Design Guide: TIDM-02010

具有数字交错式 PFC、适用于 HVAC 的双电机控制参考设计

说明

TIDM-02010 参考设计是一款适用于 HVAC 应用变频空调室外机控制器的 1.5kW 双电机驱动和 PFC 控制参考设计，展示了一种对压缩机和风扇电机驱动器以及数字交错式升压 PFC 实现无传感器三相 PMSM 矢量控制的方法，可通过单个 C2000™ 微控制器满足新的效率标准。此参考设计提供的硬件和软件已经过测试，而且可随时使用，有助于加快开发，从而缩短产品上市时间。本设计指南提供了硬件设计详细信息和测试结果。

资源

TIDM-02010	设计文件夹
TMS320F2800137、TMS320F280025C	产品文件夹
TMS320F280039C	产品文件夹
UCC28740、UCC27517、TLV9062	产品文件夹
C2000WARE-MOTORCONTROL-SDK	工具文件夹

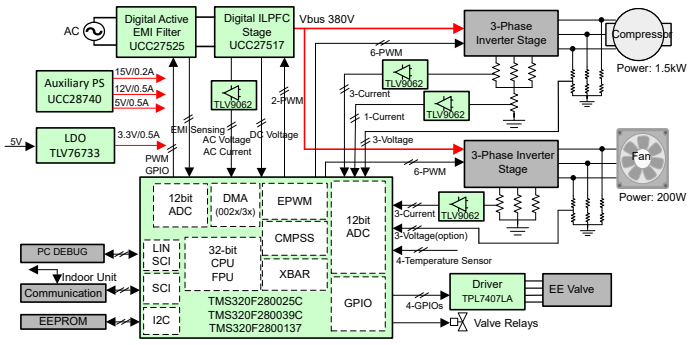
请咨询我司 TI E2E™ 支持专家

特性

- 宽工作电压输入范围：165V 至 265V 交流，50/60Hz。
- 使用一个 C2000 控制器实现对压缩机、风扇电机和数字交错式 PFC 升压转换器的无传感器 FOC 控制。
- 为压缩机 PMSM 提供高达 1.5kW 的逆变器级、6kHz 开关频率、扭矩补偿和自动弱磁控制。
- 提供高达 150W 的逆变器级，适用于风扇 PMSM 的 18kHz 开关频率，失速检测、自动恢复、快速启动和过流保护功能。
- 功率输出高达 1.5kW 的 72kHz 开关频率、数字交错式升压 PFC 控制，在整个工作电压范围内、从中等负载到满负载情况下的功率因数大于 0.95 且 THD 小于 5%，提供具有电压和电流故障保护功能的稳健可调电压输出。

应用

- 空调室外机
- 冰箱和冷冻柜
- 洗衣机和烘干机
- 电器压缩机



English Document: TIDUBE5

1 系统说明

当今的 HVAC 应用必须满足越来越多的需求，这些需求包括低成本、更小尺寸、更舒适、更强大和更高能效等。众所周知，在 HVAC 应用中，PMSM 比感应电机更受欢迎，并且受欢迎度越来越高。该参考设计为双电机驱动器提供了真正的单芯片解决方案，包括前端交错式升压 PFC。该参考设计说明了如何使用磁场定向控制 (FOC) (无位置传感器) 以及采用单个控制器的交错式升压 PFC 实现对两个电机的控制。整个系统可帮助用户减少物料清单中的关键元件数量，减小笨重的无源器件尺寸，提高效率，尽可能减小输入电流谐波含量，尽可能增大功率因数，精确调节直流总线，以及节省开发时间。该参考设计基于 TMS320F28002x 或 TMS320F28003x 实时控制器系列，有助于轻松开始 HVAC 系统室外机控制器设计。

