

EVM User's Guide: TIEVM-MTR-HVINV

TIEVM-MTR-HVINV 750W 高压电机逆变器评估模块



说明

TIEVM-MTR-HVINV 是一款 750W 开发板，适用于高压电机驱动应用。该 EVM 使用 InstaSPIN-FOC FAST™ 和 eSMO 无传感器观测器，实现了三相 PMSM 的无传感器 FOC 控制。该模块化设计允许通过即插即用方式将不同子板连接到同一主板。该 EVM 的硬件和固件经过测试，可供随时使用，有助于缩短开发时间。本用户指南中提供了设计详细信息和测试结果。

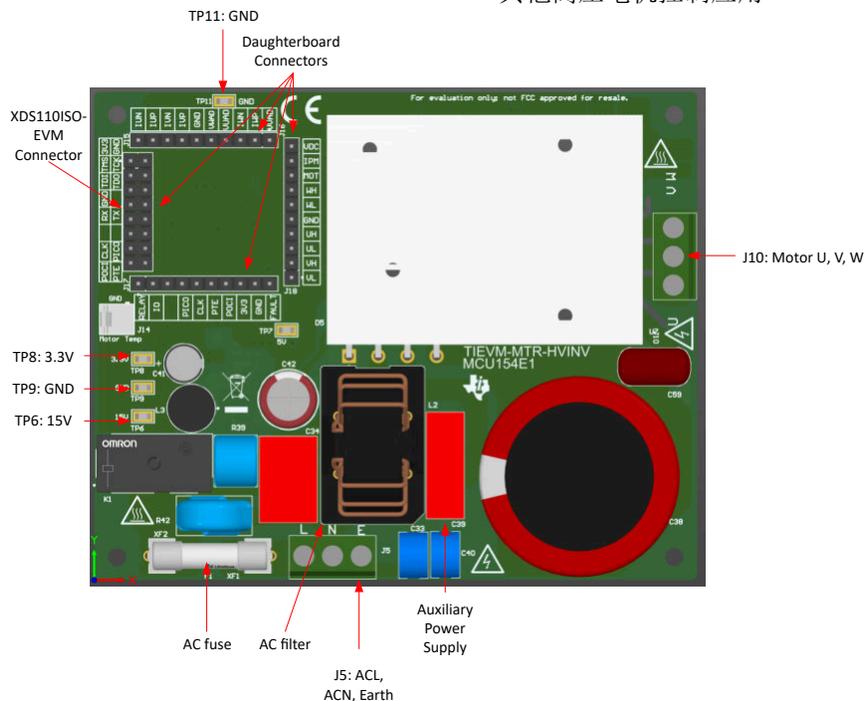
特性

- 宽电压输入工作范围：165VAC 至 265VAC (50Hz/60Hz 时)，或高达 400VDC

- 具有高达 750W 的逆变器级、15kHz 开关频率、扭矩补偿和自动弱磁控制 (FWC)
- 模块化设计支持在同一主板上使用不同 MCU 子板
- 无传感器磁场定向控制 (FOC) 电机控制，支持 FAST™ 和 eSMO 两种观测器
- 用户友好型图形用户界面 (GUI) 可方便进行无代码开发，并控制、识别和监控电机

应用

- 洗衣机和烘干机
- 空调室内机
- 冰箱和冷冻柜
- 电器：压缩机
- 其他高压电机控制应用



TIEVM-MTR-HVINV EVM 电路板布局布线

1 评估模块概述

1.1 简介

当今大型家电或类似应用所采用的电机控制技术必须满足日益增长的对于更低成本、更小尺寸、更高功率和更高能效的需求。此外，永磁同步电机 (PMSM) 在大型家电应用中越来越常用。

TIEVM-MTR-HVINV 提供了 750W 逆变器主板和带 MCU 的控制子板，如 TIEVM-MC-F280013x，便于用户在高压环境中评估 C2000 系列微控制器。

适用于 TIEVM-MTR-HVINV 的软件支持 FAST™ 和 eSMO 两种无传感器观测器，因此可以比较这两种算法的性能。用户友好型 GUI 还有助于识别电机并进行控制参数调优，从而加快开发速度。

1.2 套件内容

TIEVM-MTR-HVINV 开发套件包含以下几项：

- TIEVM-MTR-HVINV 电源板/主板
- TIEVM-MC-MODULE-F280013x 控制板/子板
- 4 个螺柱和 4 颗螺钉，将安装在 TIEVM-MTR-HVINV 上的每个角中

虽然 TIEVM-MTR-HVINV 开发套件中不包含以下几项，但使用 EVM 时需要它们：

- XDS110ISO-EVM 插件隔离仿真器板
- USB Type-A 转 USB Type-C® 电缆或 USB Type-C® 转 USB Type-C® 电缆，用于将 PC 连接到 XDS110ISO-EVM
- 1 个 HV 电机
 - 支持多种电机，包括完全定制的电机（通过 user_mtr1.h 文件和/或 MCU-MOTOR-CONTROL-GUI 定制电机接口进行定制）。
 - Estun EMJ-04APB22 电机规格实现已知非常适合此系统，并且在 MCU-MOTOR-CONTRL-GUI 接口中提供一个集成选项。

作为 TIEVM-MTR-HVINV 电源板/主板的一部分，已提供一根 8A 250V 保险丝。

如果需要更换保险丝，请参阅电路板 BOM，了解保险丝详细信息和额定值（列为元件 F1）。确保所有主板和子板元件已安全断电。轻轻拆下 XF1 和 XF2 固定的现有保险丝。更换新保险丝。

1.3 规格

表 1-1 中列出了 TIEVM-MTR-HVINV 规格。

表 1-1. 主要系统规格

参数	测试条件	最小值	标称值	最大值	单位
系统输入/外部电源输出特性					
输入电压 (V_{INAC})	-	165	230	265	VAC
输入频率 (f_{LINE})	-	47	50	63	Hz
空载待机功耗 (P_{NL})	$V_{INAC} = 230V, I_{out} = 0A$	-	3.0	-	W
输入电流 (I_{IN})	$V_{INAC} = 230V, I_{out} = I_{MAX}$	-	8	-	A
电机逆变器特性					
PWM 开关频率 (f_{sw})	-	-	15	20	kHz
额定输出功率 (P_{OUT})	$V_{INAC} =$ 标称值	-	500	750	W
输出电流 (I_{RMS}) ¹	$V_{INAC} =$ 标称值	-	3	-	A
逆变器效率 (η)	$V_{INAC} =$ 标称值, $P_{OUT} =$ 标称值	-	98	-	%
电机频率 (f)	$V_{INAC} =$ 最小值至最大值	20	200	400	Hz

¹ 参考输出功率和具体电机特性，以确定输出电流。

表 1-1. 主要系统规格 (续)

参数	测试条件	最小值	标称值	最大值	单位
故障保护	过流、恢复失速、欠压、过压				
驱动控制方法和功能	采用三个或单个用于电流检测的分流电阻器的无传感器 FOC				

表 1-1. 主要系统规格 (续)

参数	测试条件	最小值	标称值	最大值	单位
	系统特性				
内置辅助电源	V_{INAC} = 最小值至最大值	15V \pm 10%、200mA, 3.3V \pm 10%、300mA			
工作环境温度	开放式框架	-10	25	55	$^{\circ}\text{C}$
电路板尺寸	长 \times 宽 \times 高	105mm \times 85mm \times 60mm			mm^3

TI 期望该 EVM 仅在实验室环境中操作，**不应**将该器件作为成品用于一般消费用途。

TI 期望该 EVM 仅由合格工程师和技术员使用，且他们应熟悉与处理高电压电子和机械元件、系统及子系统相关的风险。

高电压！如电路板的电压和电流处理不当或施加不正确，则可能导致电击、火灾或伤害事故。将电路板连接到火线时可能会触电。电路板必须由专业人员小心处理。

为安全起见，**必须**使用具有过压/过流保护的**隔离式测试设备**。如果可能，应在符合高压测试规定的外壳中进行测试。

警告

表面高温！接触会导致烫伤。**请勿触摸！**电路板上电后，某些元件可能会达到 55°C 以上的高温。由于存在高温，在使用过程中或使用刚结束时，用户不得触摸电路板。

小心

请勿在无人照看的情况下使该 EVM 通电。

备注

TI 建议使用符合适用地区安全标准 (例如, UL、CSA、VDE、CCC 和 PSE 等) 的外部电源或电源配件。

1.4 器件信息

- TMS320F280013x** 是 C2000™ 可扩展、超低延迟实时微控制器器件系列中的一款器件，专为提高电力电子应用的效率而设计。实时控制子系统基于 TI 的 32 位 C28x 数字信号处理器 (DSP) 内核，可针对从片上闪存或 SRAM 运行的浮点或定点代码提供 120MHz 的信号处理性能。三角函数加速器 (TMU) 和循环冗余校验 (VCRC) 扩展指令集进一步增强了 C28x CPU 的性能，从而加快了实时控制系统关键常用算法的速度。高性能模拟块集成在 F280013x 实时微控制器 (MCU) 上，并与处理单元和 PWM 单元紧密耦合，从而提供出色的实时信号链性能。14 个 PWM 通道均支持与频率无关的分辨率模式，可控制从三相逆变器到高级多级电源拓扑的各种功率级。连接可以通过各种业界通用通信端口 (如 SPI、三个 SCI|URAT、I2C 和 CAN) 进行，另外还提供了多个引脚多路复用选项，可实现出色的信号布局。
- TLV740P** 低压降 (LDO) 线性稳压器是一款低静态电流 LDO，具有出色的线路和负载瞬态性能，专为对功耗敏感的应用设计。此器件可提供 1% 的典型精度。

TLV740P 还可在器件上电和使能期间提供浪涌电流控制。TLV740P 将输入电流限制为定义的电流限值，从而防止从输入电源流出的电流过大。此功能对于电池供电类器件尤为重要。

- TLV9062** 是具有轨到轨输入和输出摆动功能的双路低压 (1.8V 至 5.5V) 运算放大器。该器件是非常具有成本效益的设计，适用于需要低电压运行、小型封装尺寸和高容性负载驱动能力的应用。虽然 TLV906x 的容性负载驱动能力为 100pF，但电阻式开环输出阻抗便于在更高的容性负载下更轻松地实现稳定。TLV906xS 器件具有关断模式，允许放大器切换至典型电流消耗低于 $1\mu\text{A}$ 的待机模式。TLV906xS 系列有助于简化系统设计，因为该系列具有稳定的单位增益，集成了 RFI 和 EMI 抑制滤波器，而且在过驱条件下不会出现相位反转。

- **TMP61x** 线性热敏电阻可在整个温度范围内提供线性度和始终如一的灵敏度，支持使用简单而准确的方法进行温度转换。该器件的低功耗和较小的热质量可充分减小自发热的影

响。这些器件具有内置的高温失效防护性能以及对环境变化的强大抵抗力，设计用于长寿命的高性能应用。TMP6 系列器件外型小巧，可靠近热源放置，并具有快速响应时间。

- **TPS54202** 是一款输入电压范围为 4.5V 至 28V 的 2A 同步降压转换器。该器件包含两个集成式开关场效应晶体管 (FET) 并且具备内部回路补偿和 5ms 内部软启动功能，可降低组件数。

高级 Eco-mode 实现可尽可能提高轻负载效率并降低功率损耗。

两个高侧 MOSFET 内的逐周期电流限制可在过载情况下保护转换器，并通过低侧 MOSFET 续流电流限制防止电流失控，增强限制效果。

- **UCC2881** 在单片器件中集成了控制器和 14 Ω、700V 功率 MOSFET。该器件还集成了高压电流源，能够在经整流的市电电压下直接启动和运行。UCC2881 与 UCC2880 属于同一系列的器件，但电流更高。

该器件的静态电流较低，能够提供出色的效率。凭借 UCC2881，使用很少的外部元件即可构建降压、降压/升压以及反激拓扑等常用的转换器拓扑。

通用德州仪器 (TI) 高压评估 (TI HV EVM) 用户安全指南



务必遵循 TI 的设置和应用说明，包括在建议的电气额定电压和功率限制范围内使用所有接口元件。务必采取电气安全防护措施，这样有助于确保自身和周围人员的人身安全。如需了解更多信息，请联系 TI 的产品信息中心，网址为 <http://ti.com/customer support>。

保存所有警告和说明以供将来参考。

警告

务必遵循警告和说明，否则可能引发电击和灼伤危险，进而造成财产损失或人员伤亡。

TI HV EVM 一词是指通常以开放式框架、敞开式印刷电路板装配形式提供的电子器件。该器件严格用于开发实验室环境，仅供了解开发和应用高压电路相关电气安全风险且接受过专门培训、具有专业知识背景的合格专业用户使用。德州仪器 (TI) 严禁任何其他不合规的使用和/或应用。如果不满足合格要求，应立即停止进一步使用 HV EVM。

1. 工作区安全：

- a. 保持工作区整洁有序。
- b. 每次电路通电时，都必须由具有资质的观察员在场监督。
- c. TI HV EVM 及接口电子元件通电区域必须设有有效的防护栏和标识；指示可能存在高压操作，以避免意外接触。
- d. 开发环境中使用的所有接口电路、电源、评估模块、仪器、仪表、示波器和其他相关装置如果超过 50Vrms/75VDC，则必须置于紧急断电 EPO 保护电源板内。
- e. 使用稳定且不导电的工作台。
- f. 使用充分绝缘的夹钳和导线来连接测量探针和仪器。尽量不要徒手进行测试。

2. 电气安全：

- a. 作为一项预防措施，假定整个 EVM 可能存在用户可完全接触到的高电压是一种好的工程做法。
- b. 执行任何电气测量或其他诊断测量之前，需切断 TI HV EVM 及其全部输入、输出和电气负载的电源。再次确认 TI HV EVM 已安全断电。
- c. 确认 EVM 断电后，根据所需的电路配置、接线、测量设备连接和其他应用需求执行进一步操作，同时仍假定 EVM 电路和测量仪器均带电。
- d. EVM 准备就绪后，根据需要 will EVM 通电。

警告

EVM 通电后，请勿触摸 EVM 或电路，因为 EVM 或电路可能存在高压，会造成电击危险。

3. 人身安全

- a. 穿戴个人防护装备（例如乳胶手套或具有侧护板的安全眼镜）或将 EVM 放置于带有联锁装置的透明塑料箱，避免意外接触。

安全使用限制条件：

勿将 EVM 作为整体或部分生产单元使用。

2 硬件

2.1 硬件说明

TIEVM-MTR-HVINV 与控制板 (如 TIEVM-MC-MODULE-F280013x) 配合使用时, 可实现完整的电机驱动系统。

图 2-1 展示了该系统的概貌。

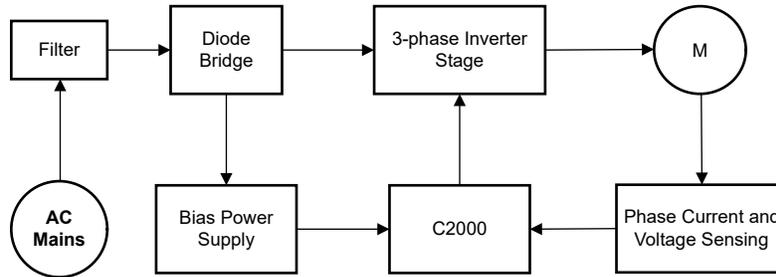


图 2-1. TIEVM-MTR-HVINV 的硬件板方框图

以下是 TIEVM-MTR-HVINV 的主要功能列表, 为方便参考, 这些功能组织成了不同的块。

- 电源线 220VAC 输入滤波器和二极管桥整流器
- 辅助电源
 - 整流交流输入产生 +15V 和 +3.3V 电压。
- 三相逆变器
 - 高达 750W 三相逆变器
 - 15kHz 开关频率
- 相电流分流电阻和相电压信号调节
- TIEVM-MC-MODULE-F280013x 子卡
 - 采用 48 引脚 LQFP 封装的单一 C2000 系列 MCU
 - 模拟检测放大和滤波
 - 支持 XDS110ISO-EVM UART 的隔离式调试器仿真器板接口

2.1.1 辅助电源

基于 UCC28881 的非隔离式高压降压电源提供辅助电力, 可为 TIEVM-MTR-HVINV 提供高达 200mA 的 15VDC 电源。图 2-2 展示了 UCC28881 高压降压电源电路。

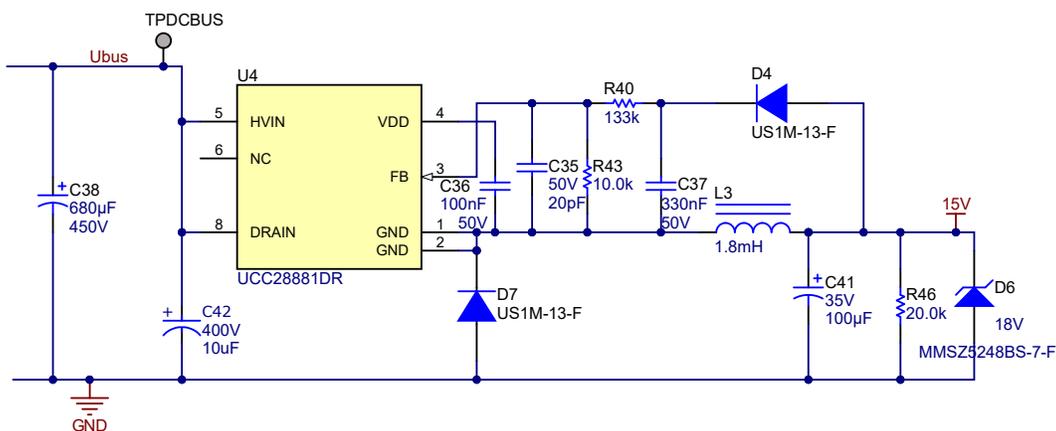


图 2-2. 高压降压电源电路

TPS54202 降压转换器用于通过 15VDC 辅助电源产生 5VDC 电源。然后, TLV74033P LDO 会调节 5V 电源轨, 以便为 MCU 子板提供 3.3VDC 电源。

2.1.2 直流链路电压检测

直流电压检测电路用于将整流后的直流总线电压信号转换为低压信号。该电路通过低成本电阻网络实现，如图 2-3 所示。直流总线电压也可用于估计交流输入电压，因为交流输入电路的物理参数为已知的常量。

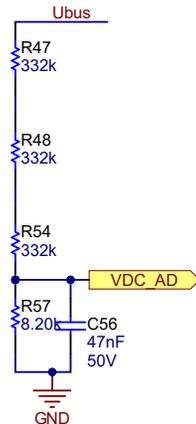


图 2-3. 直流总线电压检测电路

2.1.3 电机相电压检测

使用适用于 TIEVM-MTR-HVINV 的软件，可以实现增强型滑模观测器 (eSMO) 以及磁通、角度、速度和扭矩 (FAST™) 两种无传感器观测器。eSMO 仅需要常用的三相电机电流检测，而 FAST 观测器额外需要三相电机电压检测，以便改善低速性能并提高电机转速估算的精度。

请注意，软件参数 (USER_ADC_FULL_SCALE_VOLTAGE_V) 取决于对电机相的电压反馈进行检测的电路。图 2-4 展示了如何使用基于电阻分压器的电压反馈电路，根据 ADC 输入范围对电机电压进行滤波和缩放。类似的电路用于测量全部三个项目，包括两个压缩机和风扇电机以及直流总线。

考虑到 ADC 输入的最大电压为 3.3V，该参考设计中的微控制器可测量的最大相电压反馈可按照方程式 1 进行计算。

$$V_{FS} = V_{ADC_FS} \times G_V = 3.3 \text{ V} \times 122.46 = 404.13 \text{ V} \quad (1)$$

其中

- G_V 是衰减因子，并且 G_V 可通过方程式 2 进行计算

$$G_V = \frac{(R62 + R67 + R70 + R74)}{R74} = \frac{(332 \text{ k}\Omega + 332 \text{ k}\Omega + 332 \text{ k}\Omega + 8.2 \text{ k}\Omega)}{8.2 \text{ k}\Omega} = 122.46 \quad (2)$$

对于该电压反馈电路，在 user_mtr1.h 中进行以下设置：

```

///  
#define USER_M1_ADC_FULL_SCALE_VOLTAGE_V
    
```

FAST 估算器中需要使用电压滤波器极点，以便准确检测电压反馈。使滤波器的电压足够低，以便能够滤除 PWM 信号，同时允许高速电压反馈信号通过滤波器。通常，使用几百 Hz 的截止频率便足以过滤掉 5 至 20kHz 的 PWM 频率。只有在运行超高速电机时生成 kHz 量级相电压频率的情况下，才需更改硬件滤波器。

在该参考设计中，滤波器极点设置可以使用[方程式 3](#) 来计算：

$$f_{\text{filter_pole}} = \frac{1}{(2 \times \pi \times R_{\text{parallel}} \times C)} = 405.15 \text{ Hz} \quad (3)$$

where,

$C = 47\text{nF}$

$$R_{\text{parallel}} = \left(\frac{(332 \text{ k}\Omega + 332 \text{ k}\Omega + 332 \text{ k}\Omega) \times 8.2 \text{ k}\Omega}{(332 \text{ k}\Omega + 332 \text{ k}\Omega + 332 \text{ k}\Omega) + 8.2 \text{ k}\Omega} \right) = 8.133 \text{ k}\Omega$$

下面的代码示例显示了 `user_mtr1.h` 中是如何定义该极点的：

```
#!/ \brief Defines the analog voltage filter pole location, Hz
#define USER_M1_VOLTAGE_FILTER_POLE_Hz (416.3602877f)
```

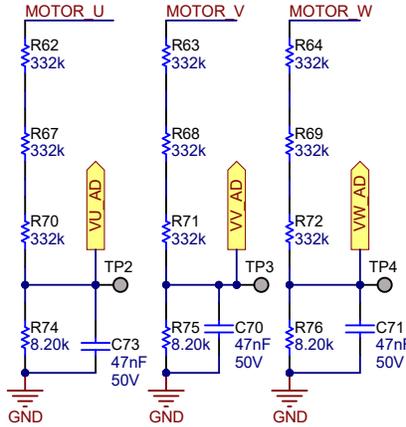


图 2-4. 电机电压检测电路

2.1.4 电机相电流检测

TIEVM-MTR-HVINV 设计支持单分流器或三分流器检测方案的相电流测量。在工程的构建配置中，可以选择这两种电流检测技术中的任何一种。默认 `Flash_MtrInv_3SC` 构建配置支持三分流器电流检测方法，`Flash_MtrInv_1SC` 支持单分流器电流检测方法，如[节 2.1.4.2](#) 中所述。

2.1.4.1 三分流器电流检测

作为电机控制算法的一部分，微控制器会对流经电机的电流进行采样，每个 PWM 周期采样一次。TIEVM-MC-MODULE 子板支持单分流器和三分流器电流检测。

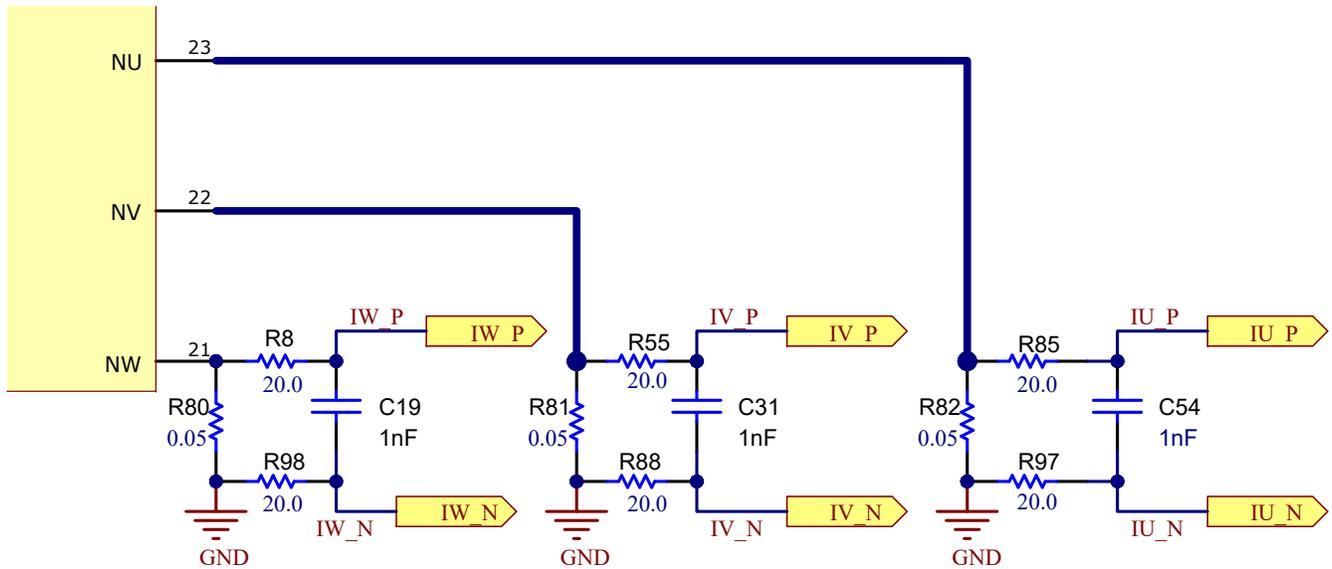


图 2-5. 三分流器电阻电路

为了测量正电流和负电流，测量电路需要一个失调基准电压，其值为 ADC 最大允许输入的一半。该 1.65V 失调基准电压通过一个电压跟随器生成，如图 2-6 所示：

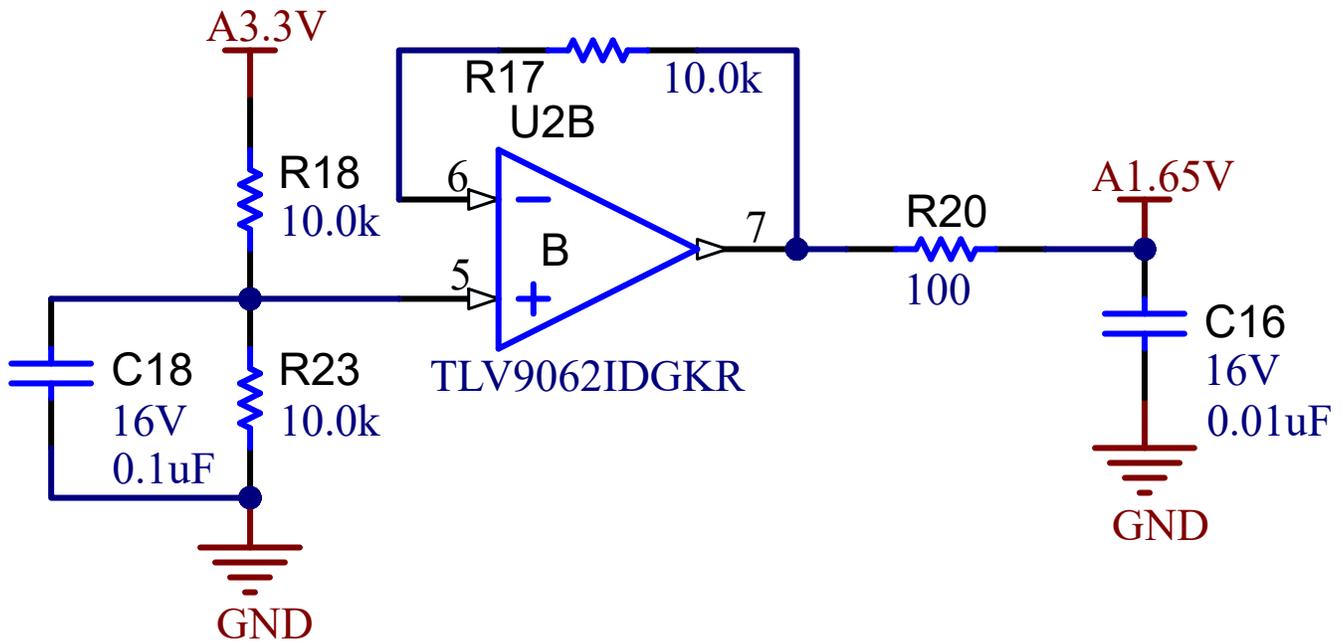


图 2-6. 3.3V 输入电路提供的 1.65V 基准

图 2-7 展示了电机电流如何表示为电压信号，其中包含滤波、放大和相对于子板 ADC 输入范围中心的偏移。该电路用于 PMSM 三相中的每个相。方程式 4 给出了该电路的传递函数。

$$V_{OUT} = V_{OFFSET} + (I_{IN} \times R_{SHUNT} \times G_i) \quad (4)$$

其中

- $R_{shunt} = 0.05 \Omega$
- $V_{offset} = 1.65 V$

利用计算出的电阻值，可得到图 2-4 所示的检测电路， G_i 由方程式 5 给出。

$$G_i = \frac{R_{fb}}{R_{in}} = \frac{R18}{(R97 + R15)} = \frac{10 \text{ k}\Omega}{20 + 2.4 \text{ k}\Omega} = 4.132 \quad (5)$$

