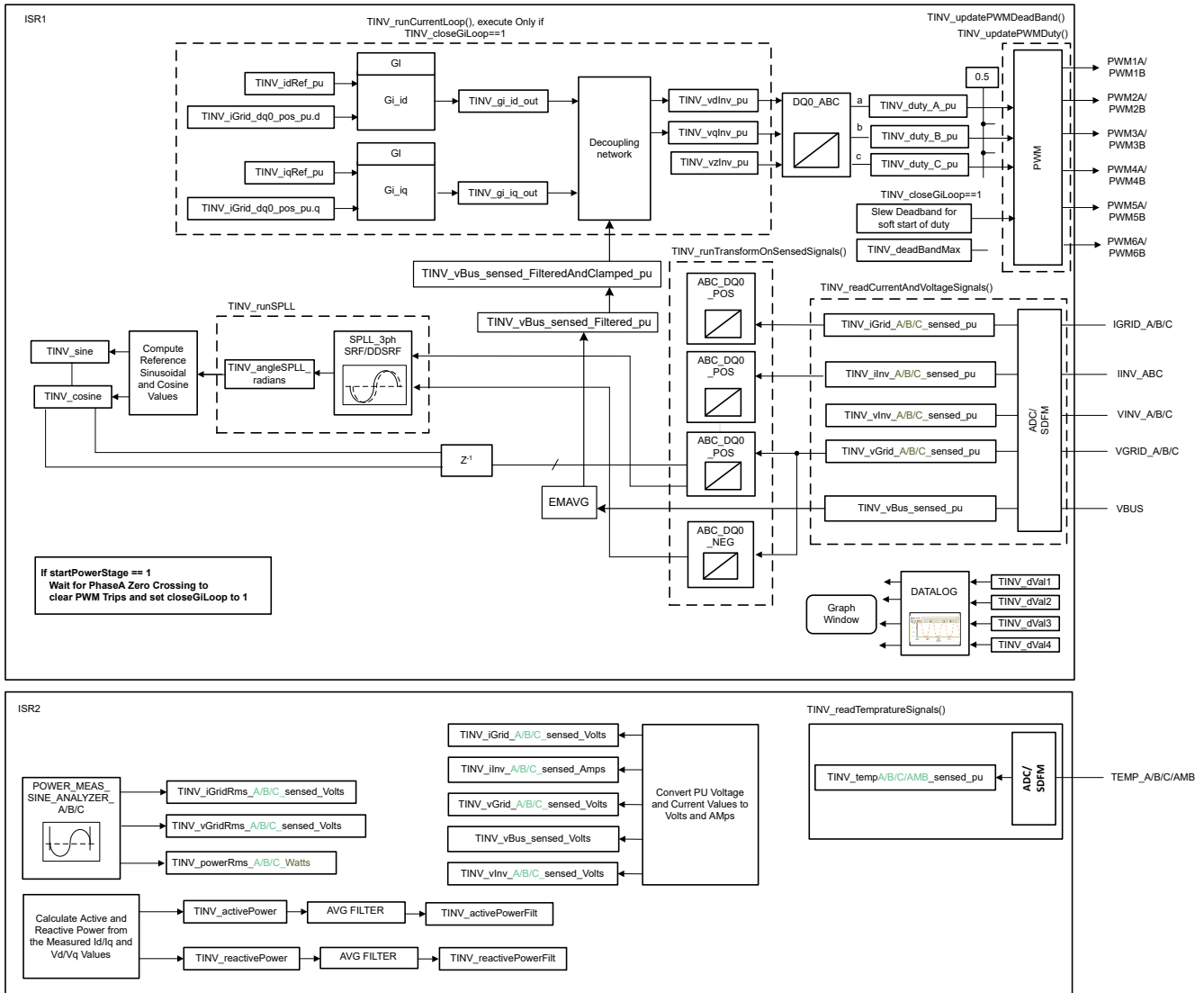


図 3-22. ラボ 6 のソフトウェア構成図

<tinvs_settings.h> または main.syscfg ファイルのラボ番号を変更してプロジェクトをラボ 6 に設定します (powerSUITE プロジェクトを使用する場合、これは powerSUITE GUI で変更されます)。

ユーザーの settings.h ファイルには他にもいくつかの追加オプションがありますが、この設計ガイドに記載されているテストでは以下のオプションを使用します。

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

このチェックでは、ソフトウェアはハードウェアまたは HIL プラットフォームのいずれか、あるいは両方で実行されます。

テストの構成に使用した機器の詳細については、[ハードウェア テスト設定](#)のセクションを参照してください。高電圧電力はまだ基板に供給しないでください。

- まず `main.syscfg` を起動し、プロジェクト オプションでラボ **6** を選択します。補償器スタイル (PI 補償器) と調整ループ (電流ループ) が自動的に設定されます。ここで **[Run Compensation Designer]** アイコンをクリックすると、**Compensation Designer** ツールが起動し、**powerSUITE** ページで指定したパラメータを持つ電流ループ プラント モデルが表示されます。
- 制御ループの実行に使用する電流補償係数を次のコードに示します。これらの係数は、必要なループ帯域幅と位相マージンを満たすように変更できます。抵抗性負荷を使用する場合の理想的な係数は、送電網接続に使用する係数とはわずかに異なります。これは、送電網のインピーダンスが非常に小さいためです。補償回路設計の伝達関数と応答は、[図 3-23](#) に示すようになります。

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

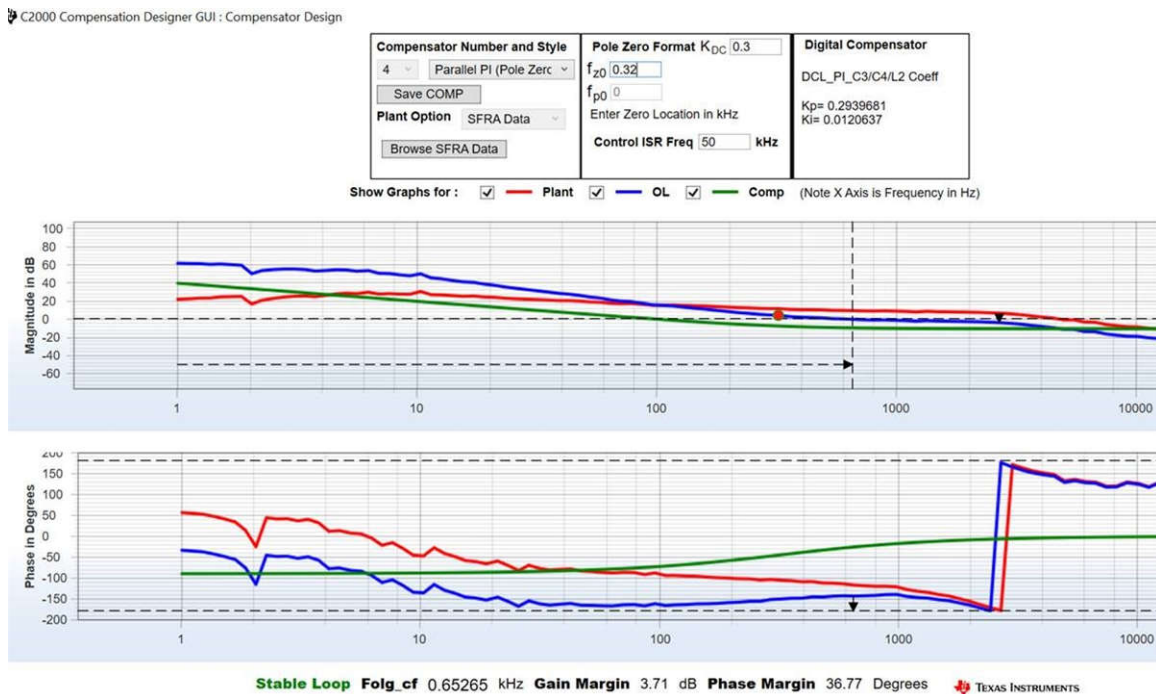


図 3-23. 補償器設計の GUI - 電流ループの PI 係数

- 比例および積分利得値が条件を満たしているなら、**[Save COMP]** ボタンをクリックします。これにより、補償器の値がプロジェクトに保存されます。**Compensation Designer** を閉じて、**powerSUITE** ページに戻ります。
- 補助電源をオンにし、**12V** に設定します。コードをビルドしてロードし、`lab6.js` ファイルを使用して **CCS** ウィンドウに監視する変数を入力します。
- 電子負荷の **CV** 電圧を **100V**、**20A** 制限に設定します。
- AC** 入力電圧を適切な電流制限を使用して **30V_{RMS}** に設定します。
- 高電力テストの際には、デバッグ セッション中に **CCS WATCH** ウィンドウの **TINV_fanSet** 機能を使用して、ファンを有効にしてください。
- AC** 電源をオンにした後、すぐに `TINV_neutralRelaySet` に **1** を書き込んでリレーをオンにします。**AC** 電源をオンにした後すぐ (**2 秒以内**) にリレーがオンになるようにします。
- `TINV_idRef_pu` を **-0.015pu** に設定します
- PFC** モードを開始するには、`TINV_startPowerStage` 変数に **1** を入力します。正弦波信号の電流が送電網から流れ (ローパワーでの電流であるため、いくらかの高調波を含む)、**vBus** 上に昇圧が見られます。出力電圧は **75V** から約 **100V** に昇圧されます。
- 負荷が増加すると、電流は正弦波になります。これにより、**PFC** が **30V_{RMS}** で開始したことを検証できます。
- 次に、**120V_{RMS}** および **230V_{RMS}** 入力で **PFC** テストを再度実施します。