控制多个电机的功能不仅可以降低系统成本，还可以提高总体功率效率和性能。对于操作双电机的应用，通过使用单个 MCU 来控制两个电机，控制器能够协调一个电机相对于另一个电机转速的加速快慢。此外，由于两个电机都从相同的电流源汲取电流，因此也可以对 PFC 实施进行协调以实现更佳结果。

该参考设计提供了支持多种不同类型空调系统的灵活性，以及用于电机驱动和 PFC 之外的更多可选功能的硬件功能，例如与室内机的串行通信、用于电子膨胀阀的步进电机驱动以及用于 HVAC 系统控制的温度传感器。

1.1 关键系统规格

表 1-1 列出了采用 PFC 的双电机控制参考设计规格。

表 1-1. 关键系统规格

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}	-	15	-	A
PFC 转换器特性					
PWM 开关频率 (f_{SW})	-	60	72	96	kHz
输出电压 (V_{OUT})	V_{IN} = 标称值，I_{OUT} = 最小值至最大值	360	375	385	V
输出电流 (I_{OUT})	V_{IN} = 最小值至最大值	-	-	5	A
线性调整率	V_{INAC} = 最小值至最大值，I_{OUT} = 标称值	-	-	2	%
负载调整率	V_{INAC} = 标称值，I_{OUT} = 最小值至最大值	-	-	3	%
输出电压纹波	V_{INAC} = 标称值，I_{OUT} = 最大值	-	-	15	V
输出过电压	V_{INAC} = 最小值至最大值	-	-	430	V
直流链路峰值过电流 (I_{OCP})	V_{INAC} = 最小值	-	-	10	A
高压线路的输出功率	V_{INAC} = 250V	-	-	1.8	kW
低压线路的输出功率	V_{INAC} = 165V	-	-	1.2	kW
效率 (η)	V_{INAC} = 满载时的标称值	-	97	-	%
压缩机逆变器特性					
PWM 开关频率 (f_{SW})	-	-	6	-	kHz
额定输出功率 (P_{OUT})	V_{INAC} = 标称值	-	1.4	1.5	kW
输出电流 (I_{RMS})	V_{INAC} = 标称值	-	10	-	A
逆变器效率 (η)	V_{INAC} = 标称值，P_{OUT} = 标称值	-	98	-	%
电机电频率 (f)	V_{INAC} = 最小值至最大值	20	200	400	Hz
故障保护	过流、失速、过热、欠压、过压				
驱动控制方法和功能	采用三个用于电流检测的分流电阻器的无传感器 FOC				
风扇逆变器特性					
PWM 开关频率 (f_{SW})	-	-	18	-	kHz
额定功率 (P)	V_{INAC} = 标称值	-	150	200	W
输出电流 (I_{RMS})	V_{INAC} = 标称值	-	1	-	A
逆变器效率 (η)	V_{INAC} = 标称值，P_{OUT} = 标称值	-	98	-	%
电机电频率 (f)	V_{INAC} = 最小值至最大值	30	50	400	Hz
故障保护	过流、恢复失速、欠压、过压				
驱动控制方法和功能	采用三个用于电流检测的分流电阻器的无传感器 FOC				
系统特点					
内置辅助电源	V_{INAC} = 最小值至最大值	15V±10%，300mA/12V±10%，500mA/5V±10%，500mA			
工作环境	开放式框架	-10	25	55	ºC
标准和规范	电磁干扰传导和辐	EC 61000-3-2 A 类			
电路板尺寸	长 × 宽 × 高	197mm * 197mm * 55mm			mm^2

WARNING

TI 建议，该参考设计仅可在实验室环境中运行，不可作为成品供一般消费者使用。

TI 建议，该参考设计仅可由熟悉处理高压电子和机械部件、系统及子系统所存在相关风险的合格工程师和技术人员使用。

高电压！ 电路板中存在可接触的高电压。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。使用该设备时应特别小心，并采取相应的保护措施，以避免伤害自己或损坏财产。