微控制器可测量的最大峰峰值电流由 [方程式 6](#) 给出。

$$I_{\text{scale_max}} = \frac{V_{\text{ADC_max}}}{R_{\text{SHUNT}} \times G_i} = \frac{3.3}{0.05 \times 4.132} = 15.97 \text{ A} \quad (6)$$

其峰峰值为 $\pm 7.99\text{A}$ 。以下代码片段显示了如何在 `user_mtr1.h` 文件中为压缩机电机定义该值：

```
/*! \brief Defines the maximum current at the AD converter
#define USER_M1_ADC_FULL_SCALE_CURRENT_A (15.97f)
```

正确的电流反馈极性也很重要，因为这样才能确保微处理器精确测量电流。在该硬件电路板配置中，分流电阻器的负引脚接地，同时与运算放大器的反相引脚连接。突出显示的符号需要在软件中配置为具有正确的电流反馈极性，如 `user.mtr1.h` 中的以下代码片段所示：

```
// define the sign of current feedback based on hardware board
#define USER_M1_SIGN_CURRENT_SF (1.0f)
```

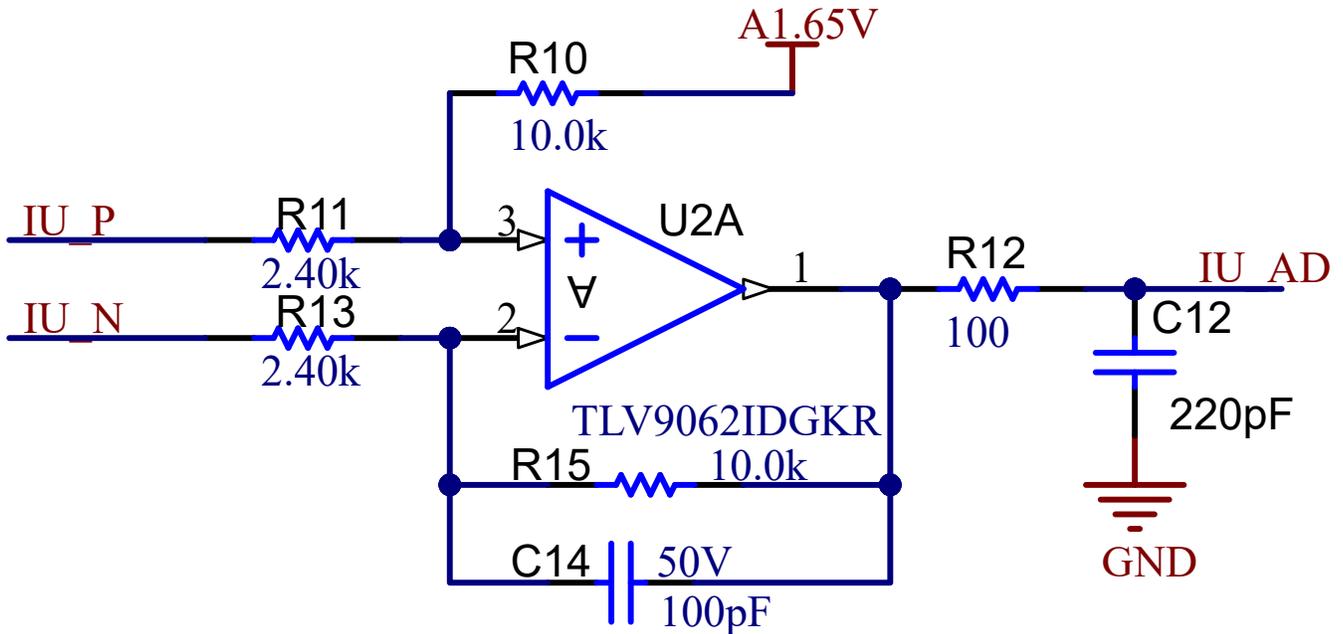


图 2-7. TMS320F2800137 的三分流器电流检测电路

2.1.4.2 单分流器电流检测

单分流器电流检测技术测量直流链路总线电流，并在了解功率 FET 开关状态的情况下重建电机的三相电流。使用 [单一直流链路分流器的 PMSM 无传感器 FOC 应用手册](#) 中详细介绍了单分流器技术。

在该参考板上，通过移除两个分流器并短接电源模块的 U/V/W 接地连接来实现单分流器电流检测技术，如 [图 2-8](#) 所示。

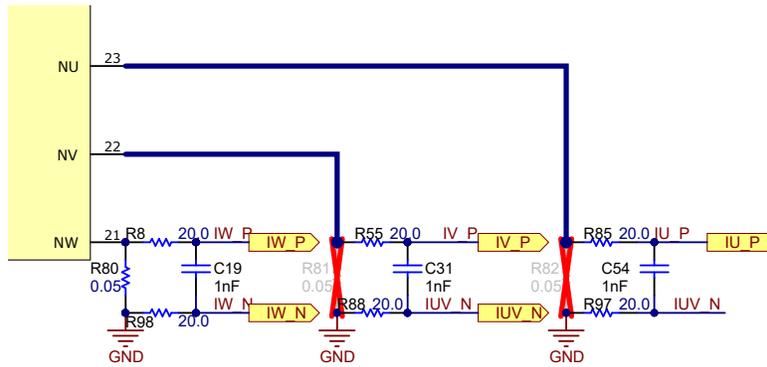


图 2-8. 单分流器电流检测电阻配置

1. 在主板上，移除电流采样电阻 R81 和 R82，只保留电流采样电阻 R80 来检测直流链路电流。
2. 在 TMS320F2800137 子板上，移除 C23 以增加单分流器采样的 U3A 带宽。
3. 使用粗导线将 NU、NV 和 NW 引脚连接在一起。

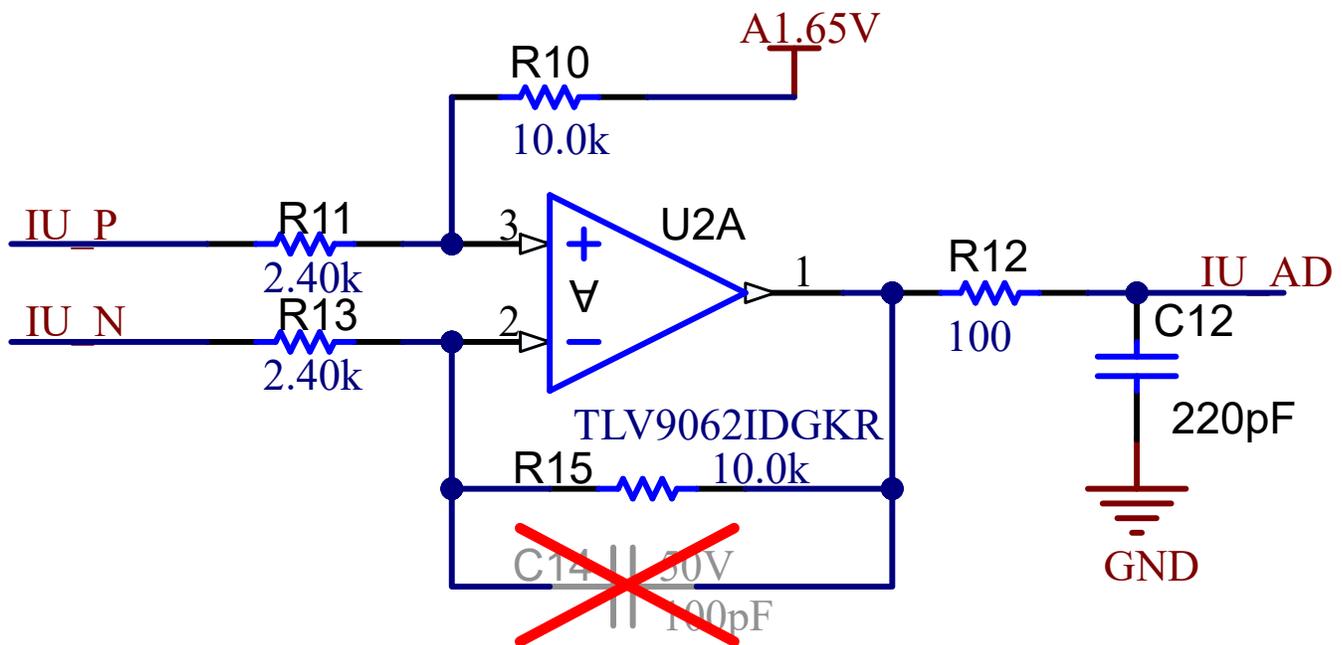


图 2-9. TMS320F2800137 的单分流器电流检测电路

默认情况下，电路板具有三个分流电阻。图 2-10 展示了分流电阻的布局。要实现单分流器电阻检测，请在保留 R80 的同时移除 R81 和 R82。将 NU、NV 和 NW (R80、R81 和 R82 的引脚 2) 焊接在一起。所有三相电流现在全部仅流经 R80。

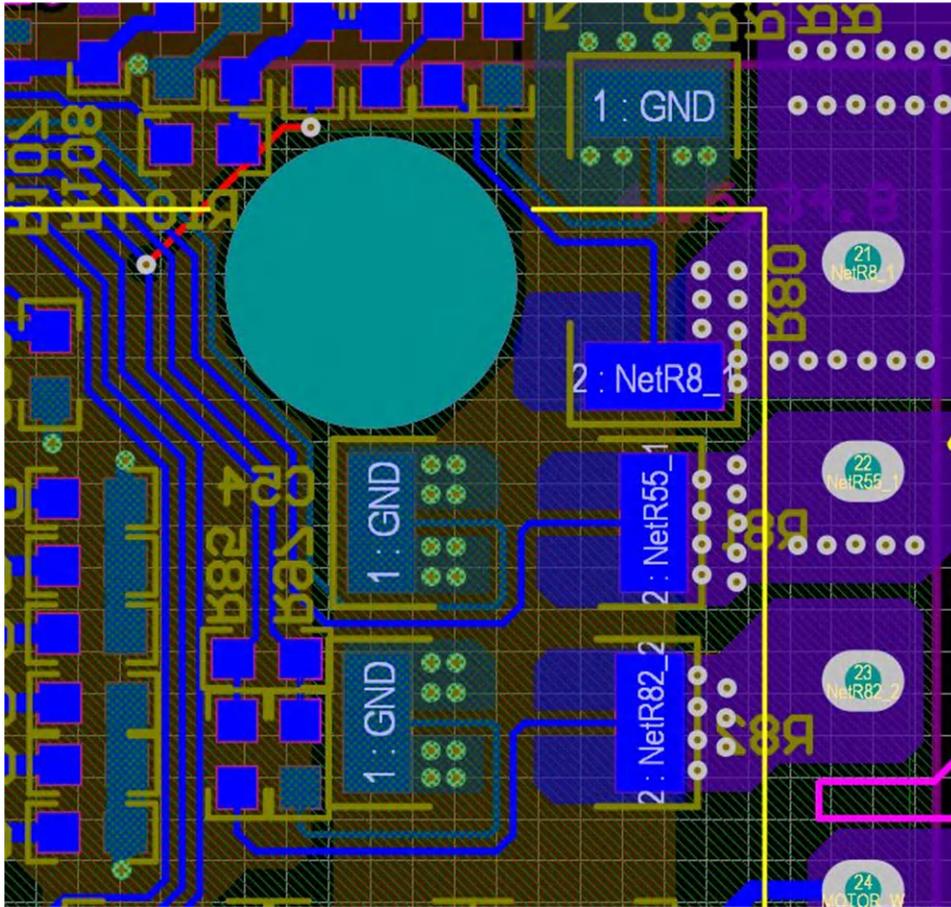


图 2-10. 分流电阻器布局

直流链路电流是单向信号，因此可以将直流电流偏移设置为最小值或最大值，以提高直流链路电流的 ADC 采样范围，如图 2-11 所示。在 TMS320F2800137 子板上，将 R23 从 10k Ω 更改为 1k Ω /1%，使基准电压具有 0.3V 失调电压，以进行直流电流检测。

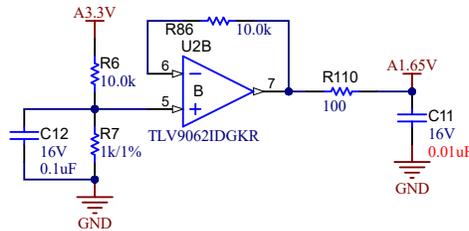


图 2-11. TMS320F2800137 子板单个分流器的直流偏移基准

该电流采样电路的传递函数和单分流器的计算与三分流器相同。

2.1.5 外部过流保护

TIEVM-MTR-HVINV 实现有两个冗余过流保护 (OCP) 电路。首先，外部比较器能够向三相逆变器报告过流故障。其次，子板 MCU 能够通过使用内部比较器报告过流故障。如果任一 OCP 机制报告故障，则三相逆变器和 MCU 都会转换到故障状态。系统将保持处于故障状态，直到手动清除该状态。

图 2-12 展示了外部 OCP 电路。该电路会将三个电流相求和，然后将该和与参考值进行比较。如果三相电流之和超过参考值，则“IPM_CIN”信号进行切换，并且 IPM 向微控制器报告过流故障。确切的过流保护电流可以通过以下公式计算。²

首先，基准电压 V_- 通过分压器从 3.3V 电源轨生成。

$$V_- = 3.3 \text{ V} \times \frac{R_{108}}{R_{107} + R_{108}} = 3.3 \text{ V} \times \frac{1 \text{ k}}{20 \text{ k} + 1 \text{ k}} = 0.15714 \text{ V} \quad (7)$$

接下来，请注意从任何 IPM 相到结点 U_{10+} 的电阻为 5.12kΩ。在此情况下，其中两个相（例如，V 和 W）约为 0A，而第三个相（U）出现电流尖峰并触发 OCP 电路。因此，请考虑将 R_V 和 R_W 并联。

$$R_U = R_V = R_W = R_{g7} + R_g = 5.12 \text{ k} \quad (8)$$

$$R_{V||W} = \left(\frac{1}{R_V} + \frac{1}{R_W} \right)^{-1} = \left(\frac{2}{5.12 \text{ k}} \right)^{-1} = 2.56 \text{ k} \quad (9)$$

结点 U_{10+} 处的电压可以称为 V_+ 。 V_+ 通过相对于 IPM 每个相的电压（ V_U 、 V_V 、 V_W ）的分压生成。请注意， V_V 和 V_W 均假设为 0V，因为 I_V 和 I_W 均声明等于或接近 0A。

$$V_+ = \left(V_U - V_{VW} \right) \times \frac{R_{V||W}}{R_U + R_{V||W}} = V_U \times \frac{R_{V||W}}{R_U + R_{V||W}} \quad (10)$$

$$V_+ = V_U \times \frac{R_{V||W}}{R_U + R_{V||W}} = V_U \times \frac{2.56 \text{ k}}{5.12 \text{ k} + 2.56 \text{ k}} = \frac{V_U}{3} \quad (11)$$

V_U 还定义为过流事件期间，分流电阻 R_{shunt} 上的电压。

$$V_U = R_{shunt} \times I_{OCP} = 0.05 \times I_{OCP} \quad (12)$$

要触发 OCP 电路， V_+ 必须大于或等于基准电压 V_- 。

$$V_+ = \frac{0.05 \times I_{OCP}}{3} \geq V_- \quad (13)$$

$$I_{OCP} \geq \frac{V_-}{\left(\frac{0.05}{3} \right)} = \frac{0.15714}{\left(\frac{0.05}{3} \right)} = 9.42857 \text{ A} \quad (14)$$

² 电路板的某些版本可能对外部电路具有更高的 OCP 限制。在这些情况下，内部 (CMPSS) 保护仍会按照预期级别完全正常运行。更多信息，请参阅节 6.1。

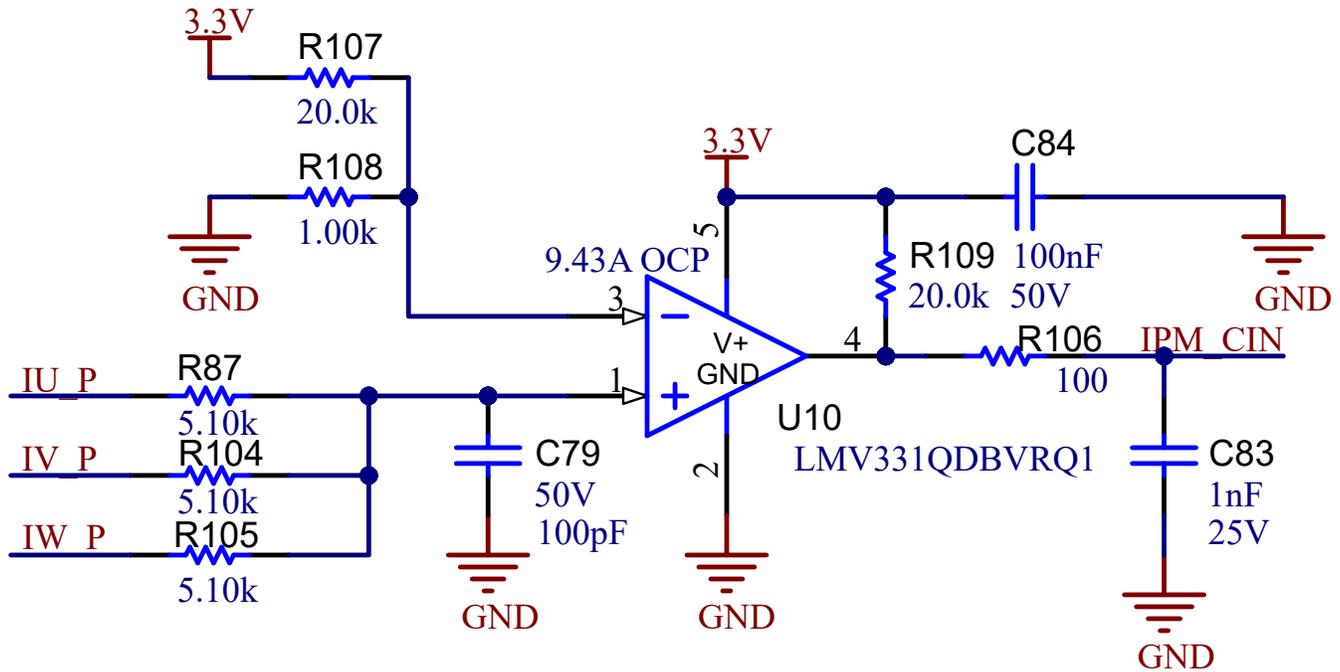


图 2-12. 外部过流保护电路

2.1.6 TMS320F2800F137 的内部过流保护

子板 MCU 具有内部窗口比较器，可将其配置用于监测电流的三相。所有比较器和 PWM 跳闸任务均独立于软件运行，并且从比较器报告过流故障到 PWM 转换为故障状态，中间没有中断延迟。这样可以快速保护外部 IPM 或功率器件。

2.2 入门硬件

本节详细介绍了电路板硬件必要的设备和测试设置。

2.2.1 测试条件和设备

验证以下内容以测试 TIEVM-MTR-HVINV：

- 对于输入，如果使用交流电源，则电源电压必须处于交流 165V 至 265V 范围内，如果使用直流电源，则电源电压必须处于 100V 至 400V 范围内。将输入交流电源的输入电流限制为 10A，或者将直流电源设置为 6.5A。
- 对于输出，搭配使用三相 PMSM 和功率计，以施加反向旋转。
 - 如套件内容中所述，德州仪器 (TI) 粗略地建议使用 Estun EMJ-04APB22 来测试 TIEVM-MTR-HVINV 系统。
 - 对于电机选择，请注意，TIEVM-MTR-HVINV 的最大输出功率为 750W。

对于电路板测试，高压合格工程师必须使用以下设备：

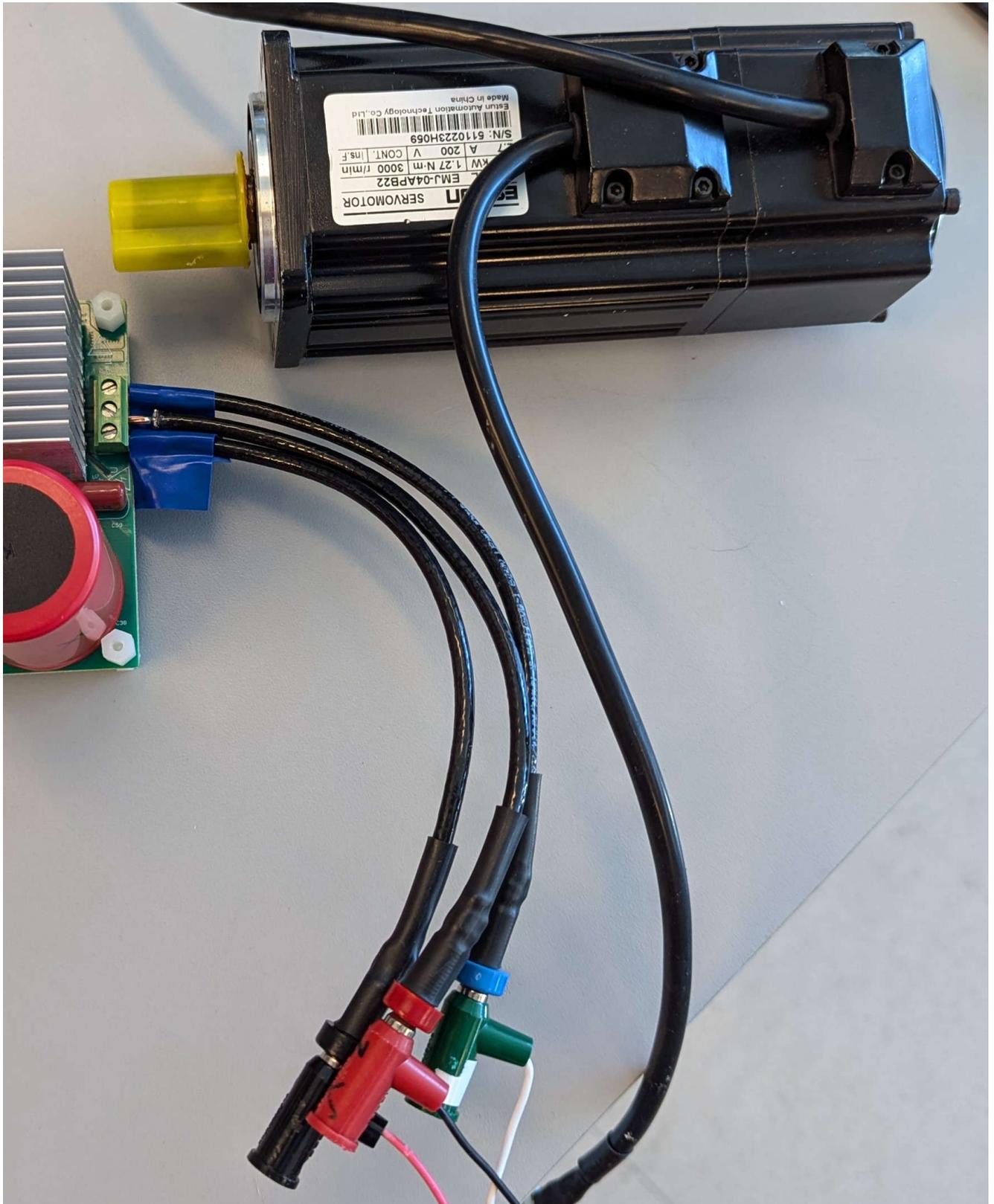
- 隔离式交流源

此外还强烈建议使用以下设备进行全面测试，但基本操作不必使用这些设备：

- 直流电源
- 测力计
- 单相功率分析仪
- 三相功率分析仪
- 示波器
 - 高压探头
 - 至少 1 个电流探头
- 万用表

2.2.2 测试设置

1. (可选) 使用随附的 4 颗螺钉将电路板连接到螺柱。TIEVM-MTR-HVINV 的每个角都有一个圆孔，正是用于此目的。
2. 如果尚未将子板连接到主板的顶部，请完成该连接。主板的 2 排接头 (J15) 应该与子板的 2 排连接器 (J3) 对齐。
3. 子板连接器 J3 的关键引脚 8 应进行削波 (如果尚未完成)。
4. 观察 XDS110ISO-EVM 的底部并找到关键引脚。将 XDS110ISO-EVM 连接到子板连接器 J3，以使每个电路板的关键引脚对齐。
5. 确认子板引导模式开关 (S1) 处于所需的状态。
6. 将电机线连接到 J10。
 - a. 对于 Estun EMJ-04APB22，下面是一张展示其连接方法的示例图像：



7. 连接测量设备，如万用表、示波器探头等，探测或分析各种信号和参数。可用的测试点可在电路板原理图 (TPx) 中查看。
8. 使用连接到 J5 ACL、ACN 和接地连接器的隔离式交流或直流电源，为电路板上电。
 - a. 交流电源的最大输出为 265-VAC (频率为 50/60Hz) 。
 - b. 直流电源的最大输出为 380-VDC。

也可使用以下选项。

- 如果用户需要外部 SPI，则子板连接器 J4 引脚 4 至 7 可用作 SPI 信号。此时 XDS110ISO-EVM 板载 DAC 不可用。此连接可通过以下方法之一使用：
 - 直接连接到子板连接器 J4 上的 SPI 信号。此连接不是隔离连接，必须使用外部隔离器。
 - 连接到隔离式 UART 接口 J14，该接口在某些电路板版本上可能不可用。
- 如果使用此连接，请取消组装子板电阻 R8，以确保 XDS110ISO-EVM 板载 DAC 不会干扰 SPI 外设到控制器信号。
- 默认电路板状态采用三相电流检测。如果需要使用单相，请参阅节 2.1.4.2。

3 电机控制软件

3.1 三相 PMSM 驱动系统设计理论

为了正确解释 TIEVM-MTR-HVINV 的软件，需要具备背景知识。一般来说，了解三相 PMSM 电机驱动系统的基本理论，以及无传感器 FOC 控制如何融入该系统非常重要。

永磁同步电机 (PMSM) 具有一个绕线定子、一个永磁转子组件，以及用于检测转子位置的内部或外部机构。检测机构会提供位置反馈以适当地调整定子基准电压的频率和振幅，从而使磁体组件保持旋转。一个内部永磁转子和外部绕组的组合提供低转子惯性、有效散热和电机尺寸减少等优势。