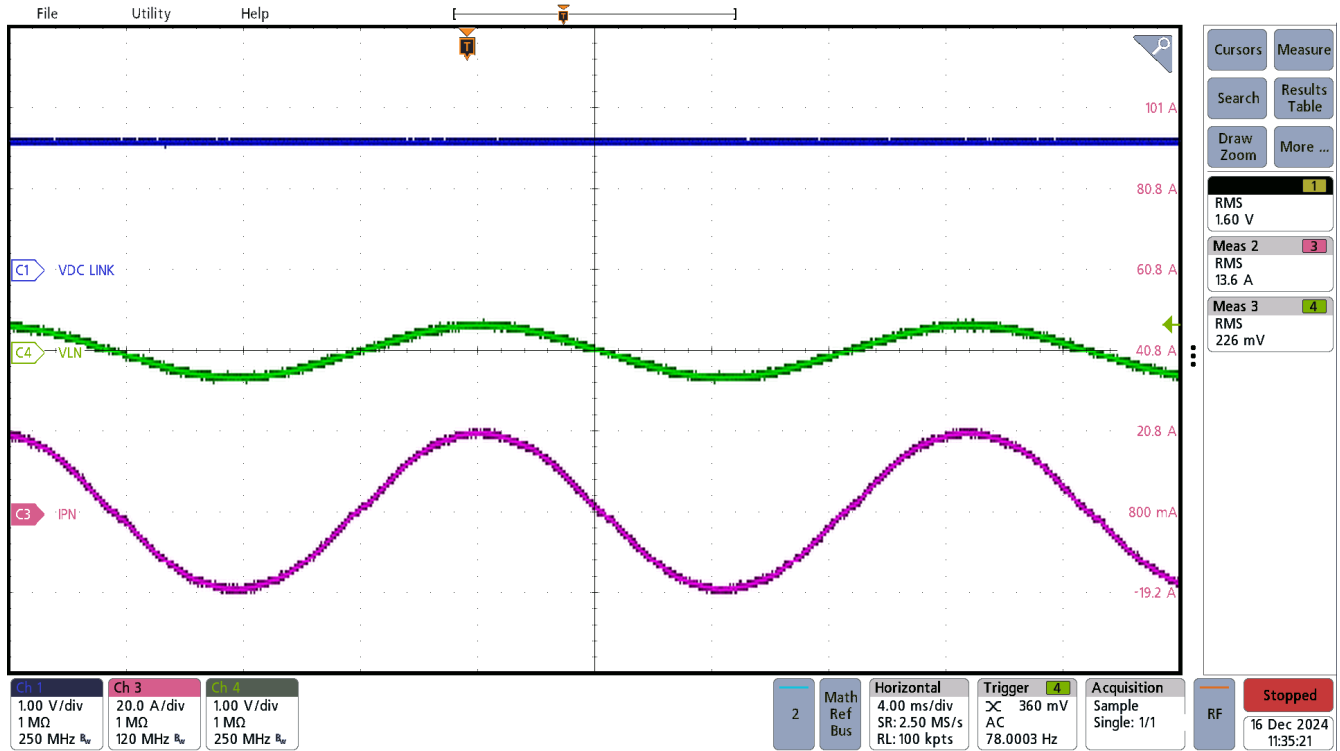
- 電子負荷の CV 電圧を 800V、20A 制限に設定します。前述の手順と同様に PFC 昇圧を開始します。

注

ラボ 7 では、PFC の電圧ループを紹介します。

このラボでは、出力電圧を調整する電子負荷を使用します。これらのテスト用に、TINV_VBUS_OVERVOLT_LIMIT を設定する必要があります。

- 120V_{RMS}、90kHz PWM で動作する PFC 閉電流ループの例を図 3-24 に示します。



スコープの信号:チャンネル 1 - DC 電圧 (青)、チャンネル 4 - AC 電圧 (緑)、チャンネル 3 - AC 電流 (赤)。電圧プローブは 500:1 に縮小されます。

図 3-24. 120V_{RMS} で動作する PFC 電流ループ

- 前述の手順を繰り返し、220V_{RMS} での PFC を確認します。
- 230V_{RMS}、90kHz PWM で動作する PFC 閉電流ループの例を図 3-25 に示します。

チャンネル 3 - AC 電流 (赤)。電圧プローブは 500:1 に縮小されます。

図 3-26. 230V_{RMS}、70kHz で動作する PFC 電流ループ

- スタートアップ時の過電流を確実に低減するため、ソフトスタート方式はスタートアップ時に実装されます。このため、デッドバンド設定を調整することにより、デューティは PWM モジュールに制限されます。デッドバンドを大きな値に設定し、徐々に公称値まで下げて電流スパイクを制限する、このセットアップの PWM 構成を図 3-27 に示します。

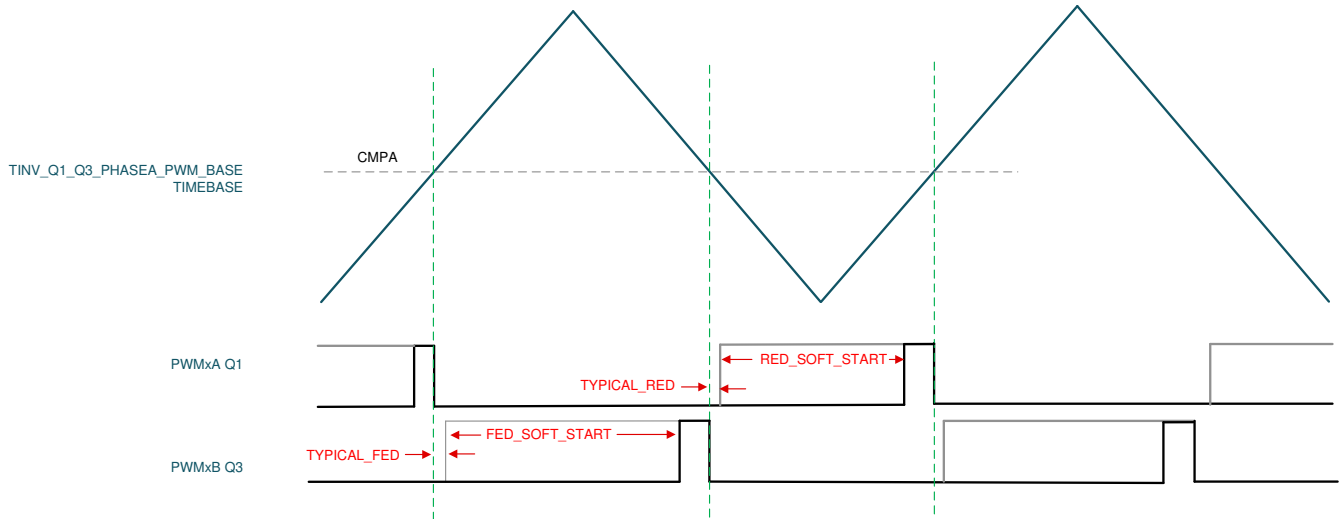
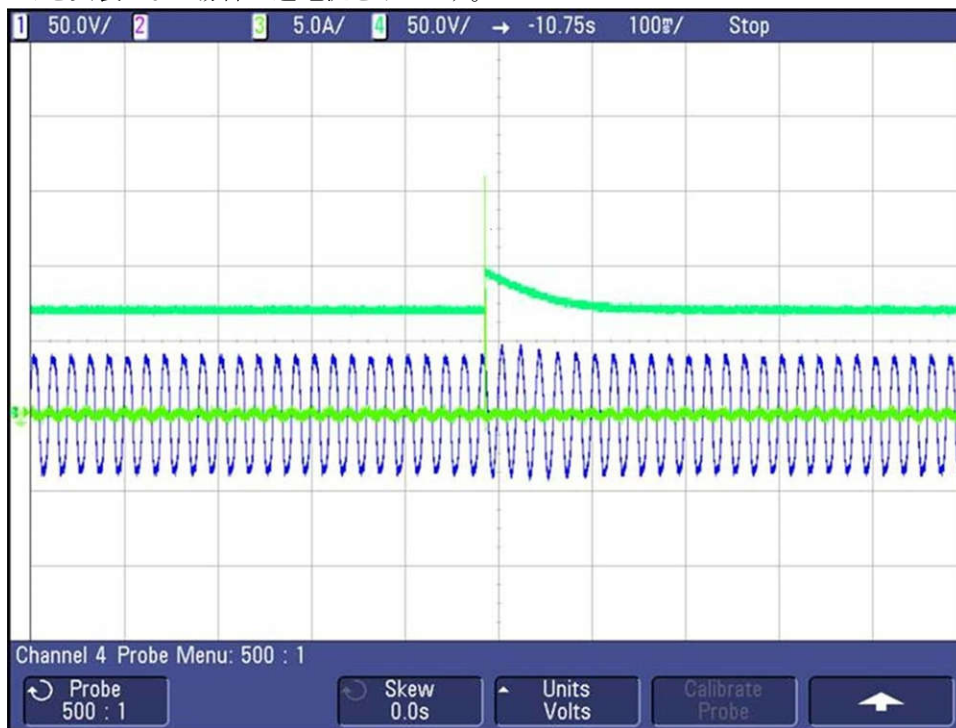