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，在运行过程中或运行刚结束时，用户不得触摸电路板。

CAUTION

请勿在无人照看的情况下使该设计通电。

2 系统概述

2.1 方框图

图 2-1 显示了该参考设计的方框图，其中突出显示了主要 TI 元件。

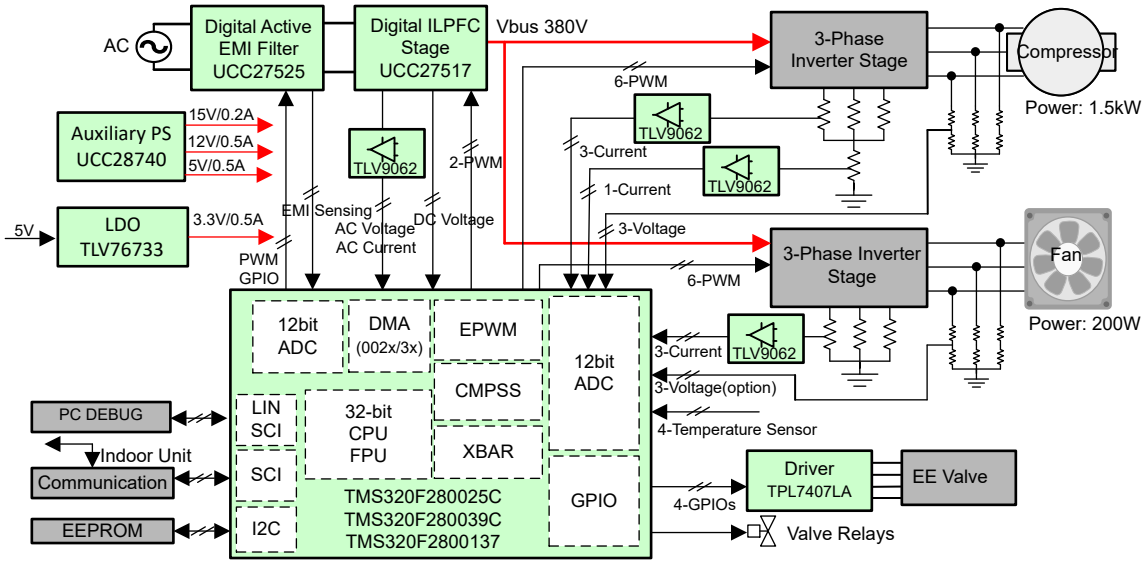


图 2-1. TIDM-02010 具有交错式 PFC 的双电机控制方框图

整个系统可分为七个块：

- 数字有源 EMI 滤波器
- 数字交错式 PFC
- 用于电机 1（压缩机）的三相逆变器
- 用于电机 2（风扇）的三相逆变器
- MCU 控制器
- 用于系统控制的阀门和继电器驱动器
- 辅助电源

2.2 设计注意事项

该设计使用单个 C2000 控制器实现双电机控制和 PFC。具有高抗噪性能的电流和电压检测解决方案对于精确的电机驱动和 PFC 控制而言是必不可少的。以下部分详细介绍了该设计中使用的检测和驱动电路。硬件设计文件位于 C2000Ware Motor Control SDK 安装目录 <install_location>\solutions \tidm_02010_dmpfc\hardware 下。

2.3 重点产品

本参考设计采用了以下重点产品。以下各节介绍为该参考设计选择器件时应考虑的主要特性。如需了解有关重点器件的更多详细信息，请参阅其各自的产品数据表。

2.3.1 TMS320F2800137

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 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 VCRC（循环冗余校验）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复用选项，可实现出色的信号布局。