- 同步电机构造：永磁体被牢牢固定在旋转轴上，生成了一个恒定的转子磁通。这个转子磁通通常具有一个恒定的磁通量。当加电时，定子绕组产生一个旋转的电磁场。为了控制旋转磁场，必须控制定子电流。
- 根据机器的额定功率范围和速度，转子的实际结构会有所不同。对于高达数千瓦的同步电机来说，永磁体是很好的选择。为了获得更高的额定功率，转子通常由接通直流电的绕组组成。转子的机械结构是根据所需的磁通量梯度和磁极数而设计的。
- 定子和转子磁通的交感产生了一个转矩。由于定子被牢固地安装在电机架上，而转子可自由旋转，因此转子的旋转会产生一个有用的机械输出，如图 3-1 所示。
- 必须仔细控制转子磁场和定子磁场间的角度，以产生最大扭矩和实现较高的机电转换效率。为了实现这一目的，在同一转速和扭矩条件下，为了尽可能少地消耗电流，在关闭速度环路后需要使用无传感器算法进行微调。
- 旋转中的定子磁场的频率必须与转子永磁磁场的频率相同，否则转子就会经历快速的正负扭矩交替。这导致扭矩产生效果不佳，并且在机器器件上产生过多的机械抖动、噪声和机械应力。此外，如果转子惯性使转子不能对这些摆动做出响应，那么转子在同步频率上停止转动，并且对静止转子的平均扭矩：零扭矩做出响应。这意味着机器会出现一种称为牵出的现象。这也是为什么同步机器不能自启动的原因。
- 转子磁场与定子磁场间的角度必须等于 90° 以获得最高的互转矩产出量。为了产生正确的定子磁场，该同步需要知道转子位置。
- 通过将不同转子相位的输出组合在一起，可将定子磁场设定为任一方向和强度以产生相应的定子磁通。

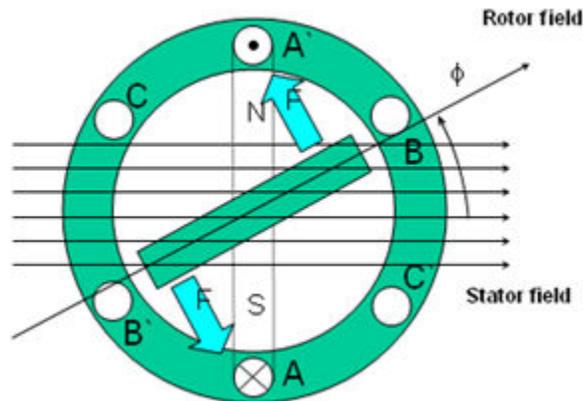


图 3-1. 旋转的定子磁通和转子磁通之间的相互作用产生扭矩

3.1.1 PMSM 的磁场定向控制

为了实现更好的动态性能，需要采用更加复杂的控制方案来控制电机。借助微控制器提供的数学处理能力，可以实施先进的控制策略，这些策略使用数学变换将永磁电机中的扭矩生成和磁化功能解耦。这种解耦的扭矩和磁化控制通常称为转子磁通定向控制、矢量控制，或简称磁场定向控制 (FOC)。

在直流 (DC) 电机中，定子和转子的励磁是独立控制的，产生的扭矩和磁通可以独立调整，如图 3-2 所示。磁场激励强度（例如，磁场激励电流的振幅）决定了磁通的大小。通过转子绕组的电流确定了扭矩是如何生成。转子上的换向器在扭矩产生过程中发挥着有趣的作用。换向器与电刷接触，这个机械构造旨在将电路切换至机械对齐的

绕组以产生最大的扭矩。这样的安排意味着，电机的扭矩产生在任何时候都非常接近于理想情况。这里的关键点是，通过管理绕组以保持转子绕组产生的磁通与定子磁场垂直。

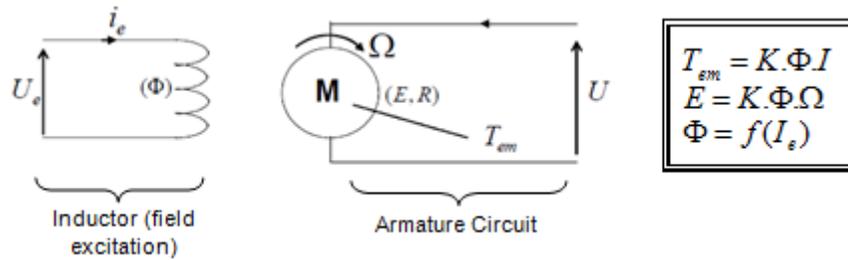


图 3-2. 在直流电机模型中磁通和扭矩是独立控制的

同步和异步电机上 FOC 的目的在于能够分别控制扭矩生成分量和磁化磁通分量。FOC 算法可实现扭矩和定子电流磁化磁通分量的解耦。借助于磁化的去耦合控制，定子磁通的扭矩生成分量现在可以被看成是独立扭矩控制。为了将扭矩和磁通解耦，有必要进行若干数学变换，此时微控制器便会充分发挥其价值。利用微控制器提供的处理能力，可以非常快速地执行使这些数学变换。相应地，这意味着控制电机的整个算法可以高速执行，从而实现更高的动态性能。除了解耦，还可以使用电机的动态模型来计算很多值，如转子磁通角和转子速度，从而提高控制质量。

根据电磁定律，同步电机中产生的扭矩等于两个现有磁场的矢量叉积，如方程式 15 所示。

$$\tau_{em} = \vec{B}_{stator} \times \vec{B}_{rotor} \quad (15)$$

该表达式表明，如果定子和转子磁场正交，即呈 90 度，则扭矩最大。如果可以一直保持此条件且磁通可以正确定向，则会减小扭矩波动并提供更好的动态响应。然而，若要如此，必须知道转子位置：这可以通过物理位置传感器（如增量编码器）或无传感器转子位置观测器来实现。

为了实现定子磁通与转子磁通正交对齐的目标，（直接，正交）旋转坐标系中定子电流的 d 轴分量应设置为零。（d, q）旋转坐标系在节 3.1.1.1.2 中有更详细的说明。当此条件为真时，定子磁通和转子磁通正交对齐。定子电流的 d 轴分量还可以在某些情况下用于弱磁，这样可以减少反电动势，使电机以更高的速度运行。

FOC 包含控制由一个矢量表示的定子电流。这个控制所基于的设计是，将三相时间和速度相关系统变换为两坐标（d 和 q 坐标）非时变系统。这些设计导致一个与 DC 机器控制结构相似的结构。FOC 电机需要两个常量作为输入基准：扭矩分量（与 q 坐标对齐）和磁通分量（与 d 坐标对齐）。由于 FOC 完全基于这些设计，此控制结构处理即时电量。这使得在每次的工作运转过程中（稳定状态和瞬态）均可实现准确控制，并且与受限带宽数学模型无关。因此，FOC 通过以下方式解决了经典电机控制方案存在的问题：

- 轻松达到恒定基准（定子电流的扭矩分量和磁通分量）
- 轻松应用直接扭矩控制，这是因为在（d, q）坐标系中，扭矩的表达式定义如方程式 16 所示。

$$\tau_{em} \propto \psi_R \times i_{sq} \quad (16)$$

通过将转子磁通（ ψ_R ）的振幅保持在一个固定值，在扭矩和扭矩分量（ i_{sq} ）之间实现线性关系。因此，可通过控制定子电流矢量的转矩分量来控制转矩。

3.1.1.1 空间矢量定义和投影

交流电机的三相电压、电流和磁通可根据复数空间矢量进行分析。对于电流，空间矢量可定义如下。假设 i_a 、 i_b 、 i_c 是定子相中的瞬时电流，则复定子电流矢量的定义如方程式 17 所示。

$$\vec{i}_s = i_a + \alpha i_b + \alpha^2 i_c \quad (17)$$

其中

- $\alpha = e^{j\frac{2}{3}\pi}$ 和 $\alpha^2 = e^{j\frac{4}{3}\pi}$ 表示空间运算符

图 3-3 展示了定子电流复空间矢量。

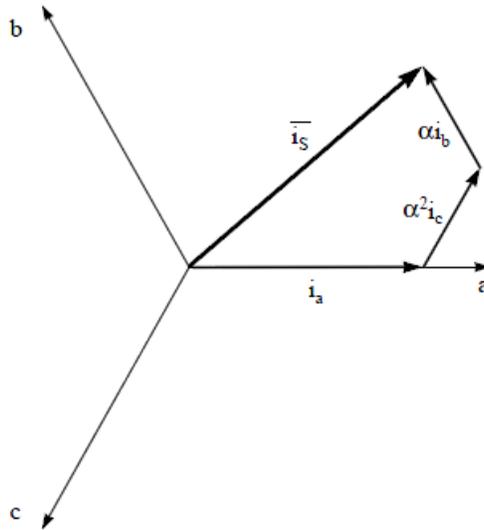


图 3-3. 定子电流空间矢量及其以 (a,b,c) 坐标系表示的分量

其中

- a、b 和 c 是三相系统轴

这个电流空间矢量描述了三相正弦系统，这个系统仍然需要变换为一个两坐标非时变系统。这个变换可拆分为两个步骤：

- $(a, b) \Rightarrow (\alpha, \beta)$ (Clarke 变换)，输出一个两坐标时变系统
- $(\alpha, \beta) \Rightarrow (d, q)$ (Park 变换)，输出一个两坐标时不变系统

3.1.1.1.1 $(a, b) \Rightarrow (\alpha, \beta)$ Clarke 变换

可以使用另外一个仅包含两相 (α, β) 正交轴的坐标系来表示该空间矢量。假设 a 轴和 α 轴方向相同，可以得到图 3-4 所示的矢量图。

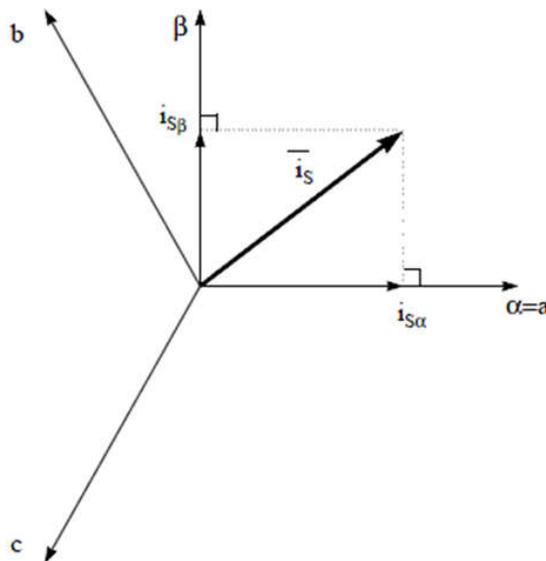


图 3-4. 静止坐标系中的定子电流空间矢量

将三相系统修改为 (α, β) 二维正交系统的投影如方程式 18 所示。

$$\begin{aligned} i_{s\alpha} &= i_a \\ i_{s\beta} &= \frac{1}{\sqrt{3}}i_a + \frac{2}{\sqrt{3}}i_b \end{aligned} \quad (18)$$

两相 (α, β) 电流仍取决于时间和速度。

3.1.1.1.2 $(\alpha, \beta) \Rightarrow (d, q)$ Park 变换

这是 FOC 内最重要的变换。事实上，该投影在 (d, q) 旋转坐标系中修改了一个两相正交系统 (α, β) 。假设 d 轴与转子磁通对齐，那么图 3-5 展示了来自该二维坐标系的电流矢量的关系。

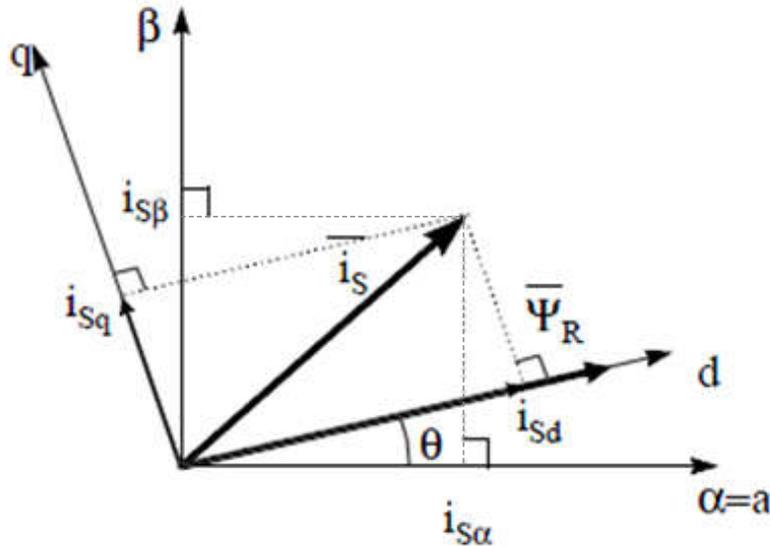


图 3-5. d, q 旋转坐标系中的定子电流空间矢量

电流矢量的磁通和扭矩分量由方程式 19 决定。

$$\begin{aligned} i_{sd} &= i_{s\alpha}\cos(\theta) + i_{s\beta}\sin(\theta) \\ i_{sq} &= -i_{s\alpha}\sin(\theta) + i_{s\beta}\cos(\theta) \end{aligned} \quad (19)$$

其中

- θ 是转子磁通位置

这些分量取决于电流矢量 (α, β) 分量和转子磁通位置；如果知道正确的转子磁通位置，那么，通过该投影， d, q 分量就变成一个常量。现在，两个相位电流变换为直流数量（非时变）。此时扭矩控制变得更容易，其中恒定的 i_{sd} （磁通分量）和 i_{sq} （扭矩分量）电流分量单独受到控制。

3.1.1.2 交流电机 FOC 基本配置方案

图 3-6 总结了使用 FOC 进行扭矩控制的基本配置方案。请注意，为了与示意图一致，本说明假设使用双分流器检测系统，但单分流器和三分流器检测方案也很常见。三分流器控制会采用略有不同的 Clarke 变换实现方式，而单分流器要求在 Clarke 变换之前进行程控相电流重构。无论上述什么情况，配置方案从 Clarke 变换的 α 和 β 输出开始是相同的。

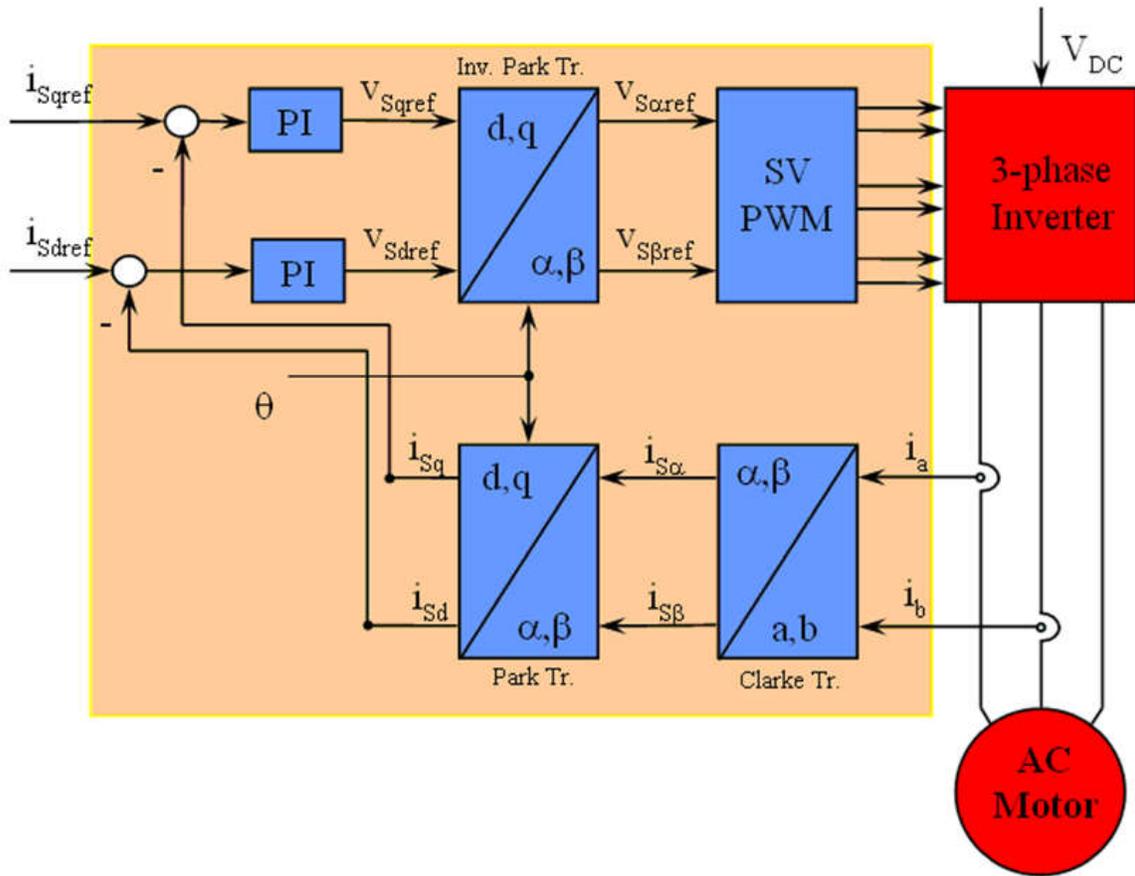


图 3-6. 交流电机 FOC 基本配置方案

测量了两个电机相电流。这些测量值馈入 Clarke 变换模块。这个模块的输出为 $i_{s\alpha}$ 和 $i_{s\beta}$ 。电流的这两个分量是 Park 变换的输入，该变换会计算 d,q 旋转坐标系中的电流。

i_{sd} 和 i_{sq} 分量与基准 i_{sdref} (磁通基准分量) 和 i_{sqref} (扭矩基准分量) 进行比较。此时，这个控制结构具有一个有意思的优势：只需改变磁通基准并获得转子磁通位置，该结构即可用于控制同步或感应电机。

在 PMSM 中，转子磁通是固定的，由磁体决定；无需产生任何磁通。因此，当控制 PMSM 时，将 i_{sdref} 设置为零。由于交流感应电机需要生成转子磁通才能运行，因此磁通基准一定不能为零。这很方便地解决了经典控制结构的一个主要缺陷：异步驱动至同步驱动的可移植性。

当使用转速 FOC 时，扭矩命令 i_{sqref} 可以是转速调节器的输出。电流调节器的输出是 V_{sdref} 和 V_{sqref} ；这些输出应用于 Park 逆变换。这个模块的输出是 $V_{s\alpha ref}$ 和 $V_{s\beta ref}$ ，它们是 (α, β) 静止正交坐标系中定子矢量电压的分量。这些是空间矢量脉宽调制 (PWM) 的输入。这个块的输出是驱动此反相器的信号。

请注意，Park 和 Park 逆变换均需要转子磁通位置。这个转子磁通位置的获得由交流机器的类型 (同步或异步机器) 而定。

3.1.1.3 转子磁通位置

转子磁通位置的相关知识是 FOC 的核心。事实上，如果该变量存在误差，则转子磁通与 d 轴不对齐，并且定子电流的磁通和扭矩分量 i_{sd} 和 i_{sq} 不正确。图 3-7 展示了 (a, b, c)、(α, β) 和 (d, q) 坐标系，以及转子磁通的正确位置和以同步速度随 (d, q) 坐标旋转的定子电流和定子电压空间矢量。

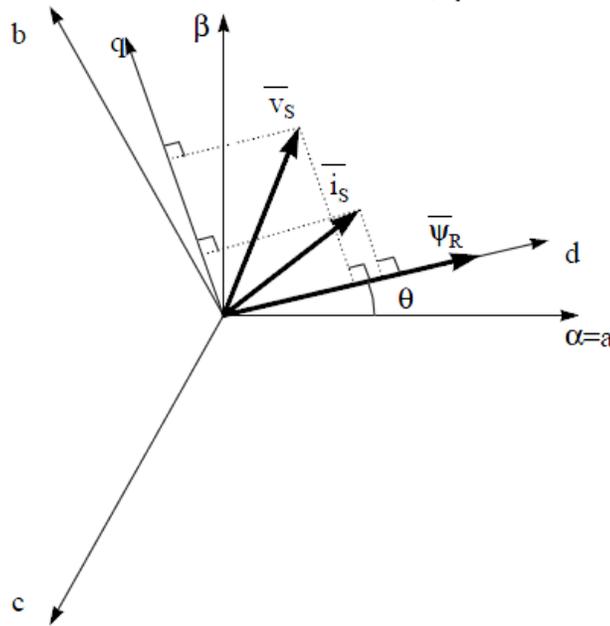


图 3-7. (d, q) 旋转坐标系中的电流、电压和转子磁通空间矢量

考虑同步或异步电机，转子磁通位置的测量是不同的：

- 在同步电机中，转子转速等于转子磁通转速。然后 θ （转子磁通位置）由位置传感器或转子速度的积分直接计算。
- 在异步电机中，转子转速不等于转子磁通转速（存在转差速度），因此需要使用特定的方法来计算 θ 。基本方法是使用一个电流模型，该模型需要 d, q 坐标系中的电机模型的两个公式。

理论上，针对 PMSM 驱动的 FOC 可用磁通实现对电机转矩的单独控制，这与直流电机的运行类似。换句话说，转矩和磁通互相之间去耦合。从静止坐标系到同步旋转坐标系间的变量变换需要知道转子位置信息。由于这种变换（所谓的 Park 变换），q 轴电流控制扭矩，而 d 轴电流强制设置为零。因此，该系统的关键模块是使用增强型滑模观测器 (eSMO) 或 FAST 估算器来估算转子位置。

图 3-8 展示了该参考设计中风扇 PMSM 的无传感器 FOC（使用 eSMO 并具有快速启动功能）的整体方框图。

图 3-9 展示了该参考设计中压缩机 PMSM 的无传感器 FOC（使用 eSMO 并具有弱磁控制 (FWC) 和每安培最大扭矩 (MTPA) 功能）的整体方框图。

图 3-10 展示了该参考设计中风扇 PMSM 的无传感器 FOC（使用 FAST 并具有快速启动功能）的整体方框图。

图 3-11 展示了该参考设计中压缩机 PMSM 的无传感器 FOC（使用 FAST 并具有弱磁控制 (FWC) 和每安培最大扭矩 (MTPA) 功能）的整体方框图。