図 3-27. デッドバンドのソフトスタート PWM 構成

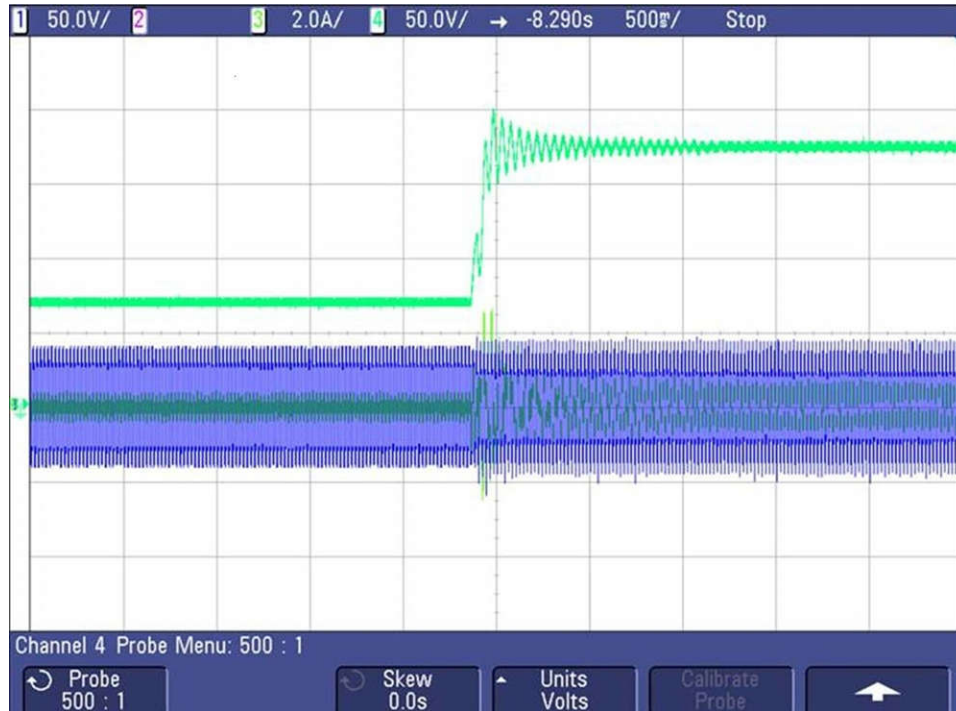
- 図 3-28 に、スタートアップ時にデッドバンドを実装しないスタートアップ方式の影響を示します。ソフトスタートを使用しないと、大きな電流スパイクが発生して、過電流トリップが発生し、DC バスの電圧崩壊も引き起こします。図 3-28 に、ソフトスタートを実装しない場合の過電流を示します。



スコープの信号: チャンネル 1 - AC 電圧 (青)、チャンネル 2 - AC 電流 (薄緑)、
チャンネル 3 - DC 電圧 (深緑)

図 3-28. ソフトスタートなし

図 3-29 に、ソフトスタートを実装することにより電流スパイクが低減するところを示します。



スコープの信号: チャンネル 1 - AC 電圧 (青)、チャンネル 2 - AC 電流 (薄緑)、
チャンネル 3 - DC 電圧 (深緑)。

図 3-29. 適応型デッドバンドを採用したソフトスタート

- このラボのソフトウェアには SFRA が統合されているため、ハードウェアを測定して、設計した補償器が十分なゲインマージンと位相マージンを提供していることを検証できます。SFRA を実行するには、プロジェクトを実行している状態で、cfg ページから SFRA アイコンをクリックします。SFRA GUI がポップアップ表示されます。
- SFRA GUI でデバイスのオプションを選択します。例として、F28377D の場合には浮動小数点を選択します。[Setup Connection] をクリックします。ポップアップ ウィンドウで [Boot on Connect] オプションのチェックを外し、適切な COM ポートを選択します。[OK] ボタンをクリックします。SFRA GUI に戻り、[Connect] をクリックします。
- SFRA GUI がデバイスに接続します。これで [Start Sweep] ボタンをクリックして、SFRA 掃引を開始できるようになりました。SFRA 掃引が完了するまでには数分かかります。SFRA GUI のプログレスバーを確認したり、UART の動作を示す制御カード裏面の青色 LED の点滅をチェックしたりすることで、動作を監視できます。完了すると、図 3-30 および図 3-31 に示すように、開ループプロットを含むグラフが表示されます。これらのグラフは、それぞれ SFRA GUI で測定されたプラント応答とループ応答を示します。これにより、設計した補償器が確かに安定していることを検証できます。

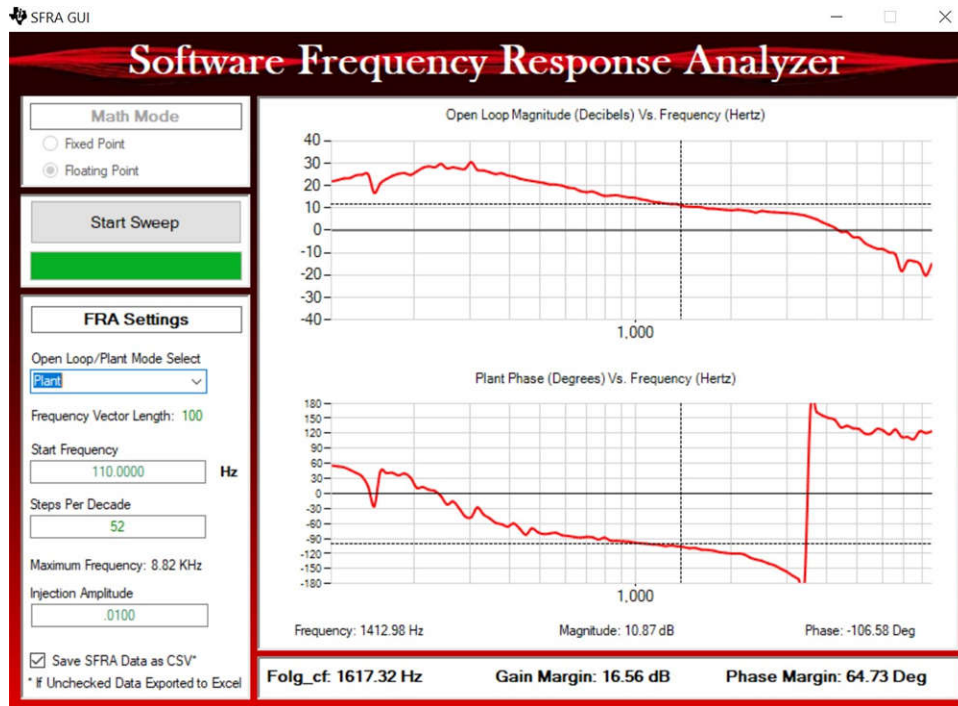


図 3-30. 電流ループに対する PFC SFRA プラントの応答

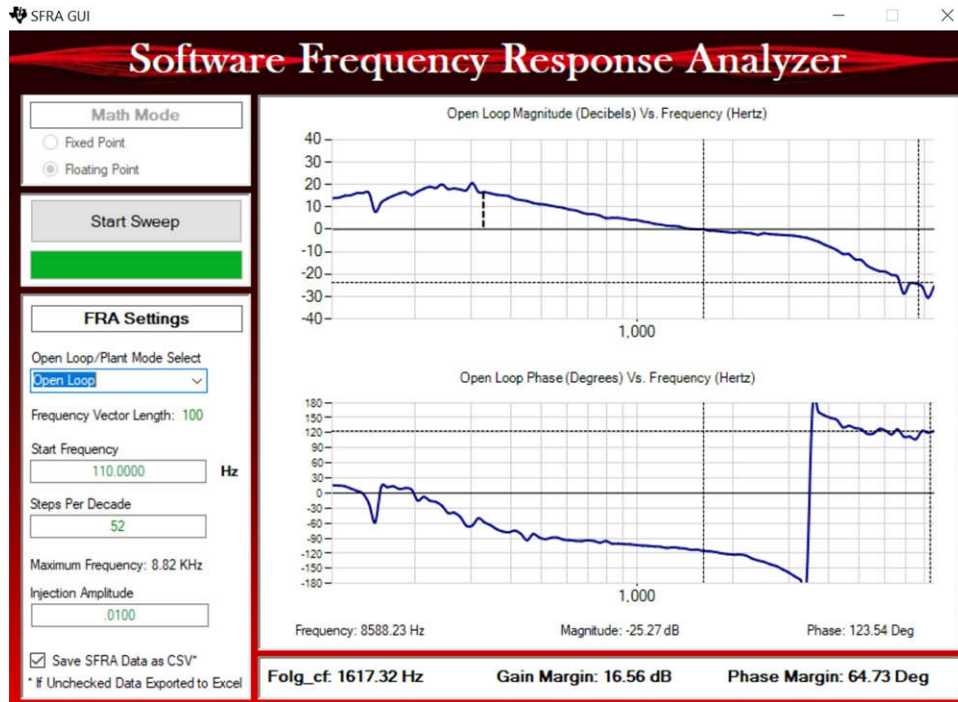


図 3-31. 電流ループに対する PFC SFRA ループの応答

- また、周波数応答データは SFRA データ フォルダ 下の プロジェクト フォルダ に保存され、SFRA 実行時のタイムスタンプが記録されます。
- これで、ラボ 6 の検証は完了です。

テストの構成に使用した機器の詳細については、[ハードウェア テスト設定](#)のセクションを参照してください。この時点では、高圧電力を基板に供給しないでください。