2.3.2 TMS320F280025C

TMS320F28002x 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 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 100MHz 的信号处理性能。三角函数加速器 (TMU) 和 VCRC（循环冗余校验）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。高性能模拟模块集成在 F28002x 实时微控制器 (MCU) 上，并与处理单元和 PWM 单元紧密耦合，以提供更好的实时信号链性能。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。通过加入可配置逻辑块 (CLB)，用户可以添加自定义逻辑，还可将类似 FPGA 的功能集成至 C2000 实时 MCU 中。连接可以通过各种业界通用通信端口（如 FSI、SPI、SCI、I2C、PMBus、LIN 和 CAN）进行，另外还提供了多个引脚复用选项，可实现出色的信号布局。

2.3.3 TMS320F280039C

TMS320F28003x (F28003x) 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 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。浮点单元 (FPU)、三角函数加速器 (TMU) 和 VCRC（循环冗余校验）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。CLA 允许从主 C28x CPU 上大量卸载常见任务。CLA 是一款与 CPU 并行执行的独立 32 位浮点数学加速器。F28003x 支持高达 384KB (192KW) 的闪存，这些闪存分为三个 128KB (64KW) 存储体，支持并行编程和执行。高达 69KB (34.5KW) 的片上 SRAM 也可用于补充闪存。高性能模拟模块集成在 F28003x 实时微控制器 (MCU) 上，并与处理单元和 PWM 单元紧密耦合，以提供更好的实时信号链性能。16 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功率因数校正和高级多级电源拓扑的各种功率级。通过加入可配置逻辑块 (CLB)，用户可以添加自定义逻辑，还可将类似 FPGA 的功能集成至 C2000 实时 MCU 中。连接可以通过各种业界通用通信端口（如 SPI、SCI、I2C、PMBus、LIN、CAN 和 CAN FD）进行，另外还提供了多个引脚复用选项，可实现出色的信号布局。

2.3.4 UCC28740

UCC28740 隔离式反激电源控制器使用光耦合器来提供恒定电压 (CV)，从而改善对大型负载阶跃的瞬态响应。恒流 (CC) 调节通过一次侧稳压 (PSR) 技术来实现。此器件处理光耦合反馈信息和来自辅助反激式绕组的信息，以实现对输出电压和电流的精准高性能控制。内部采用 700V 启动开关，可动态控制工作状态并定制调制配置文件，支持超低待机功耗，并且不会影响启动时间或输出瞬态响应。UCC28740 中的控制算法使得运行效率满足或者超过适用标准。

2.3.5 UCC27517

UCC27517 是一款单通道、高速、低侧栅极驱动器器件，可有效驱动 MOSFET 和 IGBT 电源开关。UCC27517 采用能够将击穿电流降至极低水平的设计，可以将高峰值电流脉冲灌入和拉取到容性负载中，从而提供轨至轨驱动能力和极小的传播延迟（通常为 13ns）。UCC27517 在 VDD = 12V 时提供 4A 拉电流和 4A 灌电流（对称驱动）峰值驱动电流能力。UCC27517 可以在 4.5V 至 18V 的宽 VDD 范围和 –40°C 至 140°C 的宽温度范围内工作。

2.3.6 TLV9062

TLV9062 是具有轨至轨输入和输出摆幅功能的双路低压（1.8V 至 5.5V）运算放大器。这些器件是非常具有成本效益的解决方案，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 1μA 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反相。

2.3.7 TLV76733

TLV76733 是一款宽输入线性稳压器，支持 2.5V 至 16V 的输入电压范围和高达 1A 的负载电流，其输出精度为 1%，可满足低压微控制器 (MCU) 和处理器的需求。根据设计，TLV767 的 IQ 比传统的宽输入电压稳压器低得多，因此该器件能够充分满足日益严格的待机功耗要求。

2.4 系统设计原理

该参考设计的主要焦点是使用无传感器 FOC 的双电机控制，具有低 EMI、高效率、高功率因数和受保护的电源轨，适用于目标 HVAC 或其他电器应用。

2.4.1 交错式 PFC

升压转换器是 PFC 应用中最常用的拓扑。这是因为升压转换器具有连续输入电流，可以通过平均电流模式控制进行控制，从而强制输入电流跟踪线电压的变化。此外，更高的固有效率和宽占空比范围使其比降压/升压转换器更具吸引力。