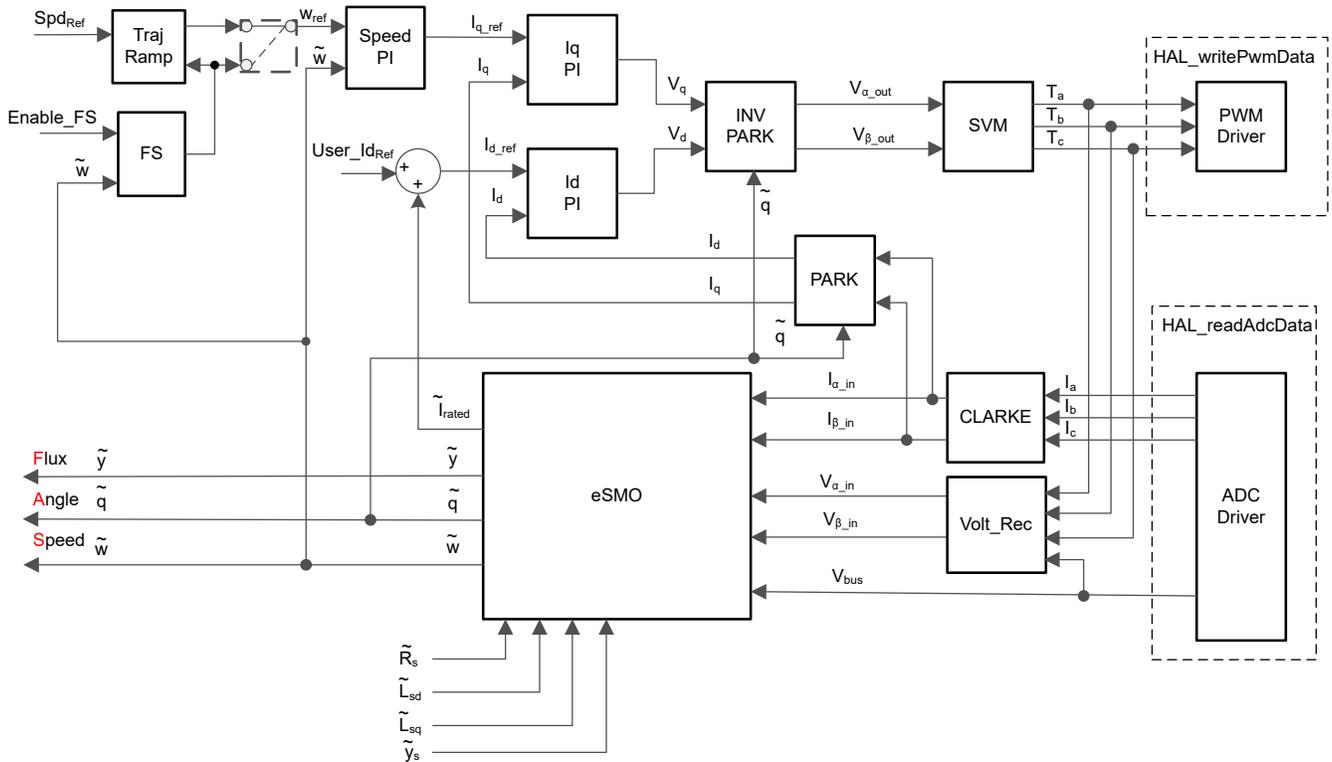


图 3-8. 使用 eSMO 并具有快速启动 (FS) 功能的 PMSM 的无传感器 FOC

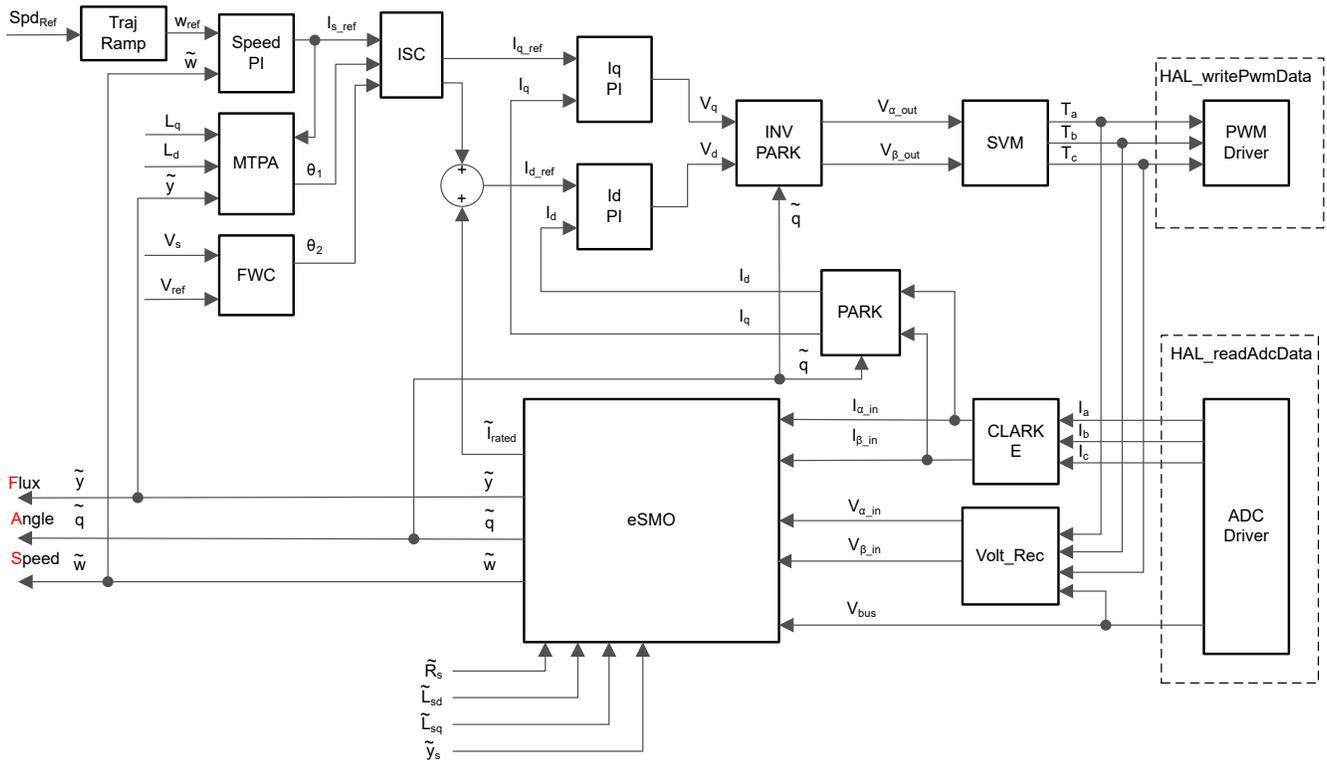


图 3-9. 使用 eSMO 并具有 FWC 和 MTPA 功能的 PMSM 的无传感器 FOC

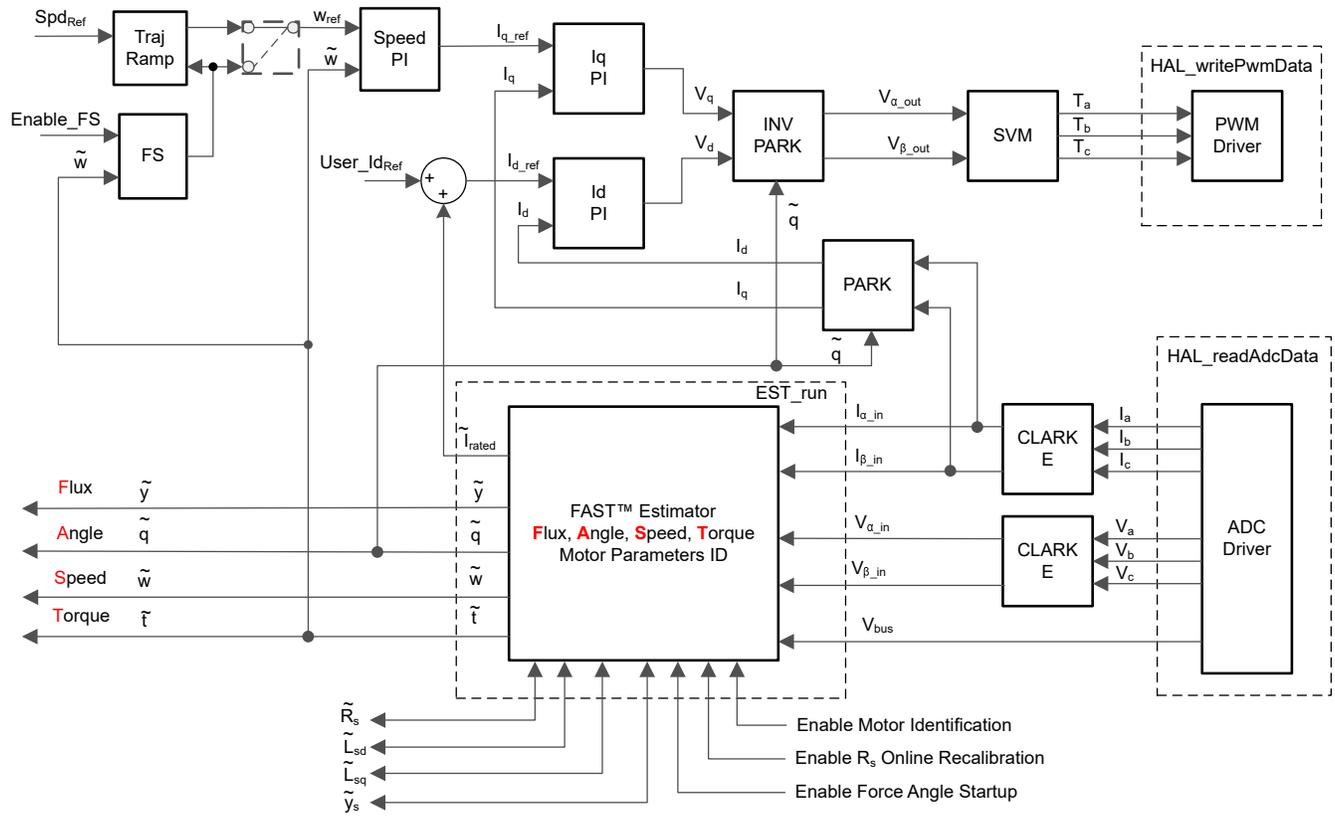


图 3-10. 使用 FAST 并具有快速启动 (FS) 功能的 PMSM 的无传感器 FOC

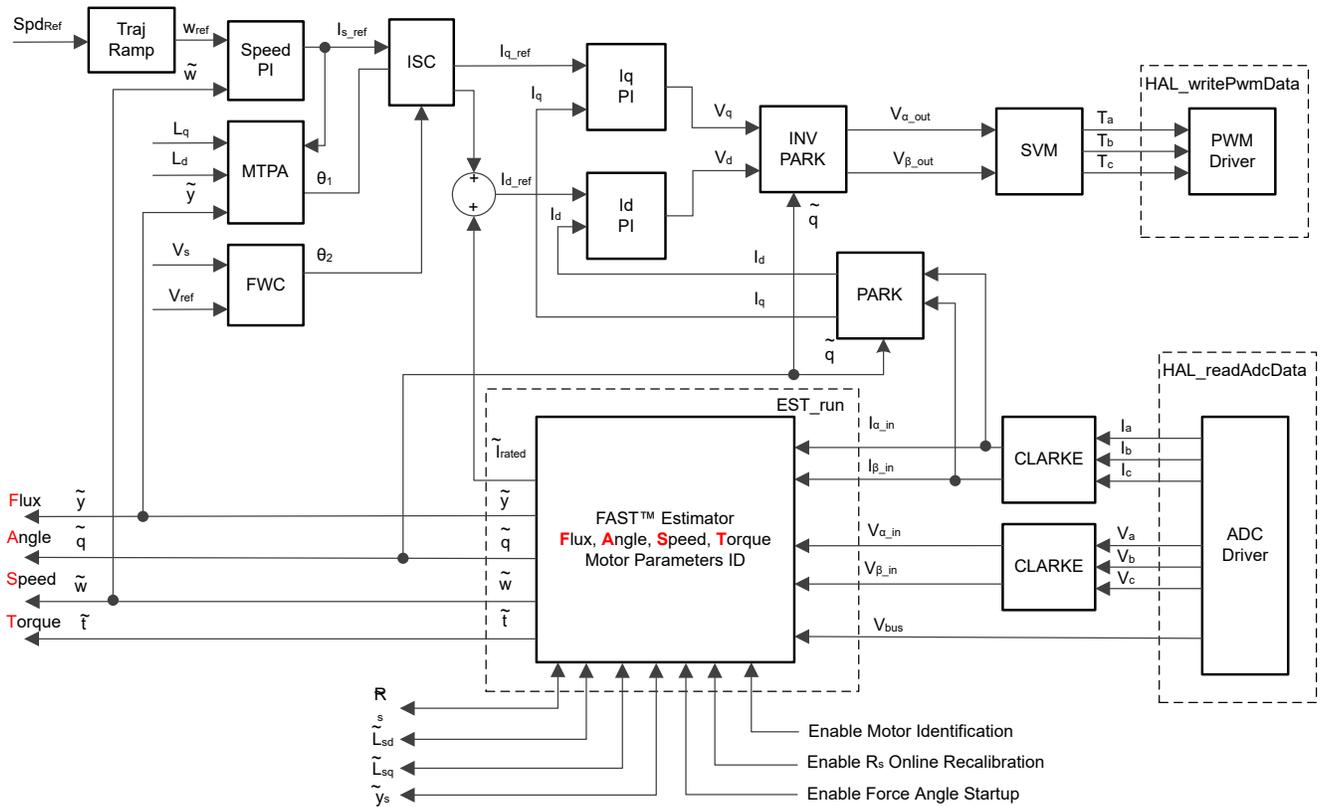


图 3-11. 使用 FAST 并具有 FWC 和 MTPA 功能的 PMSM 的无传感器 FOC

3.1.2 PM 同步电机的无传感器控制

在家用电器应用中，使用机械传感器会增加成本、尺寸和可靠性问题。为了克服这些问题，无传感器控制方法应运而生，可以通过多种估算方法在没有机械位置传感器的情况下获得转子转速和位置信息。滑模观测器 (SMO) 因其各种吸引人的特性 (包括可靠性、所需的性能和针对系统参数变化的稳健性) 而被广泛使用。

3.1.2.1 具有锁相环的增强型滑模观测器

基于模型的方法用于实现 IPMSM 驱动系统在电机以中高速运行时的无位置传感器控制。模型法通过反电动势或磁链模型估算转子位置。滑动模式观测器是基于滑模控制的观测器设计方法。系统的结构不是固定的，而是根据系统的当前状态有目的地改变，迫使系统按照预定的滑模轨迹运动。其优点包括响应速度快、稳健性高以及对参数变化和干扰不敏感。

3.1.2.1.1 IPMSM 的数学模型和 FOC 结构

IPMSM 的无传感器 FOC 结构如图 3-12 所示。在该系统中，eSMO 用于实现 IPMSM 系统的无传感器控制，eSMO 模型是利用反电动势模型和 PLL 模型设计的，用于估算转子位置和转速。

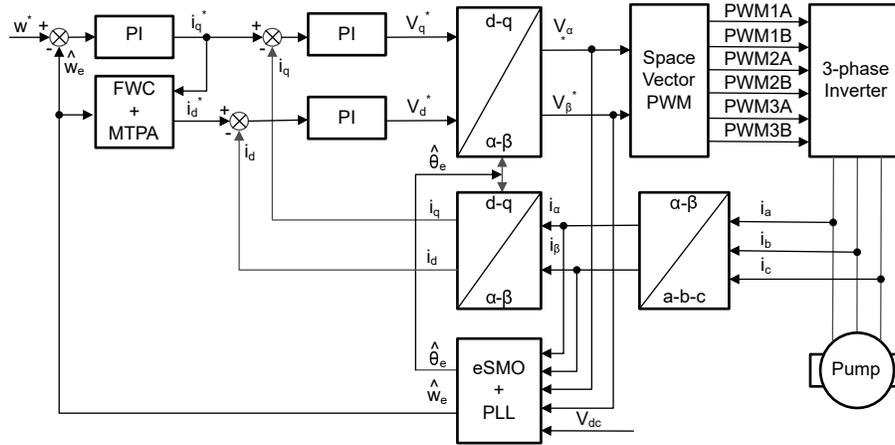


图 3-12. IPMSM 系统的无传感器 FOC 结构

IPMSM 由一个三相定子绕组 (a、b、c 轴) 和用于励磁的永磁体 (PM) 转子组成。电机由标准的三相逆变器进行控制。可以使用相位 a-b-c 量对 IPMSM 进行建模。通过适当的坐标变换，可以得到 d-q 转子坐标系和 $\alpha - \beta$ 静止坐标系中的动态 PMSM 模型。这些坐标系之间的关系如方程式 20 所示。通用 PMSM 的动态模型可以在 d-q 转子坐标系中写为：

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} R_s + pL_d & -\omega_e L_q \\ \omega_e L_d & R_s + pL_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_e \lambda_{pm} \end{bmatrix} \quad (20)$$

其中

- v_d 和 v_q 分别是 q 轴和 d 轴定子端电压
- i_d 和 i_q 分别是 d 轴和 q 轴定子电流
- L_d 和 L_q 分别是 q 轴和 d 轴电感
- p 是导数算子，用于简写 $\frac{d}{dt}$
- λ_{pm} 是永磁体产生的磁链
- R_s 是定子绕组的电阻
- ω_e 是转子的电角速度

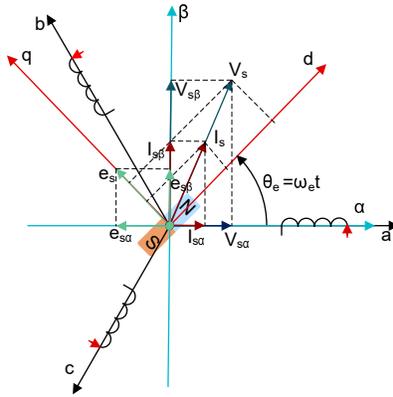


图 3-13. PMSM 建模坐标系的定义

通过使用如图 3-13 所示的 Park 逆变换，PMSM 的动力学可以在 α - β 静止坐标系中按照方程式 21 所示进行建模：

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \begin{bmatrix} R_s + pL_d & \omega_e(L_d - L_q) \\ -\omega_e(L_d - L_q) & R_s + pL_q \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \begin{bmatrix} e_\alpha \\ e_\beta \end{bmatrix} \quad (21)$$

其中

- e_α 和 e_β 是 α - β 轴上扩展电动势 (EEMF) 的分量，可以按照方程式 22 中所示进行定义：

$$\begin{bmatrix} e_\alpha \\ e_\beta \end{bmatrix} = (\lambda_{pm} + (L_d - L_q)i_d)\omega_e \begin{bmatrix} -\sin(\theta_e) \\ \cos(\theta_e) \end{bmatrix} \quad (22)$$

根据方程式 21 和方程式 22，通过等效变换和引入 EEMF 概念，可以将转子位置信息从电感矩阵中解耦出来，从而使 EEMF 成为唯一包含转子磁极位置信息的项。然后可以直接利用 EEMF 相位信息实现转子位置观测。使用定子电流作为状态变量，将 IPMSM 电压公式方程式 21 改写为状态公式：

$$\begin{bmatrix} \dot{i}_\alpha \\ \dot{i}_\beta \end{bmatrix} = \frac{1}{L_d} \begin{bmatrix} -R_s & -\omega_e(L_d - L_q) \\ \omega_e(L_d - L_q) & -R_s \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \frac{1}{L_d} \begin{bmatrix} V_\alpha - e_\alpha \\ V_\beta - e_\beta \end{bmatrix} \quad (23)$$

由于定子电流是唯一可以直接测量的物理量，因此在定子电流路径上选择滑动面：

$$s(x) = \begin{bmatrix} \hat{i}_\alpha - i_\alpha \\ \hat{i}_\beta - i_\beta \end{bmatrix} = \begin{bmatrix} \tilde{i}_\alpha \\ \tilde{i}_\beta \end{bmatrix} \quad (24)$$

其中

- \hat{i}_α 和 \hat{i}_β 是估算的电流
- 上标 \wedge 表示变量为估算值
- 上标 “ \sim ” 表示变量为变量误差，即观测值与实际测量值之间的差异

3.1.2.1.2 IPMS 的 ESMO 设计

图 3-14 展示了集成到 SMO 中的传统 PLL。

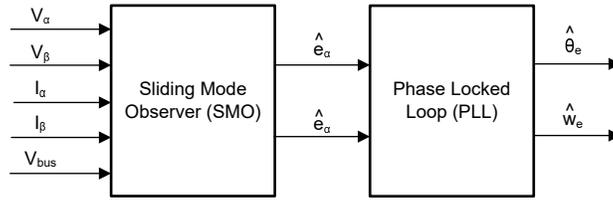


图 3-14. 包含用于 PMSM 的 PLL 的 eSMO 方框图

这里构建了传统的降阶滑模观测器，其数学模型如方程式 25 所示，方框图如图 3-15 所示。

$$\begin{bmatrix} \dot{\hat{i}}_{\alpha} \\ \dot{\hat{i}}_{\beta} \end{bmatrix} = \frac{1}{L_d} \begin{bmatrix} -R_s & -\hat{\omega}_e(L_d - L_q) \\ \hat{\omega}_e(L_d - L_q) & -R_s \end{bmatrix} \begin{bmatrix} \hat{i}_{\alpha} \\ \hat{i}_{\beta} \end{bmatrix} + \frac{1}{L_d} \begin{bmatrix} V_{\alpha} - \hat{e}_{\alpha} + z_{\alpha} \\ V_{\beta} - \hat{e}_{\beta} + z_{\beta} \end{bmatrix} \quad (25)$$

其中

- z_{α} 和 z_{β} 是滑模反馈分量，其定义如方程式 26 所示：

$$\begin{bmatrix} z_{\alpha} \\ z_{\beta} \end{bmatrix} = \begin{bmatrix} k_{\alpha} \text{sign}(\hat{i}_{\alpha} - i_{\alpha}) \\ k_{\beta} \text{sign}(\hat{i}_{\beta} - i_{\beta}) \end{bmatrix} \quad (26)$$

其中

- k_{α} 和 k_{β} 是通过李雅普诺夫稳定性分析设计的恒定滑模增益

如果 k_{α} 和 k_{β} 是足够大的正值，以提供 SMO 的稳定运行，然后 k_{α} 和 k_{β} 足够大，以保持 $k_{\alpha} > \max(|e_{\alpha}|)$ 和 $k_{\beta} > \max(|e_{\beta}|)$ 。

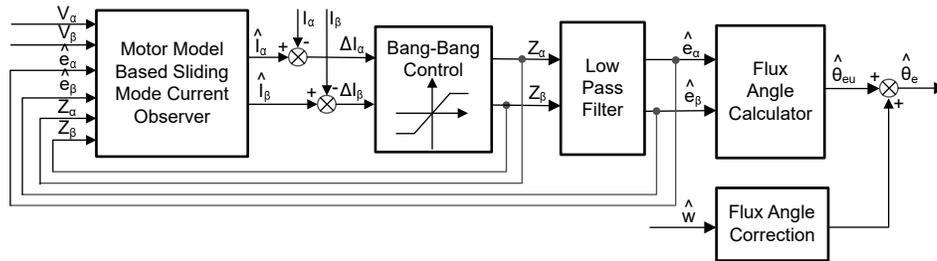


图 3-15. 传统滑模观测器的方框图

α - β 轴上的 EEMF 估算值 (\hat{e}_{α} 、 \hat{e}_{β}) 可通过低通滤波器从不连续开关信号中获得，这些信号为 z_{α} 和 z_{β} ：

$$\begin{bmatrix} \hat{e}_{\alpha} \\ \hat{e}_{\beta} \end{bmatrix} = \frac{\omega_c}{s + \omega_c} \begin{bmatrix} z_{\alpha} \\ z_{\beta} \end{bmatrix} \quad (27)$$

其中

- $\omega_c = 2\pi f_c$ 是 LPF 的截止角频率，通常根据定子电流的基频来选择该截止角频率

因此，转子位置可以直接通过反电动势的反正切计算得出，其定义如[方程式 28](#)所示：

$$\hat{\theta}_e = -\tan^{-1}\left(\frac{\hat{e}_\alpha}{\hat{e}_\beta}\right) \quad (28)$$

低通滤波器消除了滑模函数的高频项，从而导致出现相位延迟。可以通过截止频率 ω_c 和反电动势频率 ω_e 之间的关系对其进行补偿，定义如[方程式 29](#)所示：

$$\Delta\theta_e = -\tan^{-1}\left(\frac{\omega_e}{\omega_c}\right) \quad (29)$$

这样使用 SMO 方法估算的转子位置就如[方程式 30](#)所示：

$$\hat{\theta}_e = -\tan^{-1}\left(\frac{\hat{e}_\alpha}{\hat{e}_\beta}\right) + \Delta\theta_e \quad (30)$$

在数字控制应用中，需要使用 SMO 的时间离散方程。欧拉法是变换为时间离散观测器的合适方法。在 α - β 坐标中，[方程式 25](#) 的时间离散系统矩阵由[方程式 31](#)给出：

$$\begin{bmatrix} \hat{i}_\alpha(n+1) \\ \hat{i}_\beta(n+1) \end{bmatrix} = \begin{bmatrix} F_\alpha \\ F_\beta \end{bmatrix} \begin{bmatrix} \hat{i}_\alpha(n) \\ \hat{i}_\beta(n) \end{bmatrix} + \begin{bmatrix} G_\alpha \\ G_\beta \end{bmatrix} \begin{bmatrix} V_\alpha^*(n) - \hat{e}_\alpha(n) + z_\alpha(n) \\ V_\beta^*(n) - \hat{e}_\beta(n) + z_\beta(n) \end{bmatrix} \quad (31)$$

其中

- 矩阵 [F] 和 [G] 由[方程式 32](#)和[方程式 33](#)给出：

$$\begin{bmatrix} F_\alpha \\ F_\beta \end{bmatrix} = \begin{bmatrix} e^{-\frac{R_s}{L_d}} \\ e^{-\frac{R_s}{L_q}} \end{bmatrix} \quad (32)$$

$$\begin{bmatrix} G_\alpha \\ G_\beta \end{bmatrix} = \frac{1}{R_s} \begin{bmatrix} 1 - e^{-\frac{R_s}{L_d}} \\ 1 - e^{-\frac{R_s}{L_q}} \end{bmatrix} \quad (33)$$

[方程式 27](#) 的时间离散形式由[方程式 34](#)给出：

$$\begin{bmatrix} \hat{e}_\alpha(n+1) \\ \hat{e}_\beta(n+1) \end{bmatrix} = \begin{bmatrix} \hat{e}_\alpha(n) \\ \hat{e}_\beta(n) \end{bmatrix} + 2\pi f_c \begin{bmatrix} z_\alpha(n) - \hat{e}_\alpha(n) \\ z_\beta(n) - \hat{e}_\beta(n) \end{bmatrix} \quad (34)$$

3.1.2.1.2.1 使用 PLL 的转子位置和转速估算

在反正切法中，由于噪声和谐波分量的存在，位置和转速估算的精度会受到影响。为了消除该问题，可使用 PLL 模型对 IPMSM 的无传感器控制结构中的转速和位置进行估算。[节 3.1.2.1.2](#) 说明了与 SMO 配合使用的 PLL 结构。反电动势估算 \hat{e}_α 和 \hat{e}_β 可与 PLL 模型配合使用来估算电机角速度和位置，如[图 3-16](#)所示。

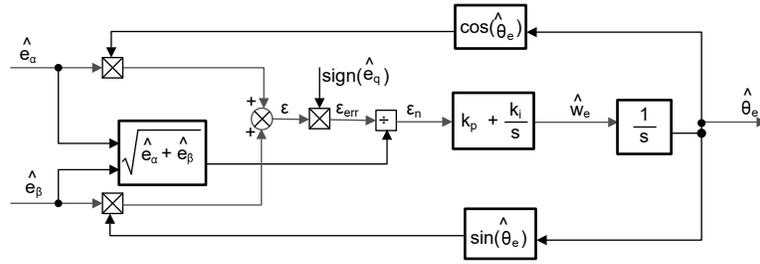


图 3-16. 锁相环位置跟踪器的方框图

由于 $e_\alpha = E \cos(\theta_e)$ 、 $e_\beta = E \sin(\theta_e)$ 和 $E = \omega_e \lambda_{pm}$ ，位置误差定义如方程式 35 所示：

$$\varepsilon = \hat{e}_\beta \cos(\hat{\theta}_e) - \hat{e}_\alpha \sin(\hat{\theta}_e) = E \sin(\theta_e) \cos(\hat{\theta}_e) - E \cos(\theta_e) \sin(\hat{\theta}_e) = E \sin(\theta_e - \hat{\theta}_e) \quad (35)$$

其中

- E 是 EEMF 的幅度，与电机转速成正比 ω_e

当 $(\theta_e - \hat{\theta}_e) < \frac{\pi}{2}$ 时，方程式 35 可以简化为方程式 36。

$$\varepsilon = E(\theta_e - \hat{\theta}_e) \quad (36)$$

可以进一步得到 EEMF 归一化后的位置误差 (方程式 37)：

$$\varepsilon_n = \theta_e - \hat{\theta}_e \quad (37)$$

根据分析，可以得到正交锁相环位置跟踪器的简化方框图，如图 3-17 所示。PLL 的闭环传递函数可表示为方程式 38：

$$\frac{\hat{\theta}_e}{\theta_e} = \frac{k_p s + k_i}{s^2 + k_p s + k_i} = \frac{2\xi \omega_n s + \omega_n^2}{s^2 + 2\xi \omega_n s + \omega_n^2} \quad (38)$$

其中

- k_p 和 k_i 是标准 PI 调节器的比例增益和积分增益

固有频率 ω_n 和阻尼比 ξ 由方程式 39 给出：

$$k_p = 2\xi \omega_n, \quad k_i = \omega_n^2 \quad (39)$$

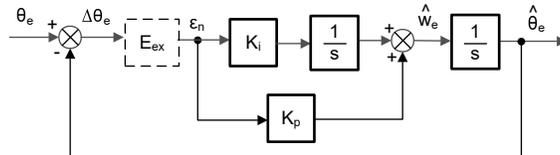


图 3-17. 锁相环位置跟踪器的简化方框图

3.1.3 弱磁 (FW) 和每安培最大扭矩 (MTPA) 控制

永磁同步电机 (PMSM) 因其高功率密度、高效率 and 宽转速范围而广泛应用于家用电器应用。PMSM 包含两种主要类型：表面贴装式 PMSM (SPM) 和内嵌式 PMSM (IPM)。由于 SPM 电机在扭矩和 q 轴电流之间具有线性关系，因此更易于控制。不过，IPMSM 由于凸极比大而具有电磁扭矩和磁阻扭矩。总扭矩相对于转子角度是非线性的。因此，MTPA 技术可用于 IPM 电机，以优化恒定扭矩区域中的扭矩生成。弱磁控制的目的是优化以达到 PMSM 驱动器的最高功率和效率。弱磁控制可以使电机以其基本转速运行，扩大其运行限值以使转速高于额定转速，并允许在整个转速和电压范围内实现最佳控制。

IPMSM 数学模型的电压公式可以用 d - q 坐标来描述，如[方程式 40](#) 和[方程式 41](#) 所示。

$$v_d = L_d \frac{di_d}{dt} + R_s i_d - p \omega_m L_q i_q \quad (40)$$

$$v_q = L_q \frac{di_q}{dt} + R_s i_q + p \omega_m L_d i_d + p \omega_m \psi_m \quad (41)$$

[图 3-18](#) 展示了 IPM 同步电机的动态等效电路。

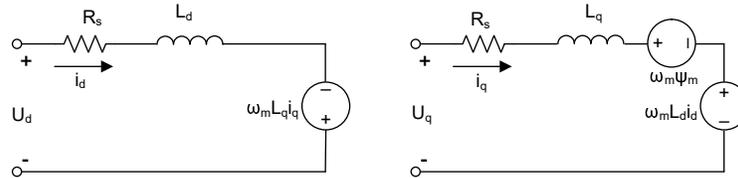


图 3-18. IPM 同步电机的等效电路

IPMSM 产生的总电磁扭矩可以由[方程式 42](#) 表示，产生的扭矩包含两个不同的项。第一项对应于扭矩电流 i_q 和永磁体 ψ_m 之间产生的相互反作用力扭矩，而第二项对应于由于 d 轴和 q 轴上的电感不同而产生的磁阻扭矩。

$$T_e = \frac{3}{2} p [\psi_m i_q + (L_d - L_q) i_d i_q] \quad (42)$$

在大多数应用中，IPMSM 驱动器具有转速和扭矩约束，这主要是由于分别存在逆变器或电机额定电流以及可用的直流链路电压限制。这些约束可以用数学公式[方程式 43](#) 和[方程式 44](#) 进行表示。

$$I_a = \sqrt{i_d^2 + i_q^2} \leq I_{\max} \quad (43)$$

$$V_a = \sqrt{v_d^2 + v_q^2} \leq V_{\max} \quad (44)$$

其中

- V_{\max} 和 I_{\max} 是逆变器或电机允许的最大电压和电流

在两级三相电压源逆变器 (VSI) 供电的电机中，可实现的最大相电压受直流链路电压和 PWM 策略的限制。如果采用空间矢量调制 (SVPWM)，则最大电压限制为[方程式 45](#) 中所示的值。

$$\sqrt{v_d^2 + v_q^2} \leq v_{\max} = \frac{v_{dc}}{\sqrt{3}} \quad (45)$$

通常，定子电阻 R_s 在高速运行时可以忽略不计，并且电流的导数在稳态下为零，因此得到[方程式 46](#)，如其所示。

$$\sqrt{L_d^2 \left(i_d + \frac{\psi_{pm}}{L_d} \right)^2 + L_q^2 i_q^2} \leq \frac{V_{\max}}{\omega_m} \quad (46)$$