- まず `main.cfg` を起動し、プロジェクト オプションでラボ 7 を選択します。補償器スタイル (PI 補償器) と調整ループ (電流ループ) が自動的に設定されます。ここで **[Run Compensation Designer]** アイコンをクリックすると、**Compensation Designer** ツールが起動し、powerSUITE ページで指定したパラメータを持つ電流ループ プラント モデルが表示されます。
- [図 3-33](#) に、制御ループの実行に使用される電流補償係数を示します。これらの係数は、必要なループ帯域幅と位相マージンを満たすように変更できます。抵抗性負荷を使用する場合の理想的な係数は、送電網接続に使用する係数とはわずかに異なります。これは、送電網のインピーダンスが非常に小さいためです。[図 3-33](#) に、補償器設計の伝達関数と応答を示します。

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

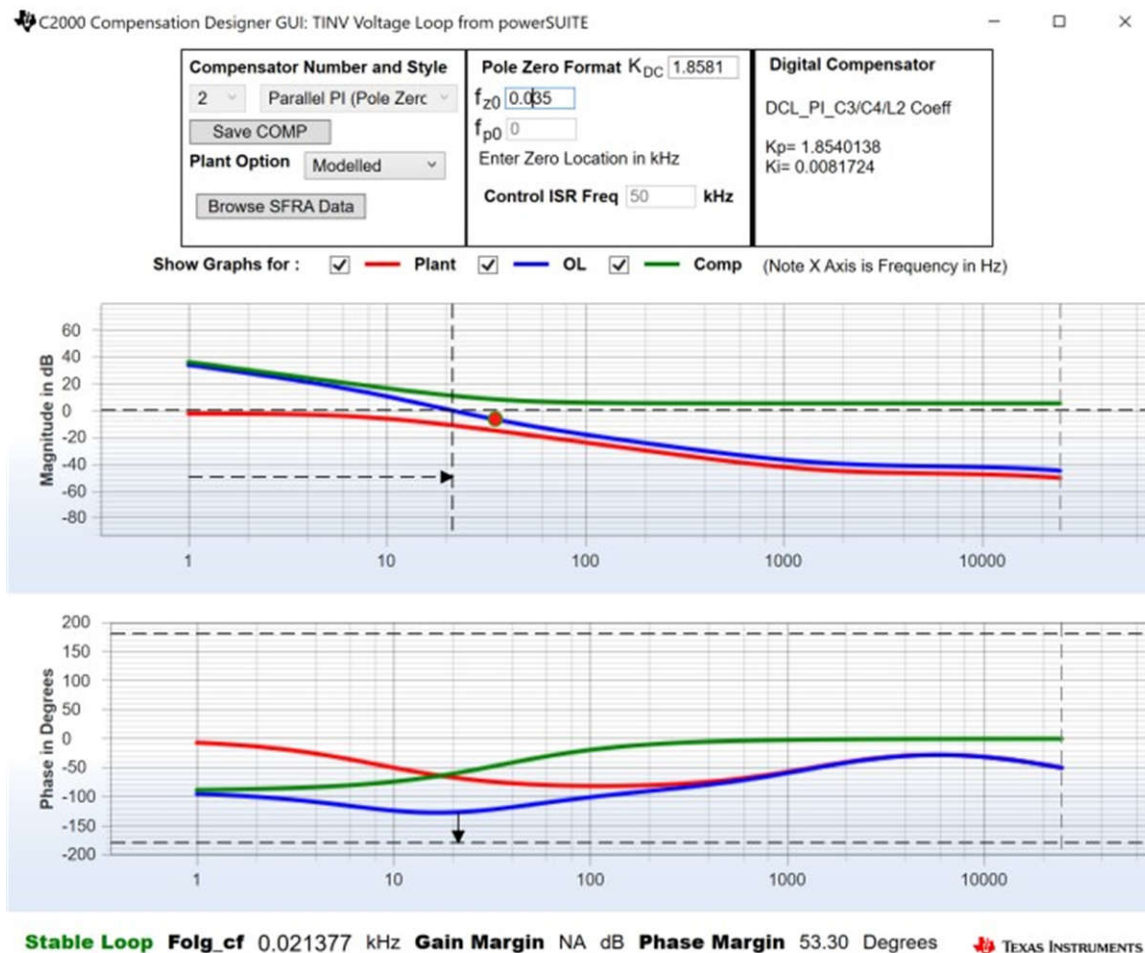


図 3-33. 補償器設計の GUI - 電圧ループの PI 係数

- 比例および積分利得値が条件を満たしているなら、**[Save COMP]** ボタンをクリックします。これにより、補償器の値がプロジェクトに保存されます。**Compensation Designer** を閉じて、powerSUITE ページに戻ります。
- 補助電源をオンにし、12V に設定します。コードをビルドしてロードし、`lab7.js` ファイルを使用して CCS ウィンドウに監視する変数を入力します。
- 電子負荷抵抗を $3\text{k}\Omega$ 付近の大きな値または CC モード 0.25A に設定します。
- 適切な電流制限を使用して、AC 入力電圧を 230V_{RMS} に設定します。
- AC 電源をオンにした後、すぐに `TINV_neutralRelaySet` に `1` を書き込んでリレーをオンにします。AC 電源をオンにした後すぐ (2 秒以内) にリレーがオンになるようにします。
- ここで、`TINV_vBusRef_pu` を 0.727pu に設定します。これは、 800V のバス電圧に相当します。

- 高電力テストの際には、デバッグ セッション中に CCS WATCH ウィンドウの TINV_fanSet 機能を使用して、ファンを有効にしてください。
- PFC モードを開始するには、TINV_startPowerStage 変数に「1」を入力します。正弦波信号の電流が送電網から流れ（ローパワーでの電流であるため、いくらかの高調波を含む）、vBus 上に昇圧が見られます。図 3-34 に示すように、出力電圧が 550V から約 800V に昇圧され、AC 電源から約 200W の電力が消費されます。この遷移は約 150ms で発生します。
- 負荷が増加すると、電流は正弦波になります。これにより、PFC が 230V_{RMS} で開始したことを検証できます（図 3-34 を参照）。
- 過電流トリップが検出されて PWM がオフになった場合は、ラボ 5 の注を参照してこの条件をデバッグしてください。
- コンバータの効率結果と過渡テストについては、「テスト結果」セクションを参照してください。
- このラボのソフトウェアには SFRA が統合されているため、ハードウェアを測定して、設計した補償器が十分なゲインマージンと位相マージンを提供していることを検証できます。SFRA を実行するには、プロジェクトを実行している状態で、.cfg ページから SFRA アイコンをクリックします。SFRA GUI がポップアップ表示されます。
- SFRA GUI でデバイスのオプションを選択します。例として、F28379D の場合には浮動小数点を選択します。[Setup Connection] をクリックします。ポップアップ ウィンドウで [Boot on Connect] オプションのチェックを外し、適切な COM ポートを選択します。[OK] ボタンをクリックします。SFRA GUI に戻り、[Connect] をクリックします。
- SFRA GUI がデバイスに接続します。これで [Start Sweep] ボタンをクリックして、SFRA 掃引を開始できるようになりました。SFRA 掃引が完了するまでには数分かかります。SFRA GUI のプログレス バーを確認したり、UART の動作を示す制御カード裏面の青色 LED の点滅をチェックしたりすることで、動作を監視できます。終了すると、図 3-34 のように開ループ プロットによるグラフが表示されます。図 3-34 に SFRA GUI で測定されたプラント応答を、図 3-35 に SFRA GUI で測定されたループ応答を示します。これにより、設計した補償器が確かに安定していることを検証できます。

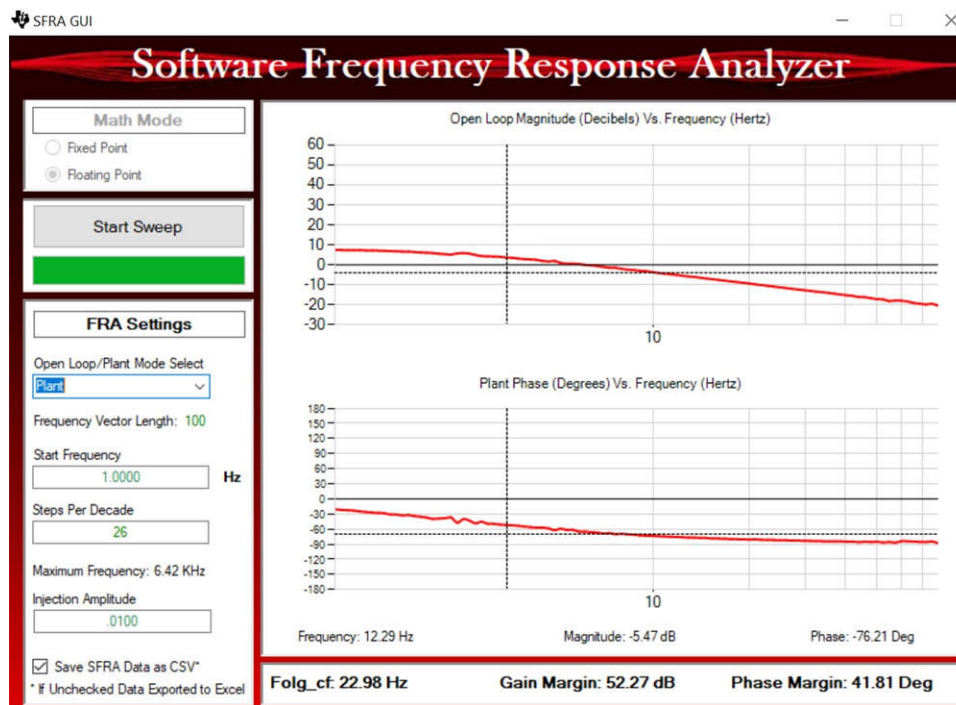


図 3-34. 電圧ループに対する PFC SFRA プラントの応答

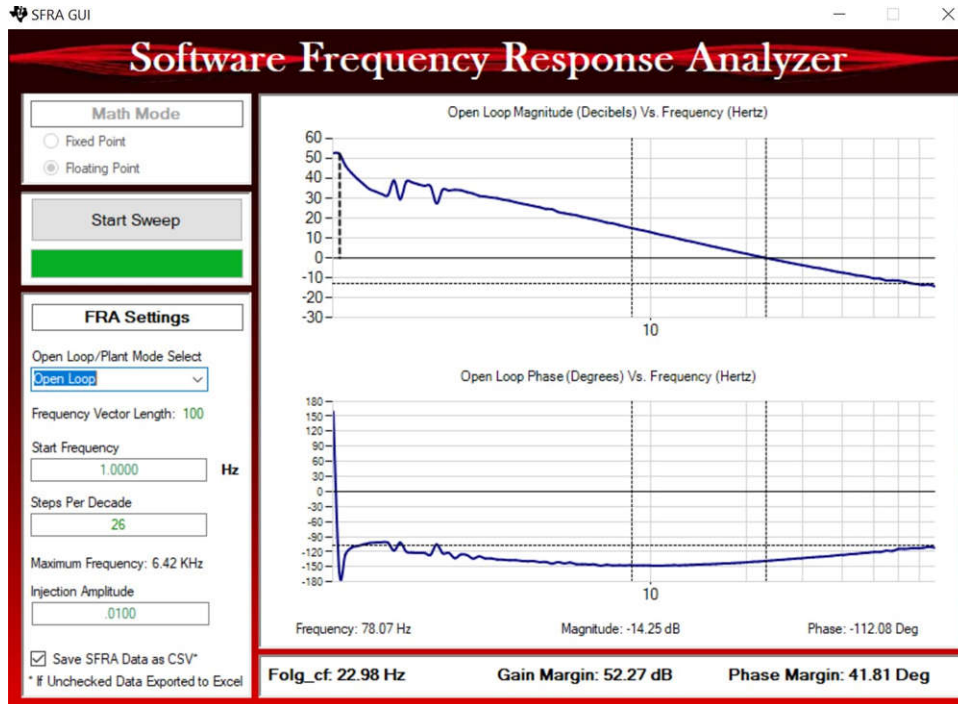


図 3-35. 電圧ループに対する PFC SFRA ループの応答

- また、周波数応答データは SFRA データフォルダ下のプロジェクトフォルダに保存され、SFRA 実行時のタイムスタンプが記録されます。また、測定されたゲイン マージンと位相マージンは、すでに説明した電圧ループ補償器の設計に示されているように、モデルの値に近くなります。
- この操作により、電圧ループ補償器設計を検証できます。システムを安全に停止させるには、入力 AC 電圧をゼロまで下げます。

3.2.4 効率に関するテストのセットアップ

図 3-36 に、このリファレンス デザインの効率をテストするために必要な機器を示します。

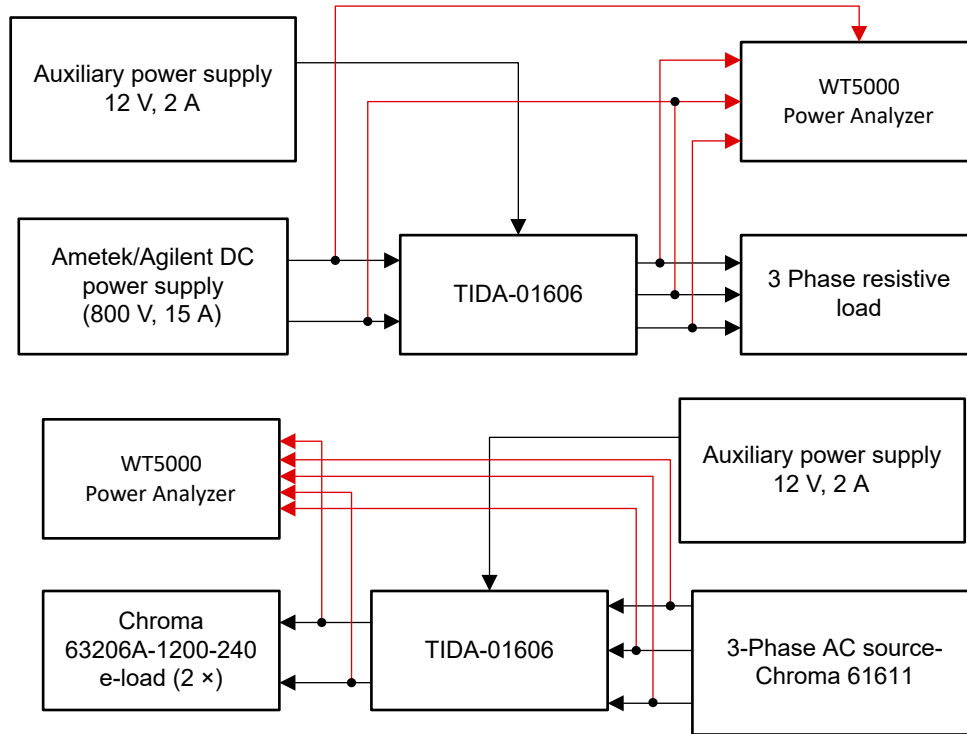


図 3-36. 効率に関するテストのセットアップ

- PFC モードのテスト用の Chroma 616xx シリーズ 3 相 AC 電源、インバータ モード (TIDA-01606) の動作をサポートする 800V の AMETEK、Agilent DC 電源
- 110kW の Simplex PowerStart (または任意の 3 相抵抗性) 負荷バンクを構成可能な負荷として使用し、さまざまな設定ポイントで設計をテスト
- Tektronix WT5000 高精度パワー アナライザ
- 外部 BK 高精度ベンチ電源を使用して、12V 入力 で DUT に電力を供給

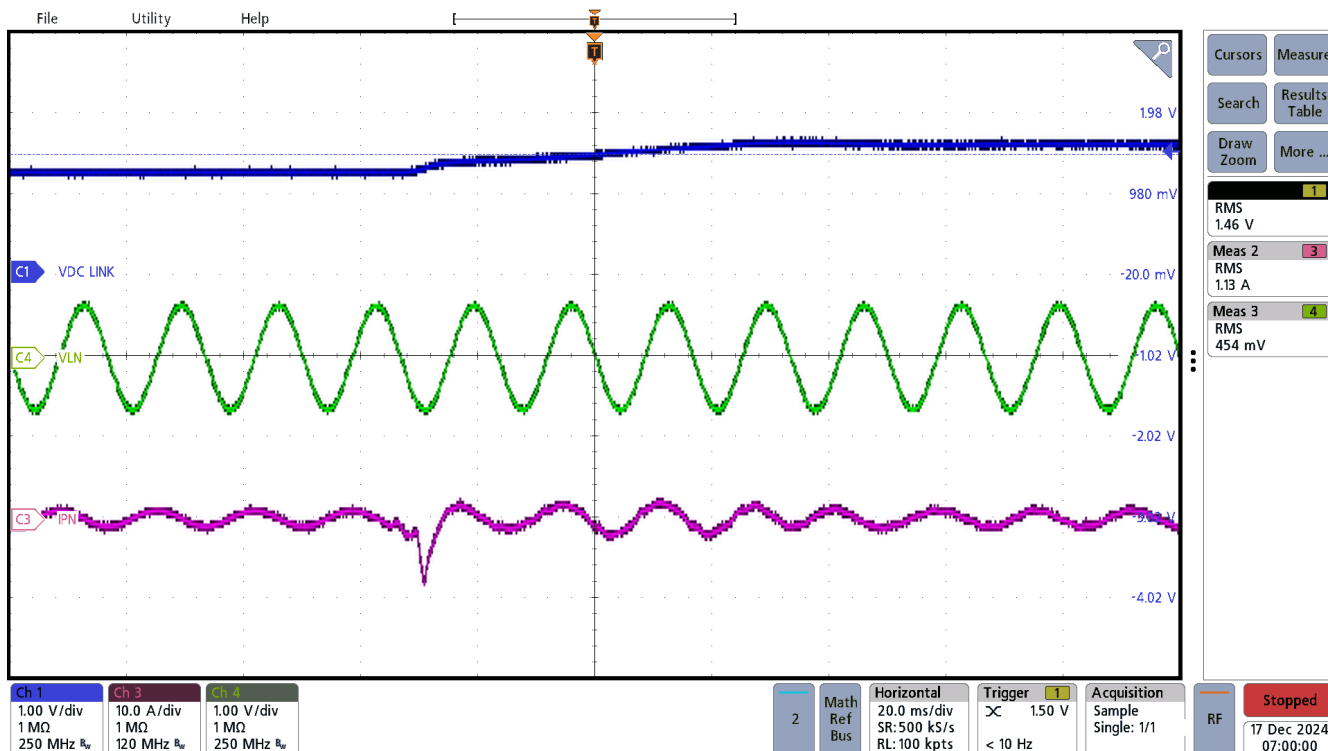
3.2.5 テスト結果

以下のセクションでは、インバータモードとPFCモードの動作結果について説明します。

3.2.5.1 PFC モード

3.2.5.1.1 PFC スタートアップ - 230V_{RMS}、400V_{L-L} AC 電圧

入力 3 相 400V_{L-L}、電圧が 800V に制御された出力バス、約 750W 負荷での電力段のスタートアップシーケンスを [図 3-37](#) に示します。550V から 800V への昇圧動作は、約 70ms で発生します。[図 3-37](#) に PFC のスタートアップ性能を示します。