[方程式 43](#) 的电流限制在 d - q 平面中产生一个半径为 I_{\max} 的圆，而[方程式 44](#) 的电压限制产生一个椭圆，其半径 V_{\max} 随着转速的增加而减小。必须对得到的 d - q 平面电流矢量进行控制，使其同时遵守电流和电压约束。根据这些约束，可以区分 IPMSM 的三个工作区域，如[图 3-19](#) 所示。

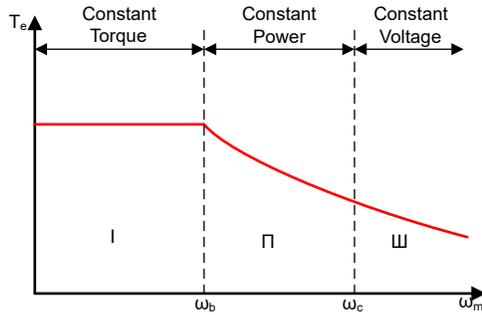


图 3-19. IPMSM 控制工作区域

1. 恒定扭矩区域：可以在该工作区域内实施 MTPA，从而可产生最大扭矩。
2. 恒定功率区域：必须采用弱磁控制，并且在达到电流约束时减小扭矩容量。
3. 恒定电压区域：在这个工作区域，深度弱磁控制使定子电压保持恒定，以尽可能大地产生扭矩。

在恒定扭矩区域，根据方程式 42，IPMSM 的总扭矩包括来自磁链的电磁扭矩和来自以下电感之间凸极的磁阻扭矩： L_d 和 L_q 。电磁扭矩与 q 轴电流 i_q 成正比，磁阻扭矩与 d 轴电流 i_d 、 q 轴电流 i_q 以及 L_d 和 L_q 。

SPM 电机的传统矢量控制系统仅通过将命令的 i_d 设置为零来实现非弱磁模式，从而利用电磁扭矩。但是，虽然 IPMSM 利用电机的磁阻扭矩，设计人员还必须控制 d 轴电流。MTPA 控制的目的是计算基准电流 i_d 和 i_q 以尽可能增大产生的电磁扭矩与磁阻扭矩之间的比率。以下各公式显示了 i_d 和 i_q 之间的关系以及定子电流 I_s 的矢量和。

$$I_s = \sqrt{i_d^2 + i_q^2} \quad (47)$$

$$I_d = I_s \cos \beta \quad (48)$$

$$I_q = I_s \sin \beta \quad (49)$$

其中

- β 是同步 (d-q) 坐标系中的定子电流角度

方程式 42 可以表示为方程式 50，其中 I_s 替换了 i_d 和 i_q 。

方程式 50 表明电机扭矩取决于定子电流矢量的角度：

$$T_e = \frac{3}{2} p I_s \sin \beta [\psi_m + (L_d - L_q) I_s \cos \beta] \quad (50)$$

该公式显示，当电机扭矩微分等于零时，可以计算出最大效率点。当该微分 $\frac{dT_e}{d\beta}$ 为零，如方程式 51 所示。

$$\frac{dT_e}{d\beta} = \frac{3}{2} p [\psi_m I_s \cos \beta + (L_d - L_q) I_s^2 \cos 2\beta] = 0 \quad (51)$$

根据这一公式，可以得出 MTPA 控制的电流角度，如方程式 52 所示。

$$\beta_{\text{mtpa}} = \cos^{-1} \frac{-\psi_m + \sqrt{\psi_m^2 + 8 \times (L_d - L_q)^2 \times I_s^2}}{4 \times (L_d - L_q) \times I_s} \quad (52)$$

因此，可以使用 MTPA 控制的电流角度通过方程式 53 和方程式 54 来表示有效的 d 轴和 q 轴基准电流。

$$I_d = I_s \times \cos \beta_{\text{mtpa}} \quad (53)$$

$$I_q = I_s \times \sin \beta_{\text{mtpa}} \quad (54)$$

不过，如**方程式 52** 所示，MTPA 控制的角度 β_{mtpa} 与 d 轴和 q 轴电感有关。这意味着电感的变化会阻碍找到出色的 MTPA 点。为了提高电机驱动器的效率，在线估算 d 轴和 q 轴电感，但参数 L_d 和 L_q 不易于在线测量，并且受饱和和效应的影响。稳健的查询表 (LUT) 方法可确保电气参数变化下的可控性。通常，为了简化数学模型，可以忽略 d 轴和 q 轴电感之间的耦合效应。因此，假设 L_d 仅随 i_d 而变化， L_q 仅随 i_q 而变化。因此，d 轴和 q 轴电感可以分别建模为 d-q 电流的函数，如**方程式 55** 和**方程式 56** 所示。

$$L_d = f_1(i_d, i_q) = f_1(i_d) \quad (55)$$

$$L_q = f_2(i_q, i_d) = f_2(i_q) \quad (56)$$

通过简化**方程式 52** 可以减轻 ISR 计算负担。基于电机参数的常数 K_{mtpa} 改为用**方程式 57** 表示，其中 K_{mtpa} 在后台循环中使用更新的 L_d 和 L_q 。

$$K_{\text{mtpa}} = \frac{\psi_m}{4 \times (L_q - L_d)} = 0.25 \times \frac{\psi_m}{(L_q - L_d)} \quad (57)$$

$$\beta_{\text{mtpa}} = \cos^{-1} \left(K_{\text{mtpa}} / I_s - \sqrt{(K_{\text{mtpa}} / I_s)^2 + 0.5} \right) \quad (58)$$

第二个中间变量 G_{mtpa} (如**方程式 59** 中的描述) 进行了定义，以便进一步简化计算。使用 G_{mtpa} ，MTPA 控制的角度 β_{mtpa} 可以按照**方程式 60** 进行计算。这两个计算在 ISR 中执行，以获得真实的电流角度 β_{mtpa} 。

$$G_{\text{mtpa}} = K_{\text{mtpa}} / I_s \quad (59)$$

$$\beta_{\text{mtpa}} = \cos^{-1} \left(G_{\text{mtpa}} - \sqrt{G_{\text{mtpa}}^2 + 0.5} \right) \quad (60)$$

在所有情况下，都可以通过作用于直轴电流 i_d 。作为进入该恒定功率工作区域的结果，选择弱磁控制而不是在恒定功率和电压区域中使用的 MTPA 控制。由于最大逆变器电压受到限制，PMSM 电机无法在反电动势 (几乎与永磁场和电机转速成正比) 高于逆变器最大输出电压的转速区域中运行。在 PM 电机中，无法直接控制磁通量。不过，通过添加负 i_d 。考虑到电压和电流约束，电枢电流和端子电压会受到限制，如**方程式 43** 和**方程式 44** 所示。逆变器输入电压 (直流链路电压) 的变化限制了电机的最大输出。此外，最大基波电机电压还取决于所使用的 PWM 方法。在**方程式 46** 中，IPMSM 有两个因素：一个是永磁值，另一个是电感和磁通电流。

图 3-20 展示了用于实现弱磁的典型控制结构。 β_{fw} 是弱磁 (FW) PI 控制器的输出，可生成基准 i_d 和 i_q 。在电压幅度达到限制之前，FW 的 PI 控制器的输入始终为正，因此输出始终在 0 处达到饱和。

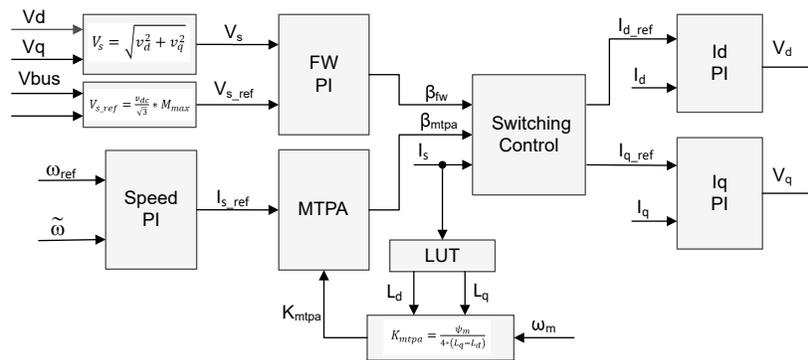


图 3-20. 弱磁和每安培最大扭矩控制的方框图

图 3-9 和**图 3-11** 显示了基于 FAST 或 eSMO 的 FOC 实现的方框图。这些方框图概述了 FOC 系统的功能和变量。电机驱动 FOC 系统中有两个控制模块：一个是 MTPA 控制，一个是弱磁控制。这两个模块根据输入参数分别生成电流角度 β_{mtpa} 和 β_{fw} ，如**图 3-21** 所示。

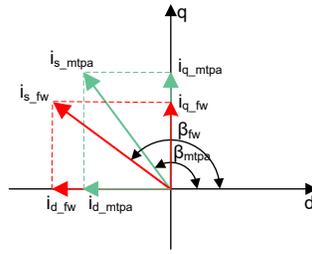


图 3-21. FW 和 MTPA 期间 IPMSM 的电流相量图

切换控制模块用于决定应用哪个角度，然后计算基准 i_d 和 i_q ，如[方程式 48](#)和[方程式 54](#)所示。可以根据以下公式来选择电流角度：[方程式 61](#)和[方程式 62](#)。

$$\beta = \beta_{fw} \text{ if } \beta_{fw} > \beta_{mtpa} \tag{61}$$

$$\beta = \beta_{mtpa} \text{ if } \beta_{fw} < \beta_{mtpa} \tag{62}$$

图 3-22 中的流程图展示了在主循环和中断中运行采用 FW 和 MPTA 的 InstaSPIN™-FOC 时所需的步骤。

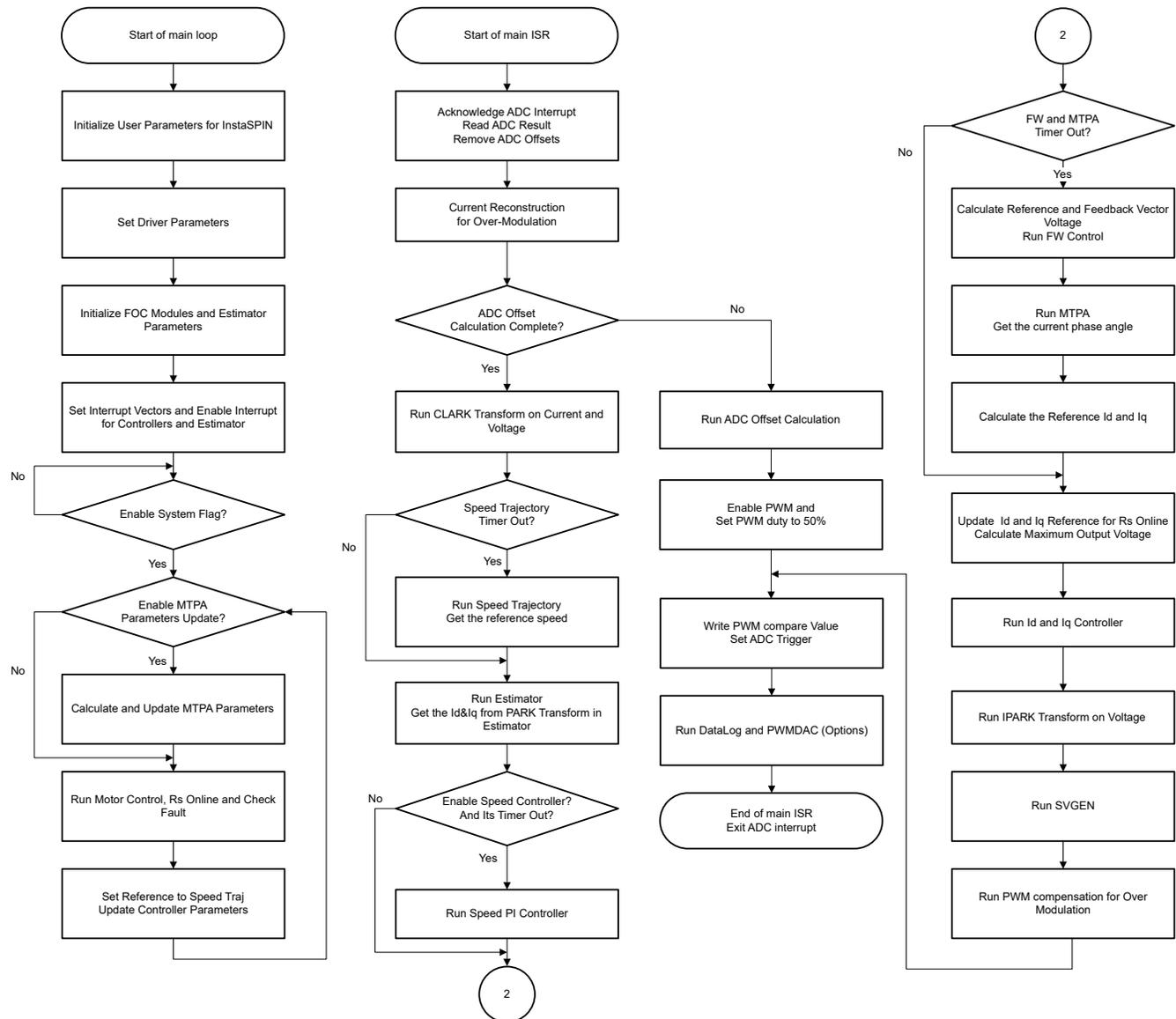


图 3-22. 采用 FW 和 MTPA 的 InstaSPIN-FOC 工程的流程图

3.2 软件入门

3.2.1 GUI

有关深入介绍 MCU 电机控制 GUI 的入门指南，请参阅[此文档](#)。

提供了该 EVM 源代码，以便直接调试固件。然而，软件调试通常是一个漫长的过程。为了缩短开发时间，该参考设计提供了一个基于 UART 的 GUI 软件，以帮助快速调整任何定制应用的参数。

请注意，该设计旨在让 GUI 通过大多数电路板版本上存在的 XDS110ISO-EVM UART 连接或隔离式 UART 连接 J14 进行连接。如果改为直接连接到子板上可用的 UART 引脚，则**接地端为 HOT**，在将接地设备连接到套件时必须使用外部隔离器。

3.2.2 下载并安装 C2000 软件

1. 从 [Code Composer Studio \(CCS\) 集成开发环境 \(IDE\)](#) 工具文件夹下载 Code Composer Studio™ IDE 并进行安装。建议使用版本 12.5 或更高版本。

2. 下载并安装 [C2000WARE-MOTORCONTROL-SDK](#)。建议使用版本 v5.01.00.00 或更高版本。
3. 安装完成后，打开 CCS 并创建一个新的工作区以导入工程。

备注

该 EVM 支持 SysConfig，能够在易于使用的图形界面中配置器件引脚并初始化器件外设。该功能仅供当前版本参考。有关详细的实现信息，请参阅 [C2000 SysConfig 软件指南](#)。

3.2.3 使用软件

F280013x 子板的工程文件夹在 C2000Ware 电机控制 SDK 中，位于：
<install_location>\solutions\tida_010265_wminv\f280013x。按照以下步骤使用不同的增量构建来构建和运行该代码。

1. 点击 *Project* → *Import CCS Projects*。点击浏览并导航至子板控制器的工程文件夹。
2. 选择正确的构建配置，如图 3-23 所示。
 - a. 使用默认的 *Flash_MtrInv_3SC* 构建配置进行三分流器电流检测。
 - b. *Flash_MtrInv_1SC* 支持单分流器电流检测。
3. 通过选择正确的预定义符号，配置工程以启用和禁用辅助功能，如图 3-24 所示。

如果使用电路板版本 E1，必须添加 *TIEVM_MTR-HVINV_REV_E1* 作为预定义符号。如果没有这个符号，电路板将无法按预期运行。

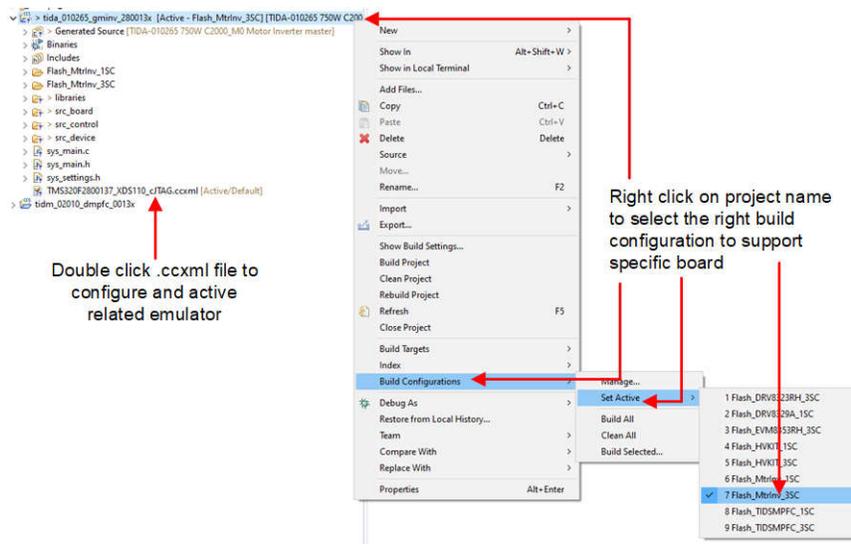


图 3-23. 选择合适的构建配置

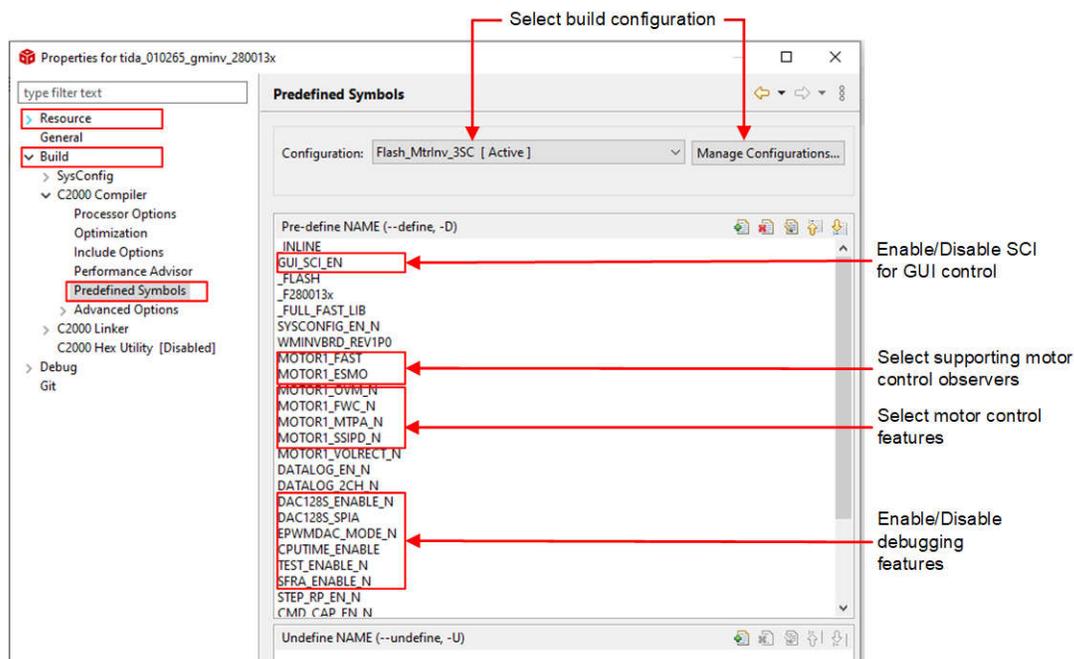


图 3-24. 在工程属性中选择合适的预定义符号

3.2.4 工程结构

导入工程后，CCS 内将显示 Project Explorer，如图 3-25 所示。

文件夹 `src_board` 中包含器件外设配置。此配置通过 C2000Ware 中的 `driverlib` 完成。MCU 配置通常位于 **Hardware (硬件) Access (访问) Layer (层) (HAL)** 文件中，即 `hal.c` 和 `hal.h`。此文件夹还包含 `user_mtr1.c`，其中会计算或分配各种参数。在大多数用例中，用户不需要调整此文件中完全定义或派生的任何参数。

`src_control` 文件夹包含电机驱动文件，这些文件在中断服务例程和后台任务中调用电机控制核心算法函数。此文件夹还包含各种辅助功能，如 SFRA、GUI 接口等。

文件夹 `src_control/common/include` 包含对系统有用的各种头文件。其中最值得注意的是 `user_mtr1.h`，它包含电机和系统参数定义。

文件夹 `src_device` 包含由 HAL 使用的 `driverlib`。

文件夹 `src_sta` 包含自检应用代码。

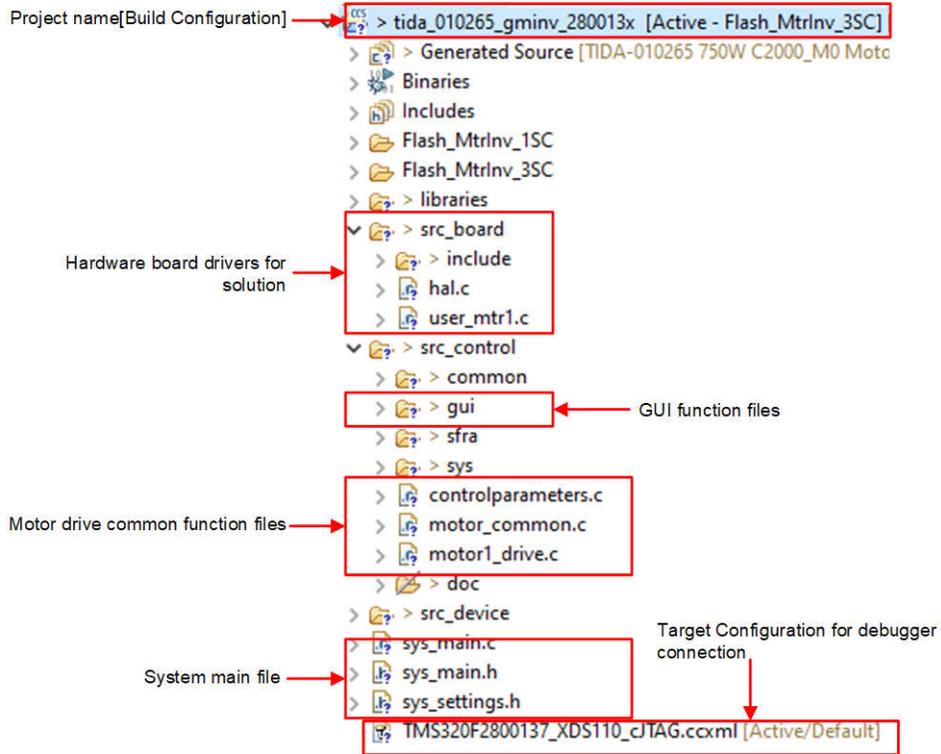


图 3-25. TIEVM-MTR-HVINV 项目资源管理器视图

图 3-26 显示了电机控制的 ISR 工程软件流程图，主循环用于在后台循环中更新电机控制参数。

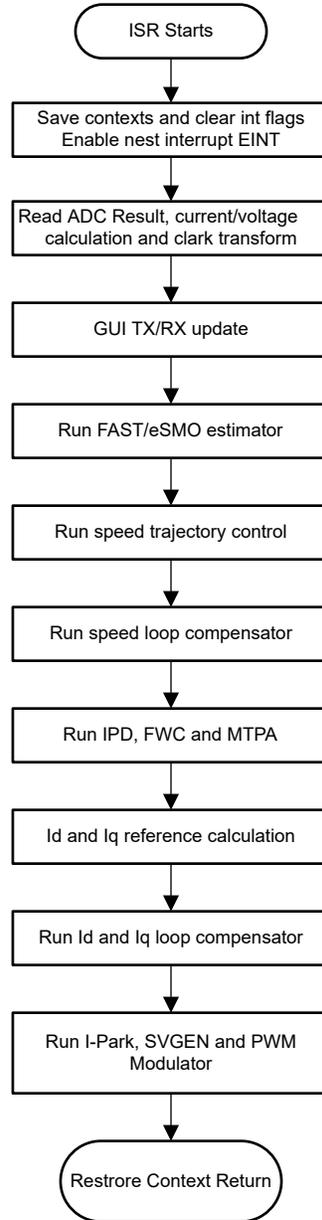


图 3-26. 固件工程流程图

该工程包含一个电机控制中断服务例程，每个 PWM 周期都会调用该例程。一些后台任务以一直循环的方式在 `main()` 进行调用，可用于运行不要求绝对计时精度的缓慢任务（如电机控制参数更新等）。CPU 计时器用于触发慢速后台任务。

`motor1CtrlISR` 被保留，用于调用电机驱动控制算法以旋转电机 1，该电机以用户定义的速率 `USER_M1_ISR_FREQ_Hz` 定期触发。

为了简化系统，该 EVM 的软件设计分为四个具有增量构建 (`DMC_BUILDLEVEL`) 的实验室，这使得学习和熟悉电路板和软件变得更加容易。这个方法对也适用于调试和测试电路板。表 3-1 列出了详细的增量构建选项。要选择特定的构建选项，请在 `sys_settings.h` 中选择相应的 `BUILDLEVEL` 选项。选择构建选项后，通过选择 `rebuild all` 编译器选项来编译工程。节 4.5 提供了有关运行每个构建级别选项的更多详细信息。

表 3-1. 递增构建选项

操作	构建选项	说明
电机驱动	DMC_LEVEL_1	50% PWM 占空比，验证 ADC 失调电压校准、PWM 输出和相移
	DMC_LEVEL_2	开环 v/f 控制，用于检查电机的电流和电压检测信号
	DMC_LEVEL_3	闭合电流环路，用于检查硬件设置
	DMC_LEVEL_4	电机参数识别，使用 InstaSPIN-FOC 或 eSMO 运行

4 测试过程和结果

4.1 构建级别 1：CPU 和电路板设置

了解该构建级别中的目标：

- 评估系统的开环运行
- 使用 HAL 对象设置 MCU 控制器并初始化逆变器
- 验证 PWM 和 ADC 驱动器模块
- 熟悉 CCS 的操作

由于该系统以开环控制方式运行，因此 ADC 测量值只用于该构建级别的仪器用途。该构建级别仅使用 MCU 控制器和栅极驱动器的偏置电源。逆变器不提供高电压交流和直流电源。

在该构建级别中，电路板以开环方式执行（采用固定占空比）。电机的占空比设置为 50%。该构建级别验证来自功率级的反馈值检测以及 PWM 栅极驱动器的运行，并确保没有硬件问题。此外，可以在该构建级别中执行输入和输出电压检测校准。图 4-1 展示了该构建级别的软件流程。

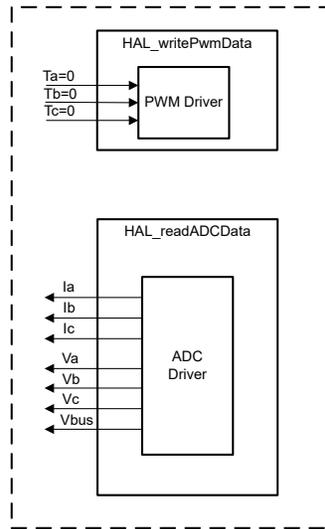


图 4-1. 控制软件方框图：构建级别 1 - 偏移验证

4.2 构建级别 2：带 ADC 反馈的开环检查

了解该构建级别中的目标：

- 实现简单的电机标量 v/f 控制以驱动电机，从而验证电流和电压检测电路以及 IPM 电路。
- 测试用于电机控制的 InstaSPIN-FOC FAST 或 eSMO 模块。

由于该系统以开环控制方式运行，因此 ADC 测量值只用于该构建级别的仪器用途。逆变器提供高电压直流电源，辅助电源模块提供 MCU 控制器和 IPM 的偏置电源。图 4-2 展示了该构建级别的软件流程图。

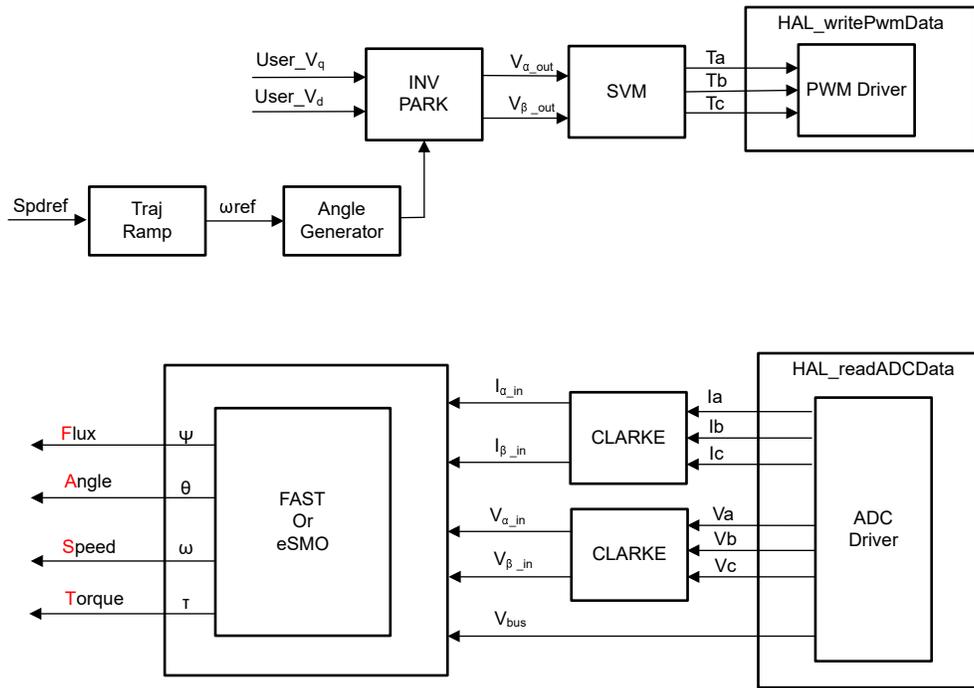


图 4-2. 控制软件方框图：构建级别 2 - 开环控制

4.3 构建级别 3：闭合电流环路检查

了解该构建级别中的目标：

- 评估电机运行的闭合电流环路。

在这个构建级别，电机由 *if* 控制进行控制，转子角度由斜坡发生器模块生成。图 4-3 展示了该构建级别的软件流程图。

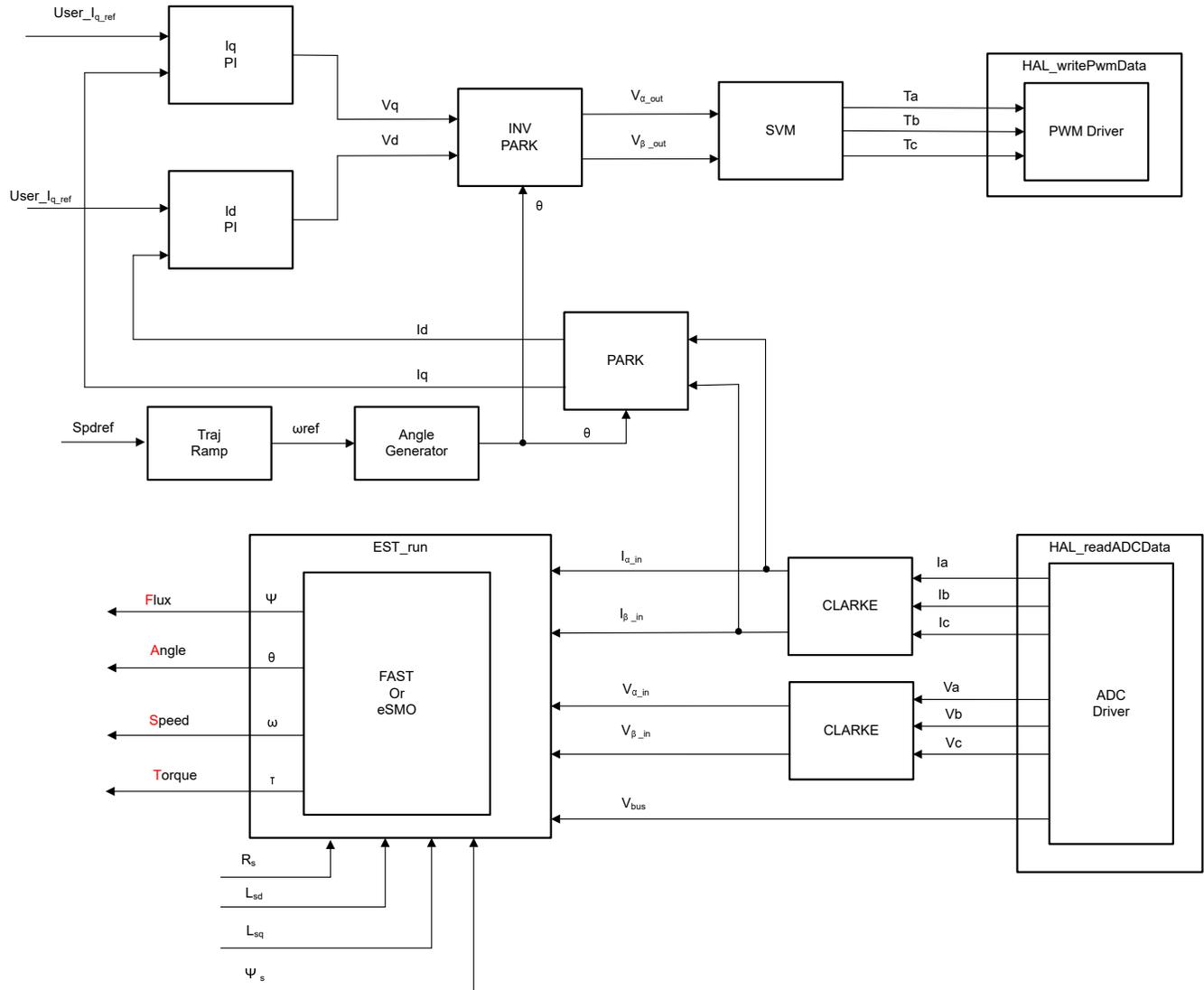


图 4-3. 控制软件方框图：构建级别 3 - 电流闭环控制

4.4 构建级别 4：完整电机驱动控制

了解该构建级别中的目标：

- 评估完整的电机驱动
- 评估其他功能，比如电机弱磁控制
- 评估完成的系统

在此构建级别，外部速度环路是闭合的，内部电流环路用于电机，使得转子角度来自 FAST 或 eSMO 估算器模块。图 4-4 展示了该构建级别的软件流程图。

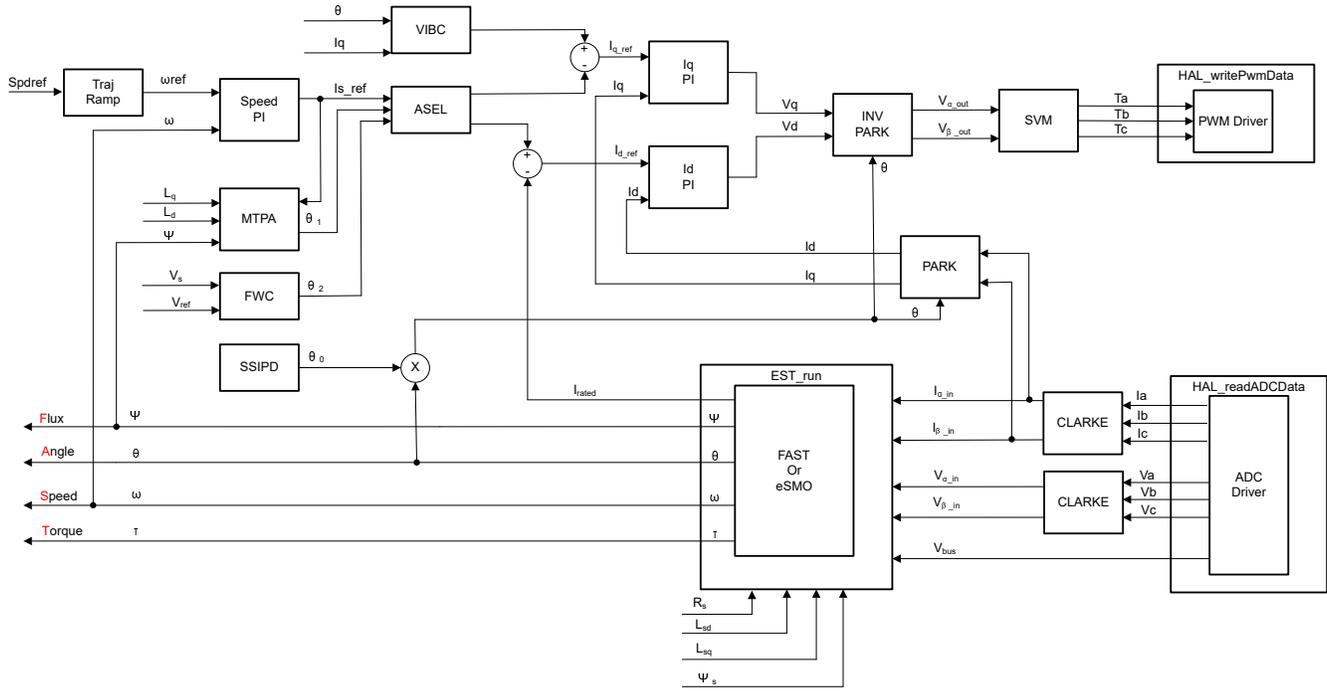


图 4-4. 控制软件方框图：构建级别 4 - 转速和电流闭环控制

4.5 测试程序

警告

电路板上存在**高压**。要安全地评估该板，请使用适当的隔离和限流电源。在向板施加电源之前，必须在输出端连接适当的阻性负载或电子负载。请勿在通电时操作设备。仅使用具有适当额定值的设备并遵循适当的隔离和安全措施。

小心

将示波器和其他测试设备连接到电路板时要小心，因为交流整流器会产生直流输出电压，该电压具有相对于保护性接地进行浮动的**热接地**。在将接地设备连接到套件时，必须使用隔离变压器。

本节假设已经完成节 3.2 和节 2.2 中列出的所有必要设置步骤。如果尚未如此，请立即完成。

此外，某些电路板版本可能需要执行额外的步骤。如果尚未完成，请在开始执行任何测试之前先参阅节 6.1。

4.5.1 启动

要启动工程，请完成以下步骤：

1. 通过 PC 连接到 XDS110ISO-EVM。
2. 将电机连接到 J10。
3. 将一个能够提供高达 750W 通用交流输入的可编程隔离式交流电源连接至 EVM 的输入端子（连接器 J5），如图 4-5 所示。将交流电源电流限制设置为 8A。此时**不要**打开电源。

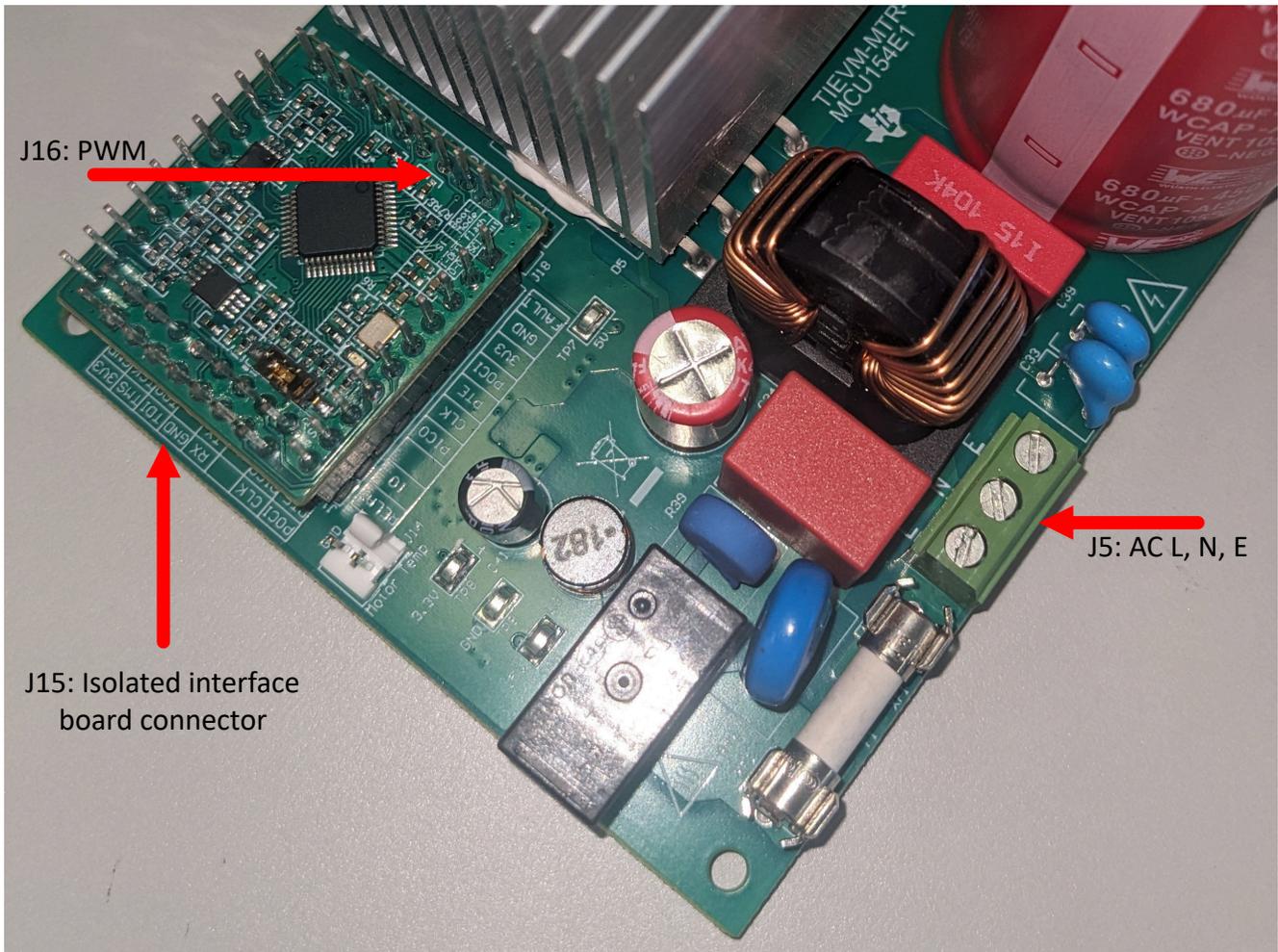


图 4-5. 连接外部交流或直流电源以验证硬件

4.5.2 构建和加载工程

要构建和加载工程，请完成以下步骤：

1. 右键点击工程名称，点击 *Properties* 命令，转到预定义的符号，以将 `GUI_SCI_EN` 更改为 `GUI_SCI_N`，从而为 GUI 禁用 SCI 功能，如图 3-24 所述。
2. 打开 `sys_settings.h` 文件，将 `DMC_BUILDLEVEL` 设置为所选的构建级别。表 3-1 中对构建级别进行了说明。
 - a. `DMC_LEVEL_1`
 - b. `DMC_LEVEL_2`
 - c. `DMC_LEVEL_3`
 - d. `DMC_LEVEL_4`
3. 构建级别 2、3 和 4 要求确切知道电机物理参数。
 - a. 必须在头文件 (`user_mtr1.h`) 中记录所需的电机参数，如以下示例代码所示。电机参数并不仅限于此，这里节选了一部分用作示例。

```
#define USER_MOTOR1_Rs_Ohm           (2.68207002f)
#define USER_MOTOR1_Ls_d_H          (0.00926135667f)
#define USER_MOTOR1_Ls_q_H          (0.00926135667f)
#define USER_MOTOR1_RATED_FLUX_Vphz (0.381890297f)
```

- b. 如果不确切知道上面列出的四个特定电机参数，可以在构建级别 4 中使用 FAST 电机识别，以便确定使用 FAST 估算器时的电机参数。执行以下操作，然后参考构建级别 4 说明以获取更多信息。

- i. 在 `main()` 中，将 `userParams_M1.flag_bypassMotorId` 值更改为“false”以启用电机识别，如下示例代码所示。

```
// true->enable identification, false->disable identification
userParams[MTR_1].flag_bypassMotorId = false;
```

根据电机规范（如果已知），在 `user_mtr1.h` 标头文件中将以下识别参数设置为适当的值。

```
#define USER_MOTOR1_RES_EST_CURRENT_A      (1.0f)      // A - 10~30% of rated current
of the motor
#define USER_MOTOR1_IND_EST_CURRENT_A      (-1.0f)     // A - 10~30% of rated current
of the motor, just enough to enable rotation
#define USER_MOTOR1_MAX_CURRENT_A         (6.5f)       // A - 30~150% of rated current
of the motor
#define USER_MOTOR1_FLUX_EXC_FREQ_HZ      (40.0f)     // Hz - 10~30% of rated
frequency of the motor
```

4. 如果之前构建了另一个构建选项，则右键单击工程名称并点击 **Clean Project**，然后点击 **Build Project**。观察构建窗口中运行的工具。等待工程构建成功。
5. 在 **Project Explorer** 窗口中，右键单击 `.ccxml` 文件，然后选择“Set as Active Target Configuration”
6. 打开交流或直流电源，向 J5 施加 30VAC 或 40VDC，从而为控制器和栅极驱动器产生 +15V 和 +3.3V 的电压。点击“Debug”按钮  或点击 **Run** → **Debug**。可以编译所选构建级别的代码，并将其加载到 C2000 器件上。请注意右上角的“CCS Debug”图标，该图标表明用户现在处于 **Debug Perspective** 视图中。启动后，程序会在 `main()` 开始位置停止。

4.5.3 设置调试环境窗口

在调试代码时观察本地和全局变量是一种标准的调试做法。在 CCS 中有多种不同的方法来实现这一做法，例如存储器视图和监视视图。此外，CCS 能够制作时域（和频域）图。该功能允许用户使用图形工具查看波形。CCS 中的主要调试器工具是 Expressions 窗口。

1. 点击菜单栏上的 **View** → **Expressions**，打开一个 **Expressions** 观察窗口。
2. 可以将一组变量导入到 **Expressions** 窗口中，方法是右键单击 **Expressions** 窗口并点击 **Import**，浏览至工程目录 `<install_location>\solutions\tida_010265_wminv\src_control\common\debug`。有 5 种不同的 `.txt` 文件可用。选择与所选构建级别对应的文件，然后点击 **OK** 按钮将变量导入到 **Expressions** 窗口中。

备注

此时主代码中的某些变量尚未初始化，可能包含一些无用的值。

3. 或者，也可以在 **Expressions** 窗口中手动添加和删除变量。
4. 结构变量 `motorVars_M1` 引用了大多数与控制电机相关的变量。通过展开该变量，您可以查看所有这些变量并根据需要进行编辑。
5. 点击“Expressions”窗口中的 **Continuous Refresh** 按钮 。这将启用窗口的实时运行模式。通过点击该 **Expressions** 窗口中的下拉箭头，您可以选择 **Customize Continuous Refresh Interval** 并编辑该“Expressions”窗口的刷新率。选择过短的刷新间隔可能会影响性能。

4.5.4 运行代码

要运行工程代码，请完成以下步骤：

1. 将交流电源的输出电压从 30VAC 缓慢调整为与构建相关的测试值。
 - a. 构建 1：请勿调整输出电压。
 - b. 构建 2：100-VAC
 - c. 构建 3 和构建 4：220-VAC
2. 通过点击按钮  来运行工程，或点击 **Debug** 选项卡中的 **Run** → **Resume**。
3. 在 **Expressions** 窗口中，等待 `systemVars.flagEnableSystem` 自动设置为“1”。将 `motorVars_M1.flagEnableRunAndIdentify` 设置为“1”。
4. 在 **Expressions** 窗口中，变量 `motorVars_M1.flagRunIdentAndOnLine` 会自动设置为“1”。`ISRCCount` 会不断增加。

5. 该工程现在已可以运行，并且 *Expressions* 窗口中的值会持续更新。根据用户的偏好，可以调整所有 CCS 窗口的大小。
6. 完成与构建相关的测试程序。
 - a. 节 4.5.4.1
 - b. 节 4.5.4.2
 - c. 节 4.5.4.3
 - d. 节 4.5.4.4
7. 通过首先点击工具栏上的 *Halt* 按钮  或点击 *Target* → *Halt* 来完全停止控制器。最后，通过点击  或点击 *Run* → *Reset* 来重置控制器。
8. 为下一个构建级别擦除控制器中的代码，方法是点击 *Tools* → *On-Chip Flash*，然后点击 *On-Chip Flash* 选项卡中的 *Erase Flash* (确保选中所有闪存组)，如图 4-6 所示。该操作会擦除闪存中存储的所有程序代码。(这一步为可选步骤)。

备注

在擦除闪存时请勿点击 *Cancel*、关闭板的电源或断开仿真器。

9. 通过点击 *Terminate Debug Session* 按钮  或点击 *Run* → *Terminate*，关闭 CCS 调试会话。

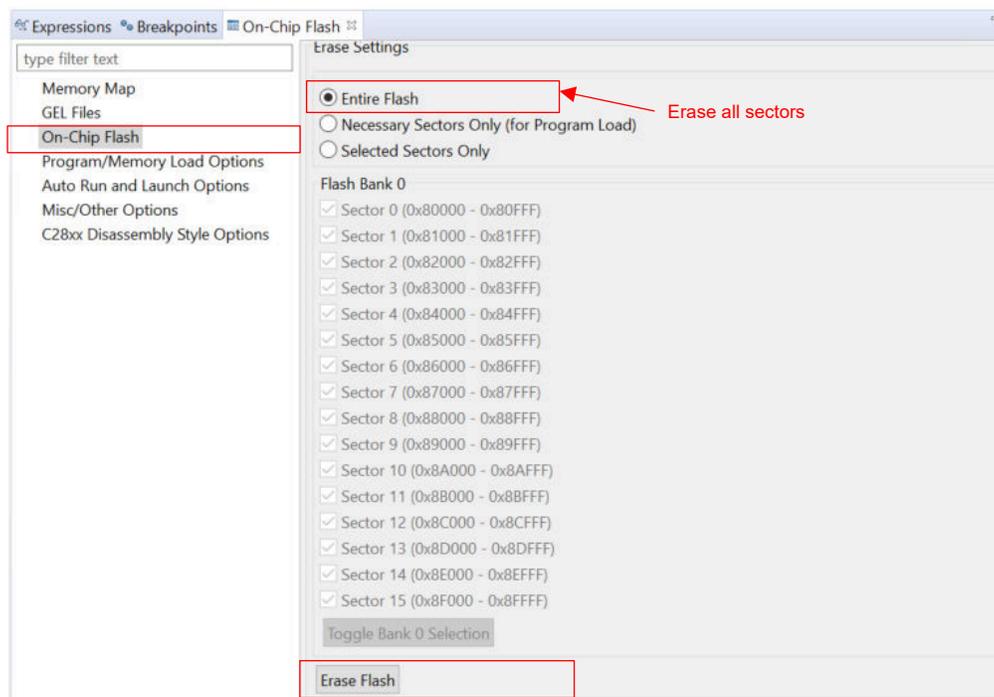


图 4-6. 擦除闪存中的程序代码以实现下一个构建级别

4.5.4.1 构建级别 1 测试程序

1. 确保已完成节 4.5.4 中列出的初始步骤。
2. 检查电机的校准偏移，电机相电流检测的偏移值可以约等于 ADC 标度电流的一半，如图 4-7 所示。
3. 在 J15 处使用示波器探测电机驱动控制的 PWM 输出，如图 4-8 中所示。在该构建级别中，所有 PWM 占空比都设置为 50%，PWM 输出波形如图 4-9 所示。motor_1 的 PWM 开关频率为 15kHz。

Expression	Type	Value	Address
motorVars_M1.ISRCCount	unsigned long	280658	0x0000894C@Data
systemVars.mainLoopCnt	unsigned long	975542	0x00008A84@Data
motorVars_M1.speed_rpm	float	-85.6637268	0x0000899C@Data
motorVars_M1.speed_Hz	float	-5.71624104	0x000089AC@Data
motorVars_M1.speedRef_Hz	float	40.0	0x000089A4@Data
motorVars_M1.speedSet_Hz	float	40.0	0x000089A2@Data
motorVars_M1.flagEnableRunAndIdentify	unsigned char	1 '\u001'	0x00008900@Data
motorVars_M1.flagRunIdentAndOnLine	unsigned char	1 '\u001'	0x00008901@Data
motorVars_M1.accelerationMax_Hzps	float	10.0	0x00008998@Data
motorVars_M1.flagEnableForceAngle	unsigned char	1 '\u001'	0x00008912@Data
motorVars_M1.flagMotorIdentified	unsigned char	1 '\u001'	0x00008902@Data
motorVars_M1.flagEnableMotorIdentify	unsigned char	0 '\u000'	0x00008923@Data
motorVars_M1.estState	enum <unnamed>	EST_STATE_ONLINE	0x0000892B@Data
motorVars_M1.motorState	unknown	member 'motorState' not found at (motorVars_M1).motorState	
motorVars_M1.adcData.VdcBus_V	float	40.6497231	0x00008950@Data
motorVars_M1.adcData	struct _HAL_ADCData_t	{VdcBus_V=40.7483864, I_A=(value=[0.218435556,0.175398439,-0.0...	0x00008950@Data
motorVars_M1.flagClearFaults	unsigned char	0 '\u000'	0x0000891D@Data
motorVars_M1.faultMtrUseAll	unsigned int	0	0x00008927@Data
motorVars_M1.faultMtrNow.bit	struct _FAULT_MTR_BITS_	{overVoltage=0, underVoltage=1, motorOverTemp=0, moduleOverT...	0x00008926@Data
motorVars_M1.speed_int_Hz	float	-362.948853	0x000089A6@Data
motorVars_M1.angleFOC_rad	float	2.62022805	0x000089AE@Data
motorVars_M1.angleEST_rad	float	-1.92112112	0x000089B0@Data
motorVars_M1.angleGen_rad	float	-2.91674447	0x000089B4@Data
motorCtrlVars_M1.flagEnableRunMotor	unsigned char	0 '\u000'	0x000086C4@Data
motorCtrlVars_M1.flagEnableSpeedCtrl	unsigned char	0 '\u000'	0x000086C6@Data
motorCtrlVars_M1.accelerationMax_Hzps	float	0.0	0x000086E0@Data
motorCtrlVars_M1.IqSet_A	float	0.0	0x000086D8@Data
motorVars_M1.speedEST_Hz	float	-4.97960234	0x000089AA@Data
motorVars_M1.speed_Hz	float	-5.71624104	0x000089AC@Data
motorVars_M1.angleFOC_rad	float	2.62022805	0x000089AE@Data
motorVars_M1.angleEST_rad	float	-1.92112112	0x000089B0@Data
motorVars_M1.anglePLL_rad	float	2.66169858	0x000089B2@Data
motorVars_M1.RsOnLine_Ohm	float	2.68207002	0x00008994@Data
motorSetVars_M1.Rs_Ohm	float	2.68207002	0x00008790@Data
motorSetVars_M1.Ls_d_H	float	0.00926135667	0x00008792@Data
motorSetVars_M1.Ls_q_H	float	0.00926135667	0x00008794@Data
motorSetVars_M1.flux_VpHz	float	0.0628318563	0x00008796@Data
motorVars_M1.Rs_Ohm	float	2.68207002	0x00008984@Data
motorVars_M1.Ls_d_H	float	0.00926135667	0x00008986@Data
motorVars_M1.Ls_q_H	float	0.00926135667	0x00008988@Data
motorVars_M1.flux_VpHz	float	0.0196469706	0x0000898A@Data