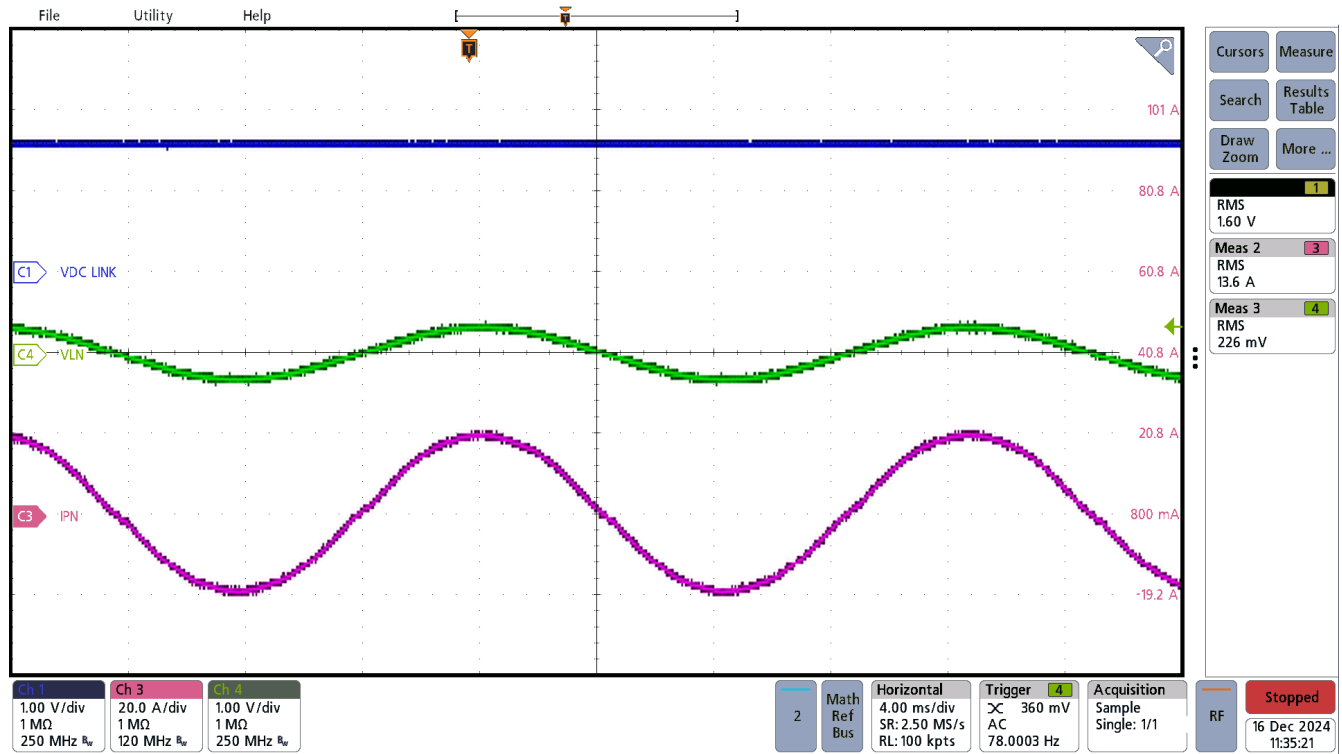
スコープの信号: チャンネル 1 - 出力時 VDC (青)、チャンネル 4 - AC 入力位相電圧 VLN (緑)、チャンネル 3 - AC 入力電流 (赤)。電圧プローブは 500:1 に縮小されます。

図 3-37. PFC 230V_{RMS} スタートアップ

スタートアップ時に DC リンク電圧を 550V から 800V に上昇させる際に過電圧 (800V 超) が発生しないように、電圧ループを調整するには注意が必要です。高負荷で PFC を起動すると過電流イベントが発生し、PWM がトリップする可能性があります。そのため、上記のテスト条件で PFC を起動するか、出力電力を下げてください。また、PWM トリップの考えられる原因と、この状況を回避するための注意事項については、[ラゴ 5](#) も参照してください。

3.2.5.1.2 定常状態の結果 - PFC モード

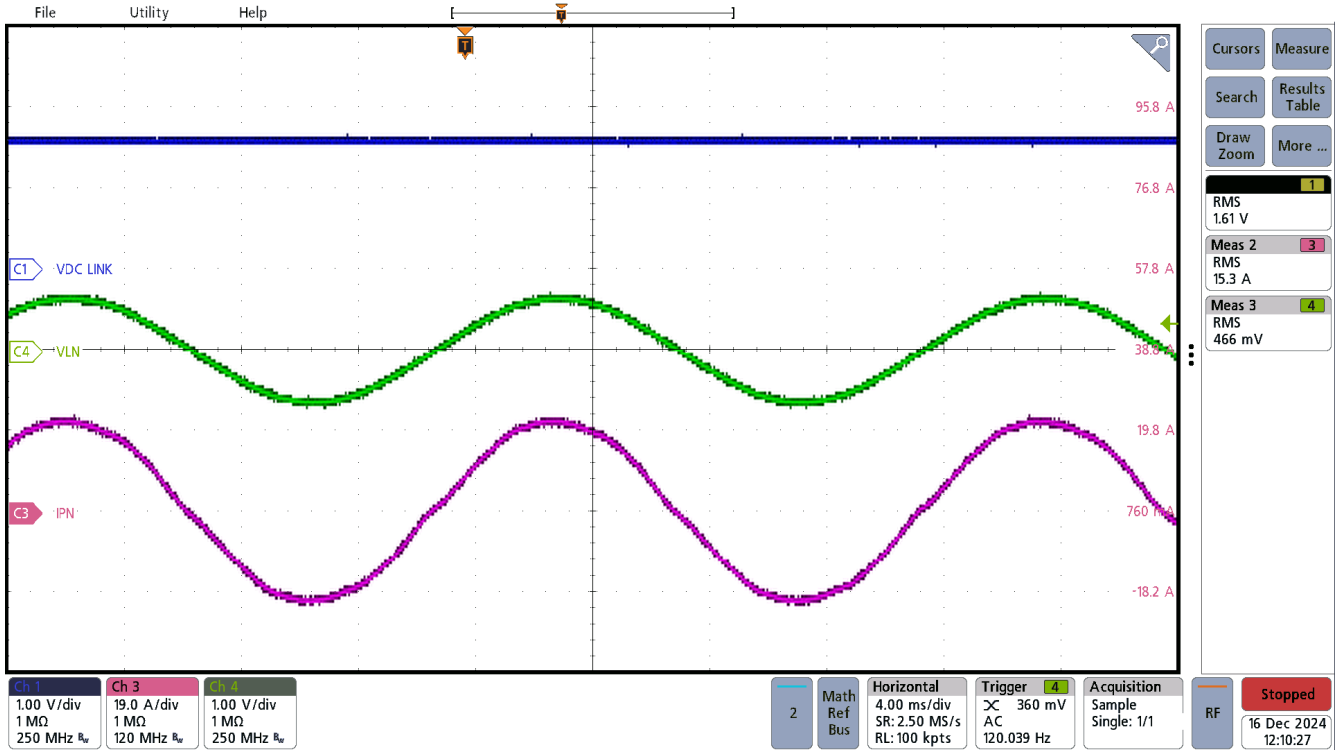
図 3-38 に、120 V_{RMS} および出力電力 4.5kW での PFC の定常状態性能を示します。



スコープの信号:チャンネル 1 - 出力時 VDC (青)、チャンネル 4 - AC 入力位相電圧 VLN (緑)、チャンネル 3 - AC 入力電流 (赤)。電圧プローブは 500:1 に縮小されます。

図 3-38. 定常状態での 120 V_{RMS} における PFC 結果 - 4.5kW

図 3-39 に、230 V_{RMS} 入力および出力電力 4.5kW での PFC の定常状態性能を示します。



スコープの信号:チャンネル 1 - 出力時 VDC (青)、チャンネル 4 - AC 入力位相電圧 VLN (緑)、チャンネル 3 - AC 入力電流 (赤)。電圧プローブは 500:1 に縮小されます。

図 3-39. 定常状態での 230 V_{RMS} における PFC 波形 - 4.5kW

3.2.5.1.3 効率、THD、力率の結果 (60Hz) – PFC モード

このセクションでは、120V_{RMS} および 230V_{RMS} で PFC モードで動作しているコンバータの効率、ITHD (電流全高調波歪み)、力率の結果について説明します。120V_{RMS} 入力については、400V と 800V の両方の出力について、90kHz PWM でテストを実施しました。230V_{RMS} 入力では、70kHz と 90kHz の両方の PWM を 800V 出力でテストを実施しました。