图 4-7. 构建级别 1：运行时的表达式窗口

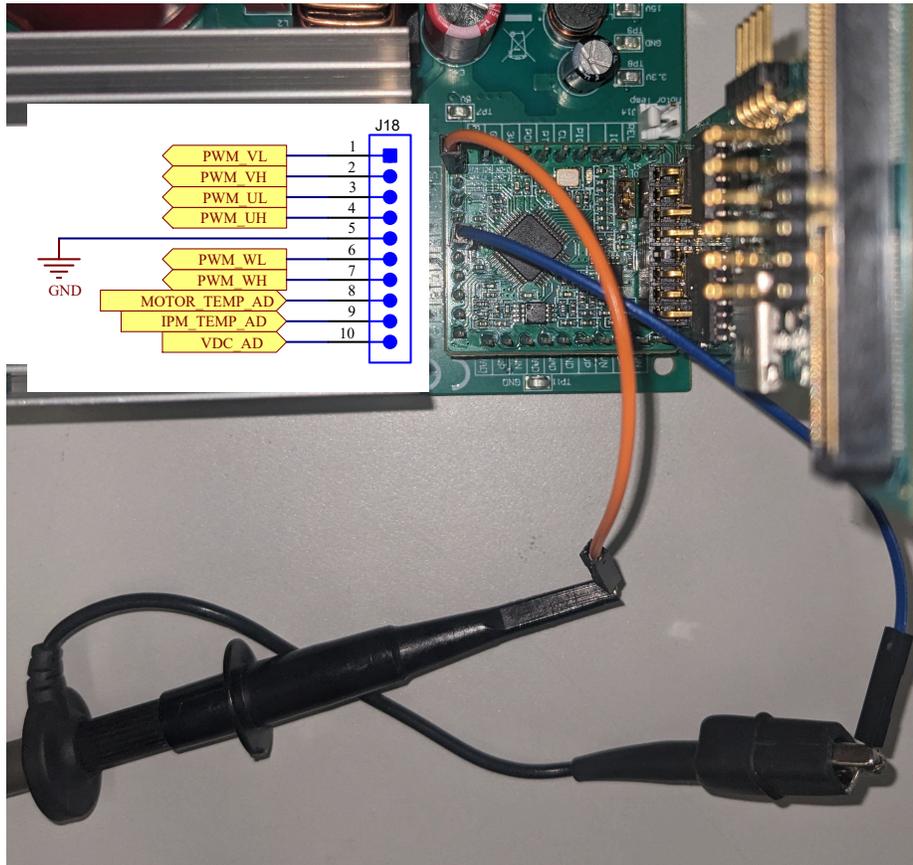


图 4-8. 探测 PWM 输出

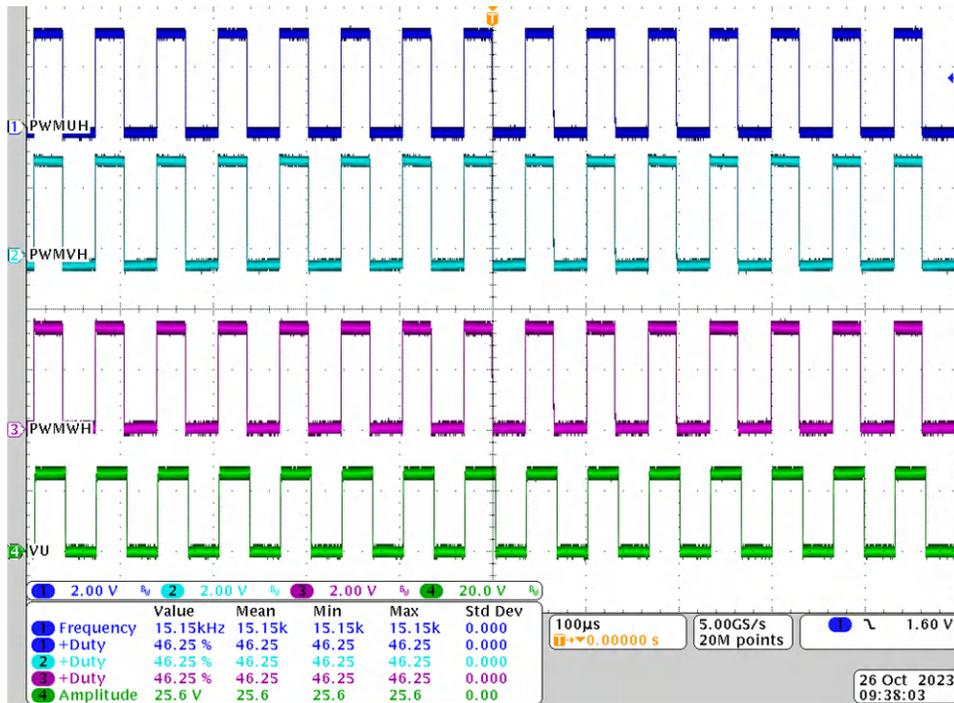


图 4-9. 构建级别 1 : MCU PWM 输出和 IPM 输出

4.5.4.2 构建级别 2 测试程序

1. 确保已完成节 4.5.4 中列出的初始步骤。
2. 要验证电机逆变器的电流和电压检测电路，MOTOR_1 需要以 *v/f* 开环运行。如果电机旋转不平稳，请根据电机规格调整 *user_mtr1.h* 中的 *v/f* 曲线参数，如下所示。

```
#define USER_MOTOR1_FREQ_LOW_HZ      (10.0f)      // Hz
#define USER_MOTOR1_FREQ_HIGH_HZ    (200.0f)     // Hz
#define USER_MOTOR1_VOLT_MIN_V      (10.0f)      // volt
#define USER_MOTOR1_VOLT_MAX_V      (200.0f)     // volt
```

3. 电机现在以变量 *motorVars_M1.speedRef_Hz* 中设置的速度旋转，并在 *Expressions* 窗口中检查 *motorVars_M1.speed_Hz* 的值。该值需要非常接近，如图 4-10 所示。
4. 如图 4-11 所示，连接示波器电压和电流探头以观察电机相电压和电流。
5. 通过减小变量 *motorVars_M1.overCurrent_A* 的值来验证过流故障保护，过流保护由 CMPSS 模块实现。如果 *motorVars_M1.overCurrent_A* 设为小于实际电流的值，则会触发过流故障，同时将禁用 PWM 输出，*motorVars_M1.flagEnableRunAndIdentify* 将清除为 0，*motorVars_M1.faultMtrUse.all* 将设置为“0x10”。

Expression	Type	Value	Address
motorVars_M1.ISRCCount	unsigned long	544363	0x0000894C@Data
systemVars.mainLoopCnt	unsigned long	2059177	0x00008A84@Data
motorVars_M1.speed_rpm	float	1176.95312	0x0000899C@Data
motorVars_M1.speed_Hz	float	80.0597916	0x000089AC@Data
motorVars_M1.speedRef_Hz	float	80.0	0x000089A4@Data
motorVars_M1.speedSet_Hz	float	80.0	0x000089A2@Data
motorVars_M1.flagEnableRunAndIdentify	unsigned char	1 '\x01'	0x00008900@Data
motorVars_M1.flagRunIdentAndOnLine	unsigned char	1 '\x01'	0x00008901@Data
motorVars_M1.accelerationMax_Hzps	float	20.0	0x0000898E@Data
motorVars_M1.flagEnableForceAngle	unsigned char	1 '\x01'	0x00008912@Data
motorVars_M1.flagMotorIdentified	unsigned char	1 '\x01'	0x00008902@Data
motorVars_M1.flagEnableMotorIdentify	unsigned char	0 '\x00'	0x00008923@Data
motorVars_M1.estState	enum <unnamed>	EST_STATE_ONLINE	0x0000892B@Data
motorVars_M1.motorState	unknown	member 'motorState' not found at (motorVars_M1).motorState	
motorVars_M1.adcData.VdcBus_V	float	141.188721	0x00008950@Data
motorVars_M1.adcData	struct_HAL_ADCData_t	{VdcBus_V=136.650162, I_A=[value=[2.74908686,3.78245902,4.2660...	0x00008950@Data
motorVars_M1.flagClearFaults	unsigned char	0 '\x00'	0x0000891D@Data
motorVars_M1.faultMtrUseAll	unsigned int	0	0x00008927@Data
motorVars_M1.faultMtrNow.bit	struct_FAULT_MTR_BITS_	{overVoltage=0, underVoltage=0, motorOverTemp=0, moduleOverT...	0x00008926@Data
motorVars_M1.speed_int_Hz	float	80.0	0x000089A4@Data
motorVars_M1.angleFOC_rad	float	-1.86383724	0x000089AE@Data
motorVars_M1.angleEST_rad	float	0.0245006047	0x00008980@Data
motorVars_M1.angleGen_rad	float	1.99211061	0x00008984@Data
motorCtrlVars_M1.flagEnableRunMotor	unsigned char	0 '\x00'	0x000086C4@Data
motorCtrlVars_M1.flagEnableSpeedCtrl	unsigned char	0 '\x00'	0x000086C6@Data
motorCtrlVars_M1.accelerationMax_Hzps	float	0.0	0x000086E0@Data
motorCtrlVars_M1.lqSet_A	float	0.0	0x000086D8@Data
motorVars_M1.speedEST_Hz	float	78.2935638	0x000089AA@Data
motorVars_M1.speed_Hz	float	80.0597916	0x000089AC@Data
motorVars_M1.angleFOC_rad	float	-1.86383724	0x000089AE@Data
motorVars_M1.angleEST_rad	float	0.0245006047	0x00008980@Data
motorVars_M1.anglePLL_rad	float	0.653951049	0x00008982@Data
motorVars_M1.RsOnLine_Ohm	float	2.68207002	0x00008994@Data
motorSetVars_M1.Rs_Ohm	float	2.68207002	0x00008790@Data
motorSetVars_M1.Ls_d_H	float	0.00926135667	0x00008792@Data
motorSetVars_M1.Ls_q_H	float	0.00926135667	0x00008794@Data
motorSetVars_M1.flux_VpHz	float	0.0628318563	0x00008796@Data
motorVars_M1.Rs_Ohm	float	2.68207002	0x00008984@Data
motorVars_M1.Ls_d_H	float	0.00926135667	0x00008986@Data

图 4-10. 构建级别 2：运行时的表达式窗口

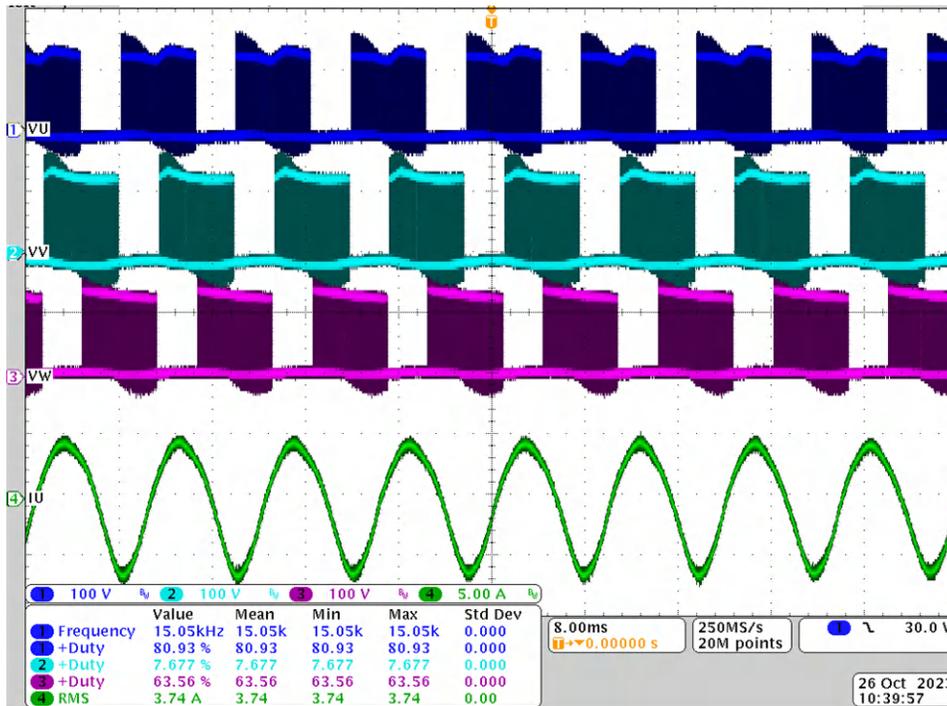
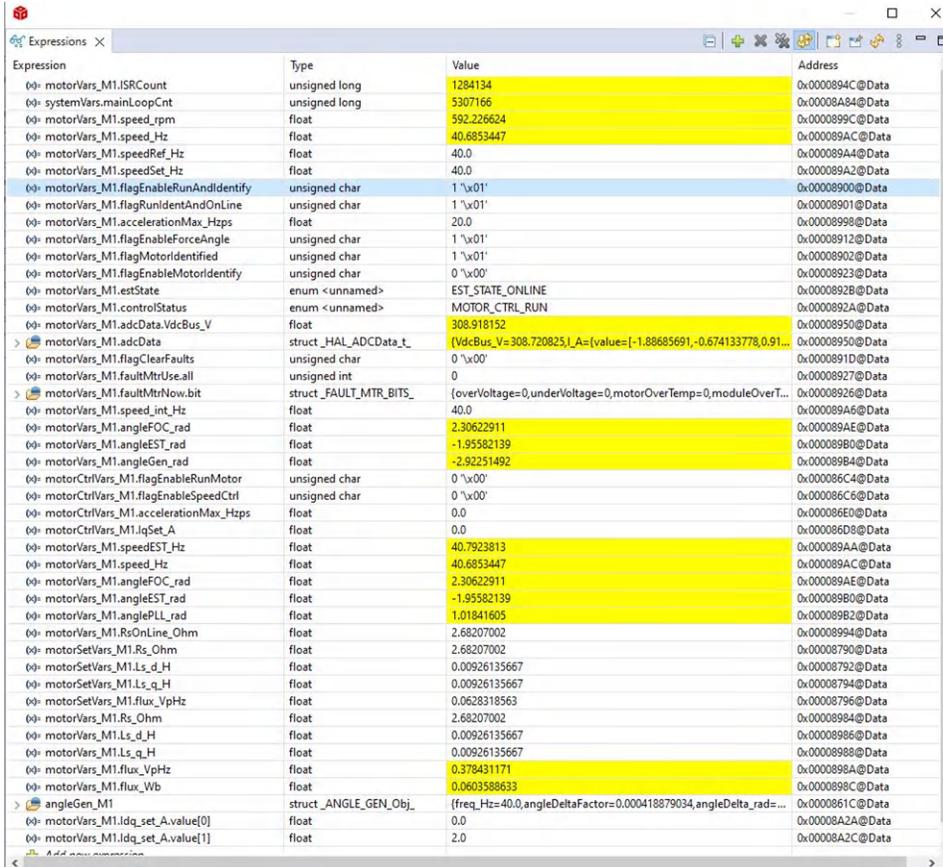


图 4-11. 构建级别 2：电机相电压和电流

4.5.4.3 构建级别 3 测试程序

1. 确保已完成节 4.5.4 中列出的初始步骤。
2. 电机需要使用来自角度发生器提供的角度，按照 `motorVars_M1.speedRef_Hz` 变量中设置的速度，以闭环控制运行。检查 Expressions 窗口中的 `motorVarsM1.speed_Hz` 值，该值需要非常接近目标 `motorVars_M1.speedRef_Hz`。
3. 电机电流 `Iq` 可通过 `motorVars_M1.Idq_Set_A.value[1]` 来调整。
4. 将示波器探头连接到 IPM 输出，以观察电机相电压和电流。更改 Expressions 窗口中的 `Idq_set_A[0].value[1]`，观察电机相电流相应地增加。



Expression	Type	Value	Address
motorVars_M1.ISRCCount	unsigned long	1284134	0x0000894C@Data
systemVars.mainLoopCnt	unsigned long	5307166	0x00008A84@Data
motorVars_M1.speed_rpm	float	592.226624	0x0000899C@Data
motorVars_M1.speed_Hz	float	40.6853447	0x000089AC@Data
motorVars_M1.speedRef_Hz	float	40.0	0x000089A4@Data
motorVars_M1.speedSet_Hz	float	40.0	0x000089A2@Data
motorVars_M1.flagEnableRunAndIdentify	unsigned char	1 '\x01'	0x00008900@Data
motorVars_M1.flagRunIdentAndOnLine	unsigned char	1 '\x01'	0x00008901@Data
motorVars_M1.accelerationMax_Hzps	float	20.0	0x00008998@Data
motorVars_M1.flagEnableForceAngle	unsigned char	1 '\x01'	0x00008912@Data
motorVars_M1.flagMotorIdentified	unsigned char	1 '\x01'	0x00008902@Data
motorVars_M1.flagEnableMotorIdentify	unsigned char	0 '\x00'	0x00008923@Data
motorVars_M1.estState	enum <unnamed>	EST_STATE_ONLINE	0x0000892B@Data
motorVars_M1.controlStatus	enum <unnamed>	MOTOR_CTRL_RUN	0x0000892A@Data
motorVars_M1.adcData.VdcBus_V	float	308.918152	0x00008950@Data
motorVars_M1.adcData	struct _HAL_ADCData_t	{VdcBus_V=308.7208251_A=[values=[-1.88685691,-0.674133778,0.91...	0x00008950@Data
motorVars_M1.flagClearFaults	unsigned char	0 '\x00'	0x0000891D@Data
motorVars_M1.faultMtrUse.all	unsigned int	0	0x00008927@Data
motorVars_M1.faultMtrNow.bit	struct _FAULT_MTR_BITS_	{overVoltage=0,underVoltage=0,motorOverTemp=0,moduleOverT...	0x00008926@Data
motorVars_M1.speed_int_Hz	float	40.0	0x000089A6@Data
motorVars_M1.angleFOC_rad	float	2.30622911	0x000089AE@Data
motorVars_M1.angleEST_rad	float	-1.95582139	0x000089B0@Data
motorVars_M1.angleGen_rad	float	-2.92251492	0x000089B4@Data
motorCtrlVars_M1.flagEnableRunMotor	unsigned char	0 '\x00'	0x000086C4@Data
motorCtrlVars_M1.flagEnableSpeedCtrl	unsigned char	0 '\x00'	0x000086C6@Data
motorCtrlVars_M1.accelerationMax_Hzps	float	0.0	0x000086ED@Data
motorCtrlVars_M1.IqSet_A	float	0.0	0x000086D8@Data
motorVars_M1.speedEST_Hz	float	40.7923813	0x000089AA@Data
motorVars_M1.speed_Hz	float	40.6853447	0x000089AC@Data
motorVars_M1.angleFOC_rad	float	2.30622911	0x000089AE@Data
motorVars_M1.angleEST_rad	float	-1.95582139	0x000089B0@Data
motorVars_M1.anglePLL_rad	float	1.01841605	0x000089B2@Data
motorVars_M1.RsOnLine_Ohm	float	2.68207002	0x00008994@Data
motorSetVars_M1.Rs_Ohm	float	2.68207002	0x00008990@Data
motorSetVars_M1.Ls_d_H	float	0.00926135667	0x00008792@Data
motorSetVars_M1.Ls_q_H	float	0.00926135667	0x00008794@Data
motorSetVars_M1.flux_VpHz	float	0.0628318563	0x00008796@Data
motorVars_M1.Rs_Ohm	float	2.68207002	0x00008994@Data
motorVars_M1.Ls_d_H	float	0.00926135667	0x00008986@Data
motorVars_M1.Ls_q_H	float	0.00926135667	0x00008988@Data
motorVars_M1.flux_VpHz	float	0.378431171	0x0000898A@Data
motorVars_M1.flux_Wb	float	0.0603588633	0x0000898C@Data
angleGen_M1	struct _ANGLE_GEN_Obj_	{freq_Hz=40.0,angleDeltaFactor=0.000418879034,angleDelta_rad=...	0x0000861C@Data
motorVars_M1.Idq_set_A.value[0]	float	0.0	0x00008A2A@Data
motorVars_M1.Idq_set_A.value[1]	float	2.0	0x00008A2C@Data

图 4-12. 构建级别 3 : 运行时的表达式窗口

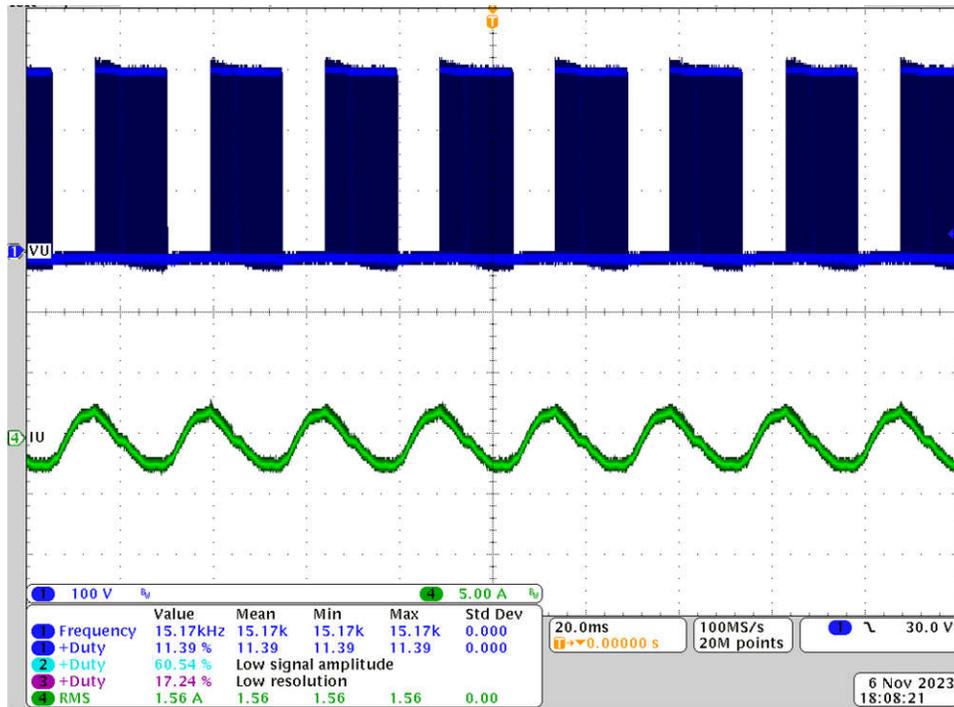


图 4-13. 构建级别 3 : 2A I_Q 设置下的电机电流

4.5.4.4 构建级别 4 测试程序

1. 确保已完成节 4.5.4 中列出的初始步骤。
2. 如果使用电机识别程序，如节 4.5.2 所述，则当在 *Expressions* 窗口中将 *motorVars_M1.flagEnableRunAndIdentify* 设置为“1”后，电机识别程序会立即开始执行。此过程大约需要 150 秒。
 - a. 当 *motorVars_M1.flagEnableRunAndIdentify* 等于“0”，表明已识别电机参数。使用 *user_mtr1.h* 中新定义的电机参数记录监视窗口值，如下所示：
 - USER_MOTOR1_Rs = *motorVars_M1.Rs_Ohm* 的值
 - USER_MOTOR1_Ls_d = *motorVars_M1.Ls_d_H* 的值
 - USER_MOTOR1_Ls_q = *motorVars_M1.Ls_q_H* 的值
 - USER_MOTOR_RATED_FLUX = *motorVars_M1.flux_VpHz* 的值
 - b. 成功识别电机参数后，将 *userParams_M1.flag_bypassMotorId* 设置为“true”，重新构建工程并将代码加载到控制器中。
3. 将变量 *motorVars_M1.speedRef_Hz* 设置为不同的值，并观察电机转轴转速的变化情况。
4. 要更改加速度，请为变量 *motorVars_M1.accelerationMax_Hzps* 和 *motorVars_M1.accelerationMax_Hzps* 输入不同的加速度值。

Expression	Type	Value	Address
motorVars_M1.ISRCCount	unsigned long	367093	0x0000894C@Data
systemVars.mainLoopCnt	unsigned long	1277708	0x00008A84@Data
motorVars_M1.speed_rpm	float	1501.23352	0x0000899C@Data
motorVars_M1.speed_Hz	float	100.178963	0x000089AC@Data
motorVars_M1.speedRef_Hz	float	100.0	0x000089A4@Data
motorVars_M1.speedSet_Hz	float	100.0	0x000089A2@Data
motorVars_M1.flagEnableRunAndIdentify	unsigned char	1 '\x01'	0x00008900@Data
motorVars_M1.flagRunIdentAndOnLine	unsigned char	1 '\x01'	0x00008901@Data
motorVars_M1.accelerationMax_Hzps	float	20.0	0x00008998@Data
motorVars_M1.flagEnableForceAngle	unsigned char	1 '\x01'	0x00008912@Data
motorVars_M1.flagMotorIdentified	unsigned char	1 '\x01'	0x00008902@Data
motorVars_M1.flagEnableMotorIdentify	unsigned char	0 '\x00'	0x00008923@Data
motorVars_M1.estState	enum <unnamed>	EST_STATE_ONLINE	0x0000892B@Data
motorVars_M1.controlStatus	enum <unnamed>	MOTOR_CTRL_RUN	0x0000892A@Data
motorVars_M1.adcData.VdcBus_V	float	309.510132	0x00008950@Data
motorVars_M1.adcData	struct _HAL_ADCData_t	{VdcBus_V=309.510132,I_A=(value=[0.198781252,-0.0507163107,-0...	0x00008950@Data
motorVars_M1.flagClearFaults	unsigned char	0 '\x00'	0x0000891D@Data
motorVars_M1.faultMtrUse.all	unsigned int	0	0x00008927@Data
motorVars_M1.faultMtrNow.bit	struct _FAULT_MTR_BITS_	{overVoltage=0,underVoltage=0,motorOverTemp=0,moduleOverT...	0x00008926@Data
motorVars_M1.speed_int_Hz	float	100.0	0x000089A6@Data
motorVars_M1.angleFOC_rad	float	-0.980699837	0x000089AE@Data
motorVars_M1.angleEST_rad	float	0.295398921	0x000089B0@Data
motorVars_M1.angleGen_rad	float	-2.30981064	0x000089B4@Data
motorCtrlVars_M1.flagEnableRunMotor	unsigned char	0 '\x00'	0x000086C4@Data
motorCtrlVars_M1.flagEnableSpeedCtrl	unsigned char	0 '\x00'	0x000086C6@Data
motorCtrlVars_M1.accelerationMax_Hzps	float	0.0	0x000086E0@Data
motorCtrlVars_M1.lqSet_A	float	0.0	0x000086D8@Data
motorVars_M1.speedEST_Hz	float	100.423782	0x000089AA@Data
motorVars_M1.speed_Hz	float	100.178963	0x000089AC@Data
motorVars_M1.angleFOC_rad	float	-0.980699837	0x000089AE@Data
motorVars_M1.angleEST_rad	float	0.295398921	0x000089B0@Data
motorVars_M1.anglePLL_rad	float	2.96600747	0x000089B2@Data
motorVars_M1.RsOnLine_Ohm	float	2.68207002	0x00008994@Data
motorSetVars_M1.Rs_Ohm	float	2.68207002	0x00008990@Data
motorSetVars_M1.Ls_d_H	float	0.00926135667	0x00008792@Data
motorSetVars_M1.Ls_q_H	float	0.00926135667	0x00008794@Data
motorSetVars_M1.flux_VpHz	float	0.0628318563	0x00008796@Data
motorVars_M1.Rs_Ohm	float	2.68207002	0x00008994@Data
motorVars_M1.Ls_d_H	float	0.00926135667	0x00008996@Data
motorVars_M1.Ls_q_H	float	0.00926135667	0x00008998@Data
motorVars_M1.flux_VpHz	float	0.389726102	0x0000899A@Data
motorVars_M1.flux_Wb	float	0.0620205812	0x0000899C@Data
angleGen_M1	struct _ANGLE_GEN_Obj_	{freq_Hz=100.0,angleDeltaFactor=0.000418879034,angleDelta_rad...	0x0000861C@Data

图 4-14. 构建级别 4 : 运行时的表达式窗口

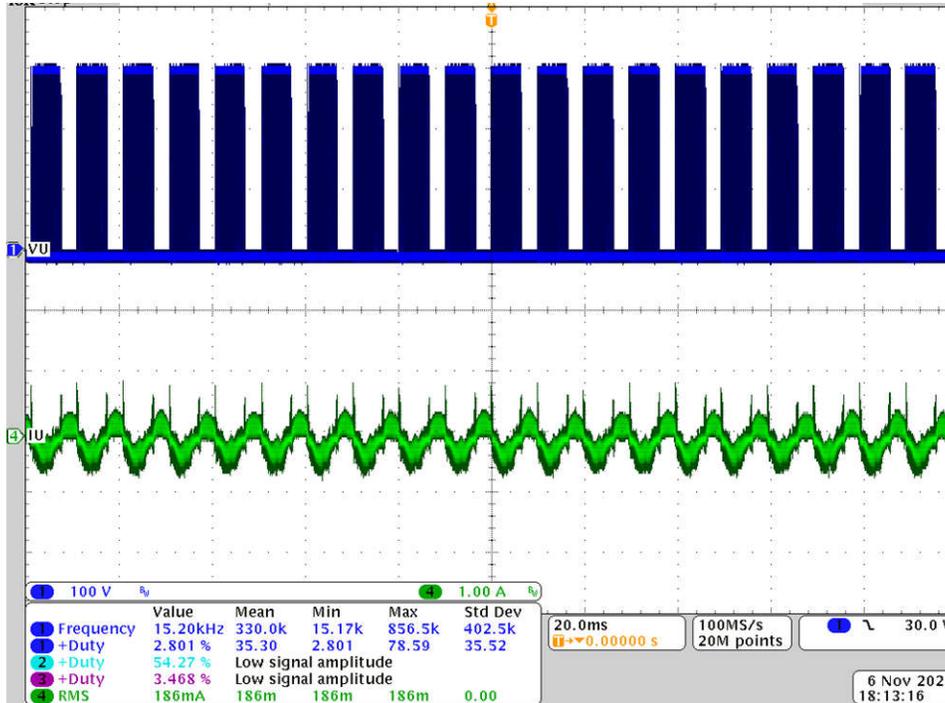


图 4-15. 构建级别 4 : 电机的转子角度、相电流

在这些初始测试之后，还可以在构建级别 4 选择性执行任何其他测试条件和调优。

1. 节 4.5.4.4.1
2. 节 4.5.4.4.2
3. 节 4.5.4.4.3

4.5.4.4.1 调整电机驱动 FOC 参数

滑模电流观测器由基于模型的电流观测器和继电器式控制发生器组成，后者依靠估算电机电流和实际电机电流之间的误差驱动。F 参数和 G 参数根据电机参数 R_s 和 L_s 进行计算，如节 3.1.2.1.2 所述。继电器式控制的观测器增益 k 、LPF 的截止频率以及 PLL 角度跟踪器的 K_p 和 K_i 必须根据测试状态进行调整，并尝试获得最佳参数。用户可以并行运行 FAST 估算器和 eSMO，从而验证来自 eSMO 的角度以调整相关参数。初始参数在 `user-mtr1.h` 文件中进行定义。

```
// Only for eSMO
#define USER_MOTOR1_KSLIDE_MAX      (1.50f)
#define USER_MOTOR1_KSLIDE_MIN      (0.75f)

#define USER_MOTOR1_PLL_KP_MAX      (10.0f)
#define USER_MOTOR1_PLL_KP_MIN      (2.0f)
#define USER_MOTOR1_PLL_KP_SF      (5.0f)

#define USER_MOTOR1_BEMF_THRESHOLD  (0.5f)
#define USER_MOTOR1_BEMF_KSLF_FC_HZ (2.0f)
#define USER_MOTOR1_THETA_OFFSET_SF (1.0f)
#define USER_MOTOR1_SPEED_LPF_FC_HZ (200.0f)
```

可以根据电机参数来计算转速和电流 PI 调节器增益，用户可以在线调整这些增益以优化系统的控制性能。

- 在 CCS Debug 透视图中向 *Expressions* 窗口添加 `motorVars[0].Kp_spd`、`motorVars[0].Ki_spd`、`motorVars[0].Kp_lq`、`motorVars[0].Ki_lq`、`motorVars[0].Kp_ld` 和 `motorVars[0].Ki_ld`。更改压缩机电机驱动器的 PI 增益并记录这些值。

4.5.4.4.2 调整弱磁和 MTPA 控制参数

添加 FWC 和 MTPA 函数并在电机驱动器 ISR 中调用来计算电流角，然后计算 d 轴和 q 轴的基准电流。

1. 在工程的构建配置中添加预定义符号 `MOTOR1_FWC` 和 `MOTOR1_MTPA` (如节 3.2.3 所述) 以分别启用 FWC 和 MTPA。
2. 在 `user_mtr1.h` 文件中，确保电机参数已知且设置正确。在 `mtpa.h` 中，确保根据电机规格正确设置表格并进行计算。
3. 在 CCS Debug 透视图中向 *Expressions* 窗口添加变量 `VsRef_pu`、`Kp_fwc` 和 `Ki_fwc`，并根据电机及系统调整这些参数以实现弱磁控制的预期性能。
4. 调整并修正这些变量后，使用 `user_mtr1.h` 文件中新定义的参数记录监视窗口值。

`USER_M1_FWC_VREF` = `VsRef_pu` 的值。用于弱磁控制的基准电压系数。

`USER_M1_FWC_KP` = `Kp_fwc` 的值。用于弱磁控制的 PI 稳压器 K_p 增益

`USER_M1_FWC_KI` = `Ki_fwc` 的值。用于弱磁控制的 PI 稳压器 K_i 增益

5. 可以根据电机参数 L_d 、 L_q 和 ψ_m 来计算 MTPA 控制参数，因此不需要在线调整任何额外的参数。

4.5.4.4.3 调整电流检测参数

精确的电流检测对于估算转子角度和转速以及实现最佳动态电机控制而言非常重要。电流检测参数必须与硬件匹配，这可以通过设置下面的相关参数来实现：

- 死区时间，上升沿延迟时间必须大于功率模块的（高侧开通时间）+（低侧关断时间），下降沿延迟时间必须大于电源模块的（高侧关断时间）+（低侧关断时间），如参考设计中使用的电源模块的以下设置所示。

```

//! \brief Defines the PWM deadband falling edge delay count (system clocks)
#define MTR1_PWM_DBFED_CNT (uint16_t)(2.5f * 120.0f) // 2.5us, (>2.0us)

//! \brief Defines the PWM deadband rising edge delay count (system clocks)
#define MTR1_PWM_DBRED_CNT (uint16_t)(2.5f * 120.0f) // 2.50us, (>2.0us)
    
```

- 脉宽 PWM 的最小持续时间，该时间必须大于（硬件延迟时间 + 死区时间 + 振铃持续时间 + ADC 采样时间）。

```

//! \brief Defines the minimum duration, clock cycle
#define USER_M1_DCLINKSS_MIN_DURATION (450U)
    
```

- 采样保持延迟时间，它指定从 PWM 输出到 ADC 采样时间的延时时间，用于电流检测。该延迟时间取决于硬件，包括栅极驱动器电路的传播延迟和功率 FET 的导通/关断延迟，其值小于或等于（最小持续时间 - ADC 采样时间）。

```

//! \brief Defines the sample delay, clock cycle
#define USER_M1_DCLINKSS_SAMPLE_DELAY (430U)
    
```

4.6 性能数据和结果

以下各节介绍了通过表征设计获得的测试数据。测试结果分为多个部分，涵盖风扇和压缩机电机的稳态性能和数据、功能性能波形以及瞬态性能波形。

4.6.1 负载和热力测试

图 4-16 是 500W 无负载条件下 3000RPM (200Hz) 时的波形。该波形显示了以下内容：

- CH1 (蓝色)：直流母线电压
- CH2 (浅蓝色)：交流输入电压
- CH4 (绿色)：U 相电流

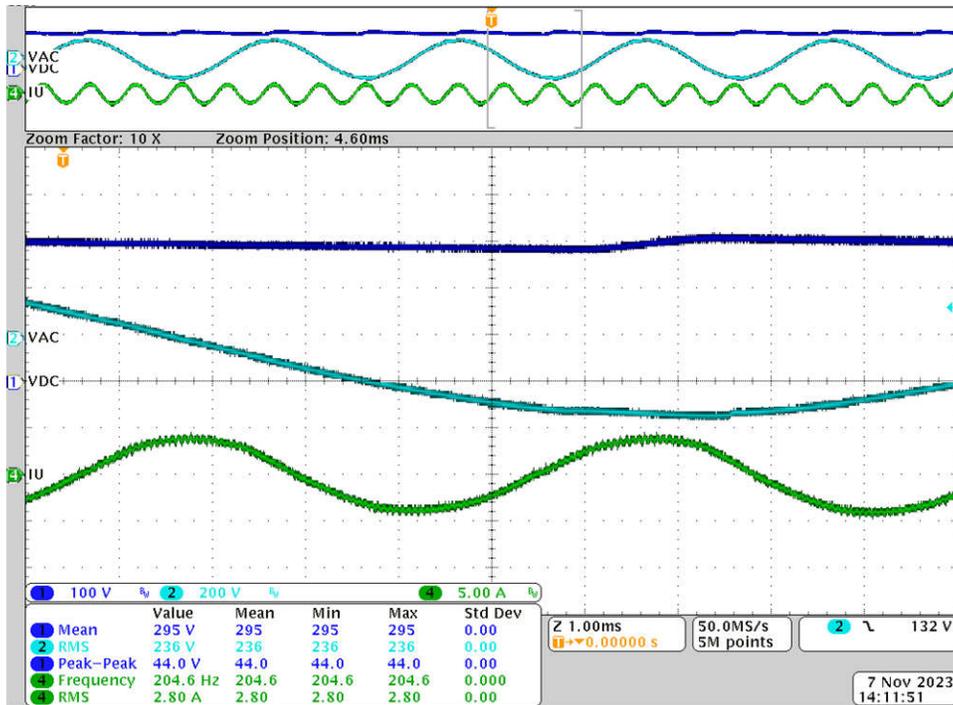


图 4-16. 500W、200Hz 条件下的电机相电流和电压波形

图 4-17 展示了启用磁场减弱时 300W 无负载和 3300RPM (220Hz) 转速条件下的波形。被测电机的额定转速为 3000RPM (200Hz)，现在以弱磁状态工作。

- CH1 (蓝色) : 直流母线电压
- CH2 (浅蓝色) : 交流输入电压
- CH4 (绿色) : U 相电流

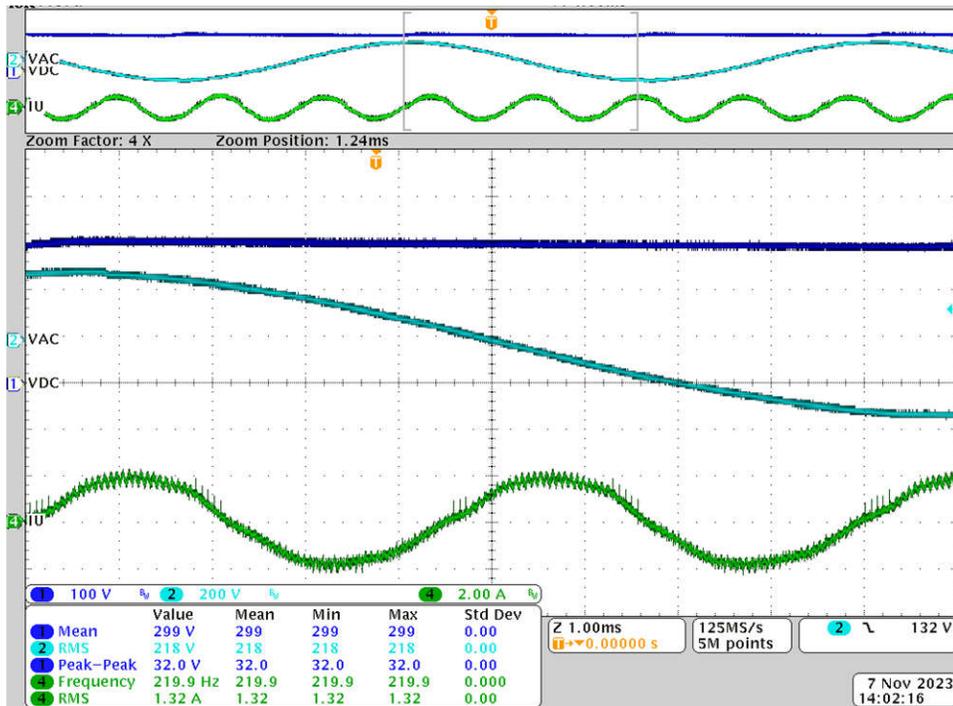


图 4-17. 300W、220Hz 条件下的弱磁测试

该板设计为在 750W 下以短时间 (≤ 1 分钟) 工作, 并注意温度升高。如果该电路板以高功率运行或运行时间过长, 请使用外部冷却风扇来冷却散热器。图 4-18 展示了 500W、3000RPM (200Hz) 时的电路板温升。

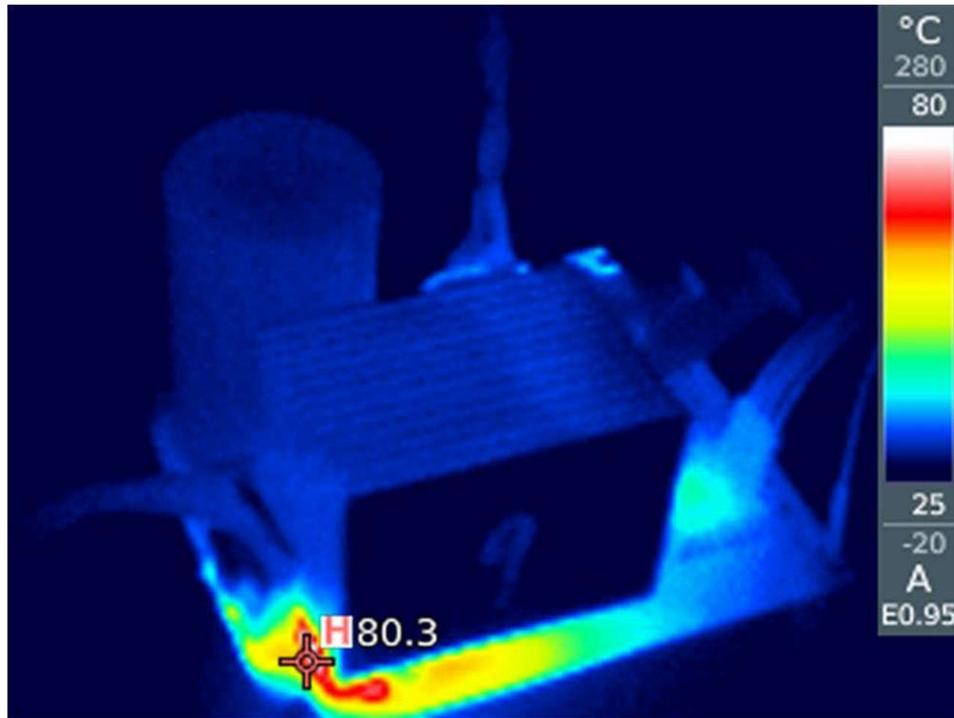


图 4-18. 220V 交流、500W、200Hz 下的热力测试

4.6.2 通过外部比较器进行过流保护

如节 2.1.5 中所述, 有一个用于外部过流保护的比较器 U10。图 4-19 展示了外部过流保护波形。当 R80 上的电流超过 U10 负输入设定的参考点时, U10 的输出 (net IPM_CIN) 为高电平, 然后高电平 IPM_CIN 触发 IPM 故障保护, 以在 IPM_FAULT 处输出低电平信号, 而 IPM_FAULT 连接到微控制器。

- CH1 (蓝色) : IPM_FAULT
- CH2 (浅蓝色) : IPM_CIN
- CH4 (绿色) : R80 的电流

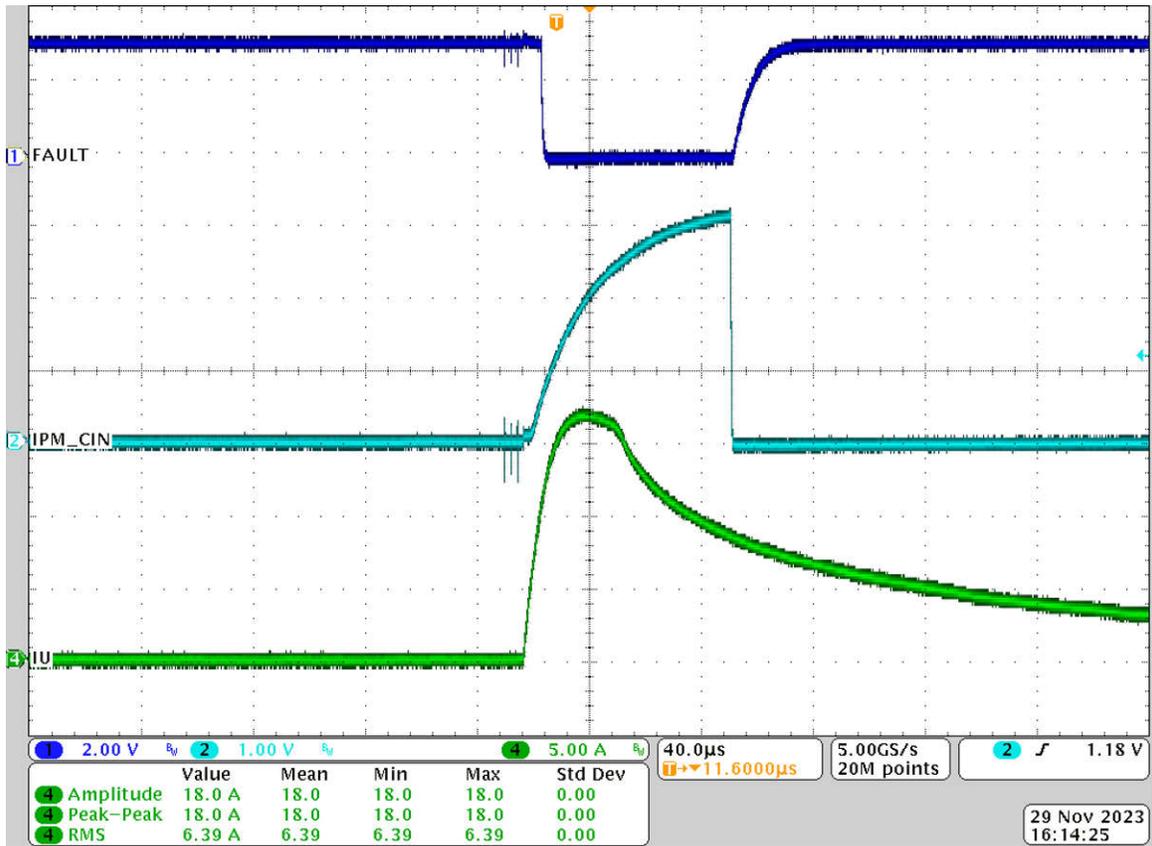


图 4-19. 通过外部比较器进行过流保护

4.6.3 通过内部 CMPSS 进行过流保护

如节 2.1.6 中所述，可以配置内部 CMPSS 来提供过流保护。图 4-20 显示内部过流保护波形由内部 CMPSS 触发，因为 IPM_FAULT 和 IPM_CIN 均未触发。过流可以通过以下代码设置。

`objSets->maxPeakCurrent_A = USER_M1_ADC_FULL_SCALE_CURRENT_A * 0.4975f;`

- CH1 (蓝色) : IPM_FAULT
- CH2 (浅蓝色) : IPM_CIN
- CH4 (绿色) : U 相电流

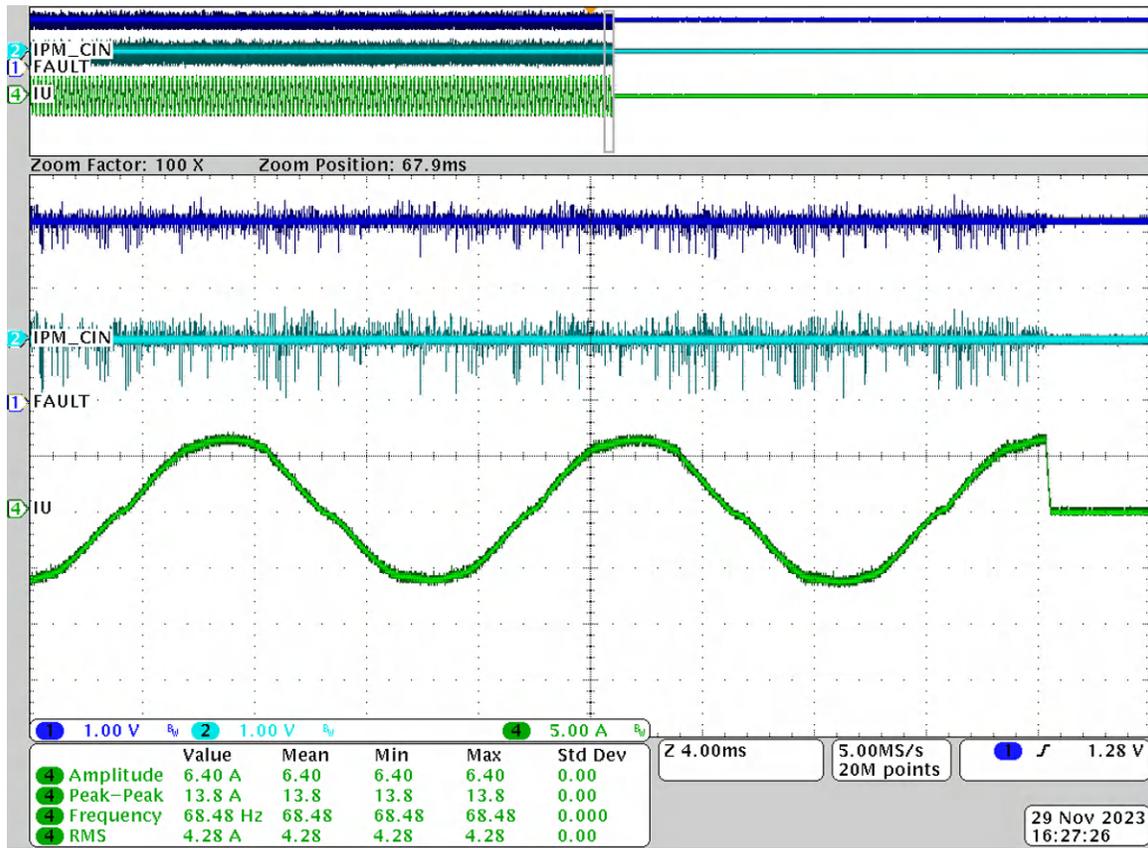


图 4-20. 通过内部 CMPSS 进行过流保护

5 硬件设计文件

5.1 原理图

TIEVM-MTR-HVINV 硬件设计文件，包括原理图，可在 TI.com 上 TIEVM-MTR-HVINV 的产品页面找到：[TIEVM-MTR-HVINV](#)

5.2 PCB 布局

TIEVM-MTR-HVINV 硬件设计文件，包括 PCB 布局布线，可在 TI.com 上 TIEVM-MTR-HVINV 的产品页面找到：[TIEVM-MTR-HVINV](#)

5.3 物料清单 (BOM)

TIEVM-MTR-HVINV 硬件设计文件，包括 BOM，可在 TI.com 上 TIEVM-MTR-HVINV 的产品页面找到：[TIEVM-MTR-HVINV](#)

6 其他信息

6.1 已知硬件或软件问题

已知某些电路板版本存在特定的硬件问题。

- 为了避免伪波非法 ISR 条目，不得组装子板外部精度 R 电阻器 R6 (默认不组装)。这适用于所有存在 R6 (连接到 GPIO19) 的版本，以及使用外部 R 功能的任何和所有设计。
- 电路板版本 E1 具有以下问题。之后的电路板版本 (如版本 A) 不存在这些问题。
 - 已知硬件问题
 - 已知电阻 R108 的值为 2.4k Ω 。这不正确，应该为 1k Ω 。这会导致外部 OCP 电路的 OCP 限制从设计值 (约 9.42A) 增加到约 21A。
 - 在此版本中，存在 J14 (电机温度外部传感器连接器)，不存在 J14 (隔离式 UART 连接器)。要利用 J17 上的 UART 信号，必须使用外部隔离器。
 - 已知软件问题
 - 要使用版本 E1，必须添加预定义符号 `TIEVM_MTR_HVINV_REV_E1`。如果没有这个预定义符号，U 桥臂和 W 桥臂的电流检测就会交换，电机就无法运行。

6.2 商标

FAST™, InstaSPIN™, and Code Composer Studio™ are trademarks of Texas Instruments.

所有商标均为其各自所有者的财产。

6.3 术语

SLYZ022	TI 术语表：本术语表列出并解释了常见的电气术语、首字母缩略词和定义
PMSM	永磁同步电机
BLDC	无刷直流
BEMF	反电动势
PWM	脉宽调制
FET、MOSFET	金属氧化物半导体场效应晶体管
IGBT	绝缘栅双极晶体管
RMS	均方根
MTPA	每安培最大扭矩
FWC	弱磁控制
FOC	场定向控制
HVAC	暖通空调
ESMO	增强型滑模观测器
PLL	锁相环
FAST	磁通、角度、转速和扭矩观测器

7 参考资料

1. 德州仪器 (TI), [TMS320F280013x 实时微控制器 数据表](#)
2. 德州仪器 (TI), [TMS320F280013x 实时微控制器技术参考手册](#)
3. 德州仪器 (TI), [InstaSPIN-FOC™ 和 InstaSPIN-MOTION™ 用户指南](#)
4. 德州仪器 (TI), [Motor Control SDK 通用工程和实验用户指南](#)
5. 德州仪器 (TI), [C2000™ 软件频率响应分析仪 \(SFRA\) 库和补偿设计器用户指南](#)
6. 德州仪器 (TI), [使用单一 DC-Link 分流器的 PMSM 无传感器 FOC 应用手册](#)
7. 德州仪器 (TI), [C2000 SysConfig 应用手册](#)

8 修订历史记录

注：以前版本的页码可能与当前版本的页码不同

Changes from OCTOBER 1, 2024 to NOVEMBER 30, 2024 (from Revision * (October 2024) to Revision A (November 2024))

	Page
• 添加了电机建议.....	2
• 针对版本 A 进行了更新并添加了螺柱.....	2
• 添加了电机建议.....	15
• 添加了用作示例的电机建议.....	16
• 添加了螺柱说明.....	16
• 增加的部分.....	36
• 添加了特定于版本 E1 的说明.....	37
• 更新了链接.....	61
• 更新了链接.....	61
• 更新了链接.....	61
• 更新了特定于版本 E1 的说明.....	62

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2024，德州仪器 (TI) 公司