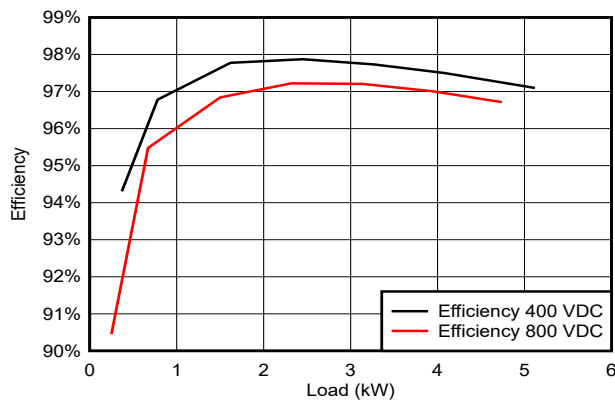


図 3-40. 効率の結果 - PFC モード、120V_{RMS}、90kHz

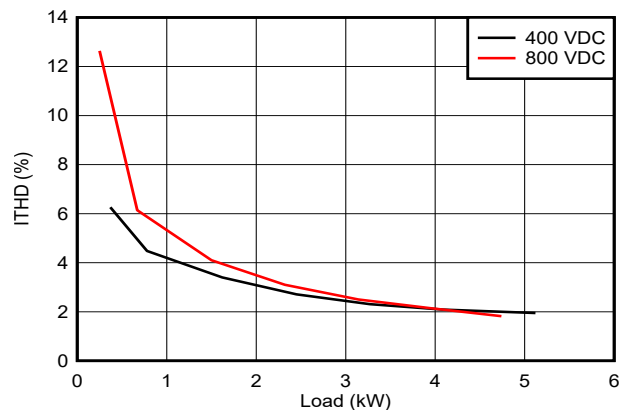


図 3-41. THD の結果 - PFC モード、120V_{RMS}、90kHz

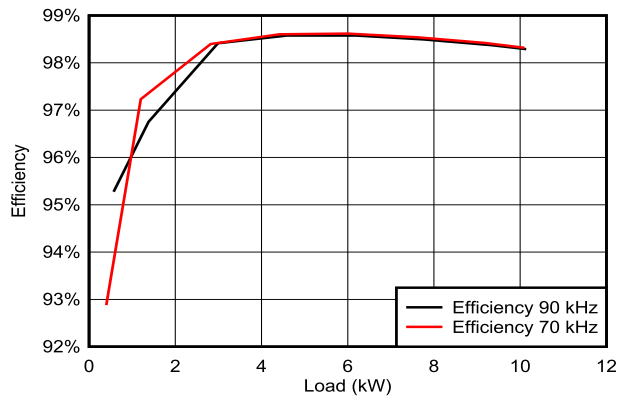


図 3-42. 効率の結果 - PFC モード、230V_{RMS}、800VDC

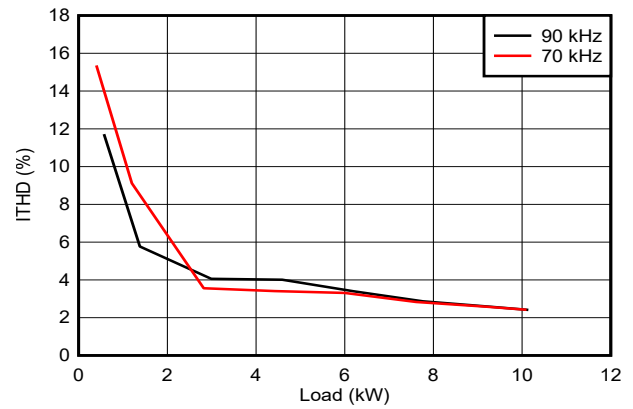


図 3-43. THD の結果 - PFC モード、230V_{RMS}、800VDC

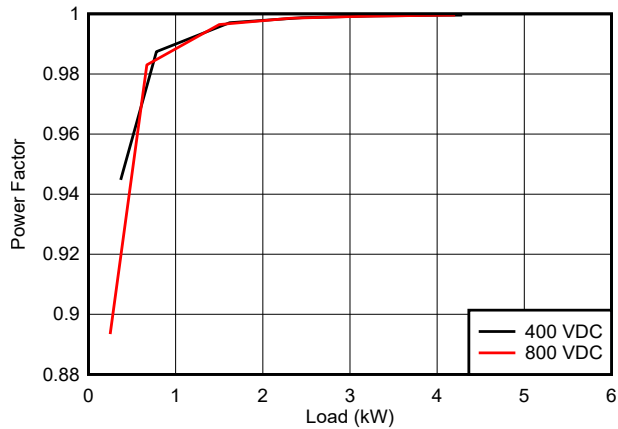


図 3-44. 力率の結果 - PFC モード、120V_{RMS}、90kHz

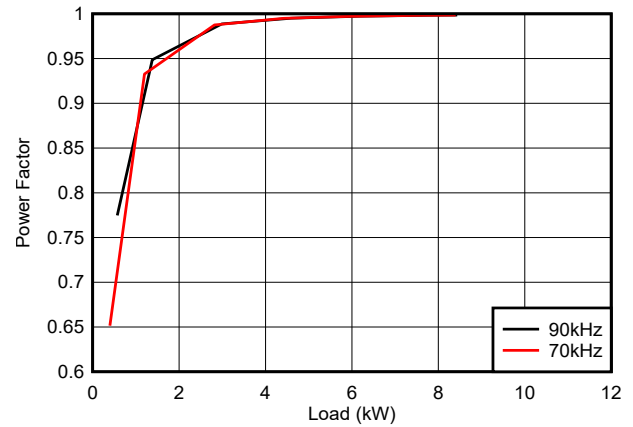
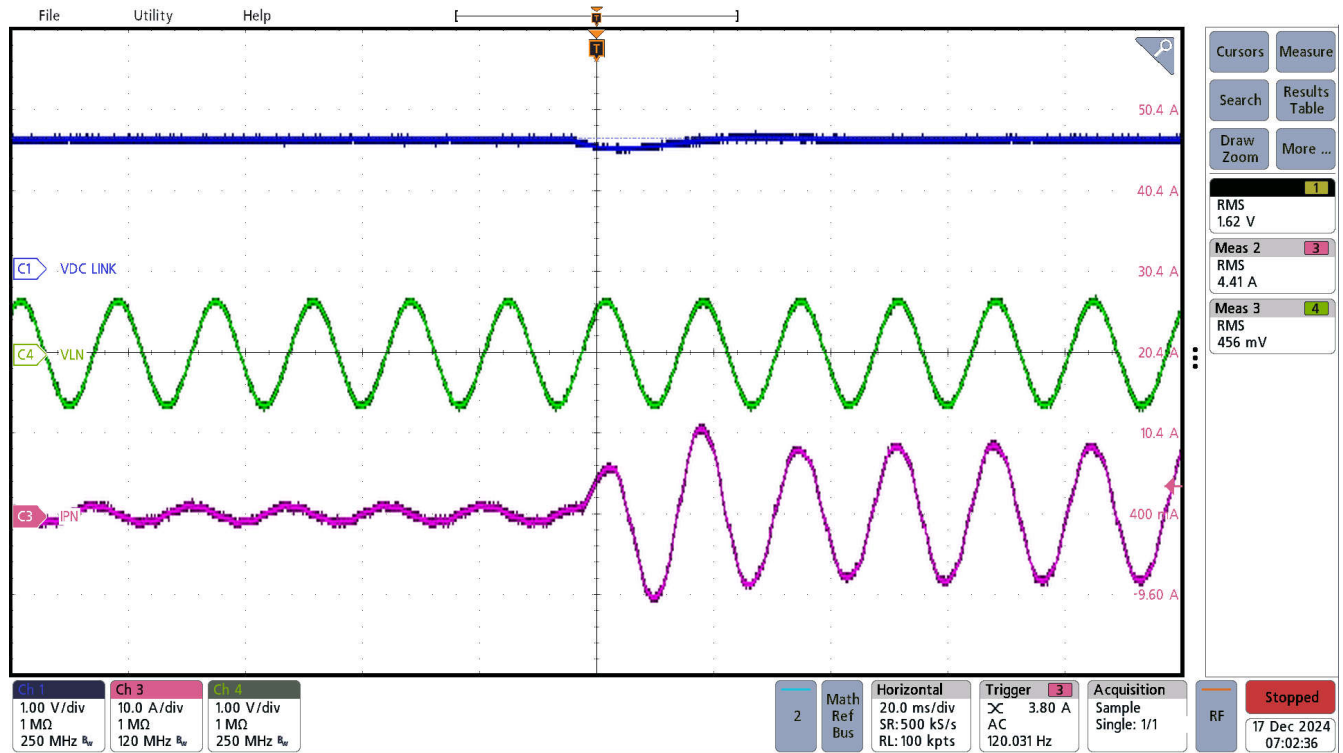


図 3-45. 力率の結果 - PFC モード、230V_{RMS}、800VDC

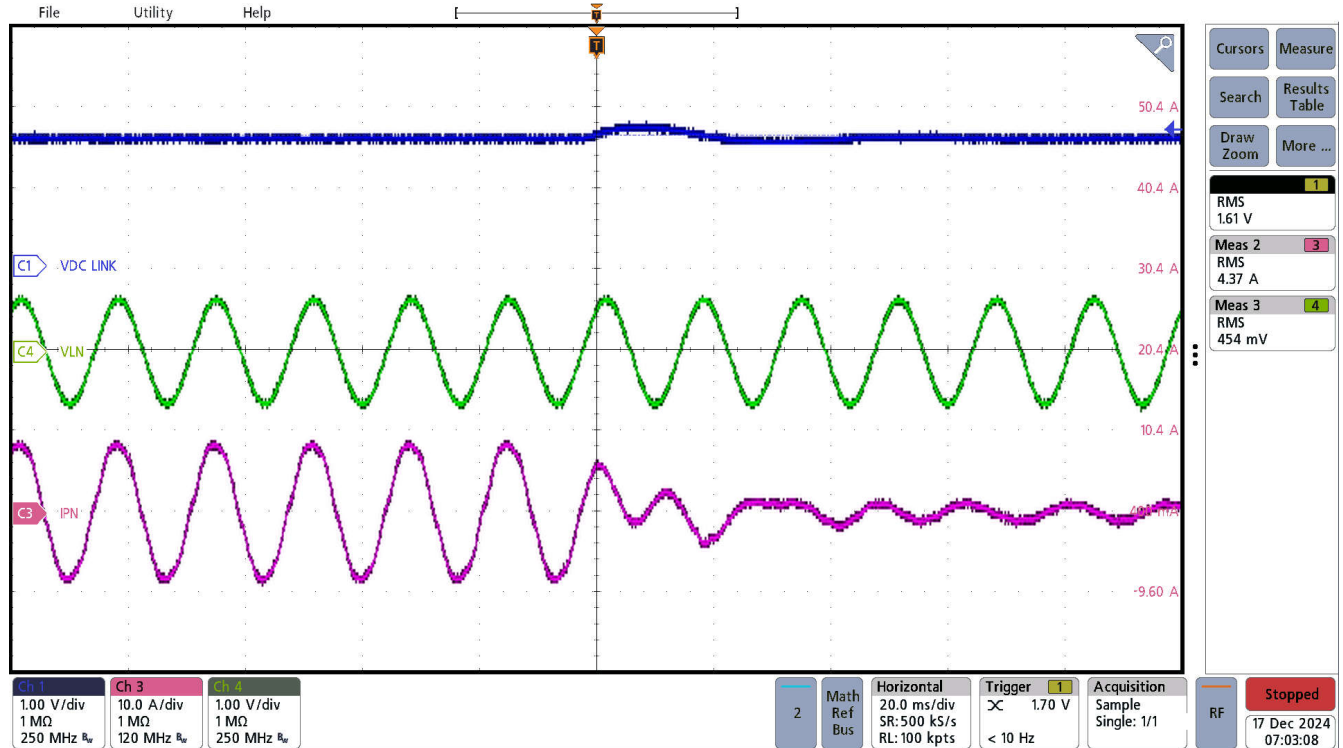
3.2.5.1.4 ステップ負荷変動による過渡応答テスト

このセクションでは、コンバータは最初、230V_{RMS} の AC 入力電圧、800V DC リンク電圧で動作し、約 0A を供給します。その後、コンバータには 5A ステップ負荷 (4kW) がかけられます。次の負荷ステップは 5A から 0A です。結果を図 3-46 と図 3-47 に示します。DC リンクで測定したピーク電圧リップルは、定常状態の約 10% でした。



スコープの信号:チャンネル 1 - 出力時 VDC (青)、チャンネル 4 - AC 入力位相電圧 VLN (緑)、
チャンネル 3 - AC 入力電流 (赤)。電圧プローブは 500:1 に縮小されます。

図 3-46. 負荷を増加させたときの過渡応答 - 0W → 4kW



スコープの信号:チャンネル 1 - 出力時 VDC (青)、チャンネル 4 - AC 入力位相電圧 VLN (緑)、

チャンネル 3 - AC 入力電流 (赤)。電圧プローブは 500:1 に縮小されます。

図 3-47. 負荷を減少させたときの過渡応答 - 4kW → 0W

3.2.5.2 インバータ モード

800V、940W での予備閉ループ インバータ モード テストでは、約 97.5% の効率が示されています。図 3-48 および図 3-49 に、インバータの熱画像を示します。

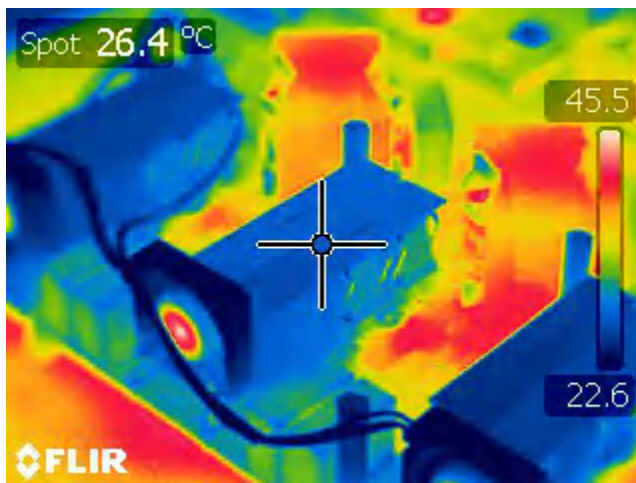


図 3-48. 基板底面の熱画像、900W で動作するインバータ

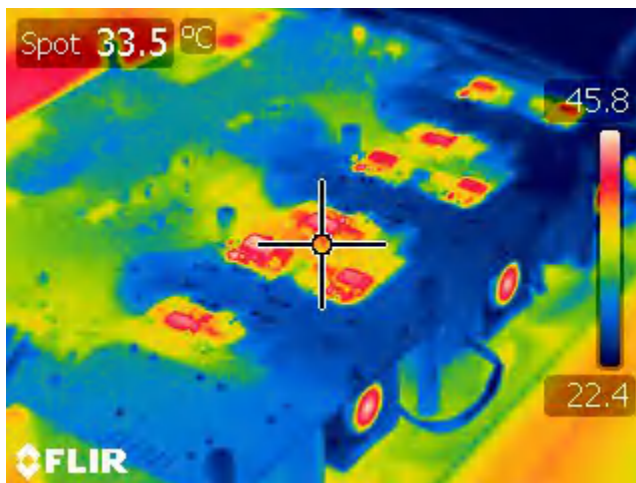


図 3-49. 基板上面の熱画像、900W で動作するインバータ

4 デザイン ファイル

4.1 回路図

回路図をダウンロードするには、[TIDA-01606](#) のデザイン ファイルを参照してください。

4.2 部品表 (BOM)

部品表 (BOM) をダウンロードするには、[TIDA-01606](#) のデザイン ファイルを参照してください。

4.3 PCB レイアウトに関する推奨事項

4.3.1 レイアウト プリント

レイヤ プロットをダウンロードするには、[TIDA-01606](#) のデザイン ファイルを参照してください。

4.4 Altium プロジェクト

Altium Designer® のプロジェクト ファイルをダウンロードするには、[TIDA-01606](#) のデザイン ファイルを参照してください。

4.5 ガーバー ファイル

ガーバー ファイルをダウンロードするには、[TIDA-01606](#) のデザイン ファイルを参照してください。

4.6 アセンブリの図面

アセンブリの図面をダウンロードするには、[TIDA-01606](#) のデザイン ファイルを参照してください。

5 商標

テキサス・インスツルメンツの™, Delfino™, TMS320C2000™, and C2000™ are trademarks of Texas Instruments.

Microsoft® and Windows® are registered trademarks of Microsoft Corporation.

Altium Designer® is a registered trademark of Altium LLC or its affiliated companies.

すべての商標は、それぞれの所有者に帰属します。

6 著者について

RUFFO RICCARDO は、2019 年にイタリア、トリノにあるトリノ工科大学で電気、電子、通信工学の博士号を取得しました。現在は、ドイツの テキサス・インスツルメンツで、送電インフラと再生可能エネルギーの分野でシステム エンジニアとして勤務しています。主な業務は、EV 充電、誘導性ワイヤレス電力伝送、太陽光発電、再生可能エネルギー、エネルギー貯蔵アプリケーションです。

KELVIN LE は、テキサス インスツルメンツのシステム エンジニアで、送電網部門で EV 充電に焦点を置いたシステム設計の開発を担当しています。Kelvin は 2015 年から テキサス・インスツルメンツに勤務しています。セントラル オクラホマ 大学で生体医工学の理学士号を、テキサス大学オースティン校で電気およびコンピュータ工学の理学修士号を取得しています。

筆者たちは、このリファレンス デザインに関してサポートを提供してくれた **MANISH BHARDWAJ**、**BART BASILE**、**HARISH RAMAKRISHNAN**、**MURALI KRISHNA PACHIPULUSU** に感謝の意を表しています。

7 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。

Changes from Revision I (July 2023) to Revision J (February 2025)	Page
• ドキュメント全体を通してハードウェア リビジョン E7 に関して更新しました。.....	1

重要なお知らせと免責事項

テキサス・インスツルメンツは、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、テキサス・インスツルメンツ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 テキサス・インスツルメンツ製品の選定、(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 テキサス・インスツルメンツ製品を使用するアプリケーションの開発の目的でのみ、テキサス・インスツルメンツ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。テキサス・インスツルメンツ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、テキサス・インスツルメンツおよびその代理人を完全に補償するものとし、テキサス・インスツルメンツは一切の責任を拒否します。

テキサス・インスツルメンツの製品は、[テキサス・インスツルメンツの販売条件](#)、または [ti.com](https://www.ti.com) やかかる テキサス・インスツルメンツ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。テキサス・インスツルメンツがこれらのリソースを提供することは、適用されるテキサス・インスツルメンツ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、テキサス・インスツルメンツ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
Copyright © 2025, Texas Instruments Incorporated

重要なお知らせと免責事項

テキサス・インスツルメンツは、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、テキサス・インスツルメンツ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テキサス・インスツルメンツ製品の選定、(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テキサス・インスツルメンツ製品を使用するアプリケーションの開発の目的でのみ、テキサス・インスツルメンツ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。テキサス・インスツルメンツ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、テキサス・インスツルメンツおよびその代理人を完全に補償するものとし、テキサス・インスツルメンツは一切の責任を拒否します。

テキサス・インスツルメンツの製品は、[テキサス・インスツルメンツの販売条件](#)、または [ti.com](https://www.ti.com) やかかるテキサス・インスツルメンツ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。テキサス・インスツルメンツがこれらのリソースを提供することは、適用されるテキサス・インスツルメンツ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、テキサス・インスツルメンツ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
Copyright © 2025, Texas Instruments Incorporated