虽然基本的单相升压转换器足以满足低功耗非隔离应用，但交错技术可在大功率电机驱动应用中提供额外的优势。两相交错式升压转换器只是两个以 180° 相位差并联运行的升压转换器。输入电流是两个电感器电流 I_{L1} 和 I_{L2} 之和。由于这两个电感的纹波电流是异相的，因此它们往往会相互抵消，从而降低升压电感器引起的输入纹波电流。在占空比为 50% 时，可以更大程度消除输入电感器纹波电流。输出电容器电流是两个二极管电流之和 (I_{D1} + I_{D2}) 减去直流输出电流。交错布置降低了输出电容器纹波电流 (I_{OUT})，该电流是占空比的函数。随着占空比接近 0%、50% 和 100%，两个二极管电流之和接近直流。在这些点上，输出电容器只需要滤除电感器纹波电流。

2.4.1.1 全桥二极管整流器额定值

- 电桥的电流额定值
  - 由于交错，输入电流的 rms 值也降低了。
  - 假设在实现最大 RMS 和峰值输入线电流时效率为 95%，校正功率因数为 0.99

$$I_{LINE\_RMS\_MAX} = \frac{P_O}{\eta \times V_{ACMIN} \times PF} \quad (1)$$

  - 最大峰值线电流是线电流峰值与最坏情况下两个电感器峰值纹波电流之和的总和。电感器额定值部分说明了峰值电感器电流的计算。

$$I_{LINE\_PEAK\_MAX} = \left(\sqrt{2} \times I_{LINE\_RMS\_MAX}\right) + \left(\frac{\Delta i_L}{2} \times 2\right) \quad (2)$$

  - 最大平均整流电流

$$I_{RCT\_AVG\_MAX} = \frac{2 \times I_{LINE\_PEAK\_MAX}}{\pi} \quad (3)$$

- 二极管桥的反向阻断电压额定值
  - 最大反向阻断是最大线电压的峰值

$$V_{LINE\_PEAK\_MAX} = \sqrt{2} \times V_{LINE\_RMS\_MAX} \quad (4)$$

- 考虑到大约 50% 的安全系数，所选电桥的反向阻断电压额定值为 600V
- 电桥中的功率耗散
  - 对于二极管电桥，求出最大平均电流下的正向电压

$$P_{Bridge\_MAX} = \sqrt{2} \times V_f \times I_{RCT\_AVG\_MAX} \quad (5)$$

2.4.1.2 电感器额定值

本节介绍了计算电感器感值和电流额定值的程序。

- 电感器的值

$$L_1 = L_2 = \frac{\sqrt{2} \times V_{ACMIN} \times D_{MIN}}{\Delta I_L \times F_{SW}} \quad (6)$$

其中 ΔI_L 是纹波电流峰峰值，最常见的设计实践建议将纹波电流峰峰值限制在平均电感器电流的 20-25%。

  - 电感器电流峰峰值额定值可通过以下公式确定：$\Delta I_L = \%\ Ripple \times I_{L(AVG)}$
  - 平均电感器电流可通过以下公式确定：$I_{L(AVG)} = \frac{\sqrt{2} \times I_{LINE\_RMS\_MAX}}{2}$

- 电感器额定电流
  - 峰值电感器电流额定值可通过以下公式确定： $I_{L2_{PEAK}} = I_{L2_{PEAK}} = I_{L(AVG)} + \left(\frac{\Delta i_L}{2}\right)$

2.4.1.3 交流电压检测

交流电压检测电路用于将电网电压信号转换为低压信号，然后对这些检测到的信号进行调节以满足 ADC 输入端口的要求。线电压和中性点电压由连接电路板地的电阻分压器检测，如图 2-2 所示。放大器在 ADC 输入端口减去这两个信号以获得 Vac 检测值。

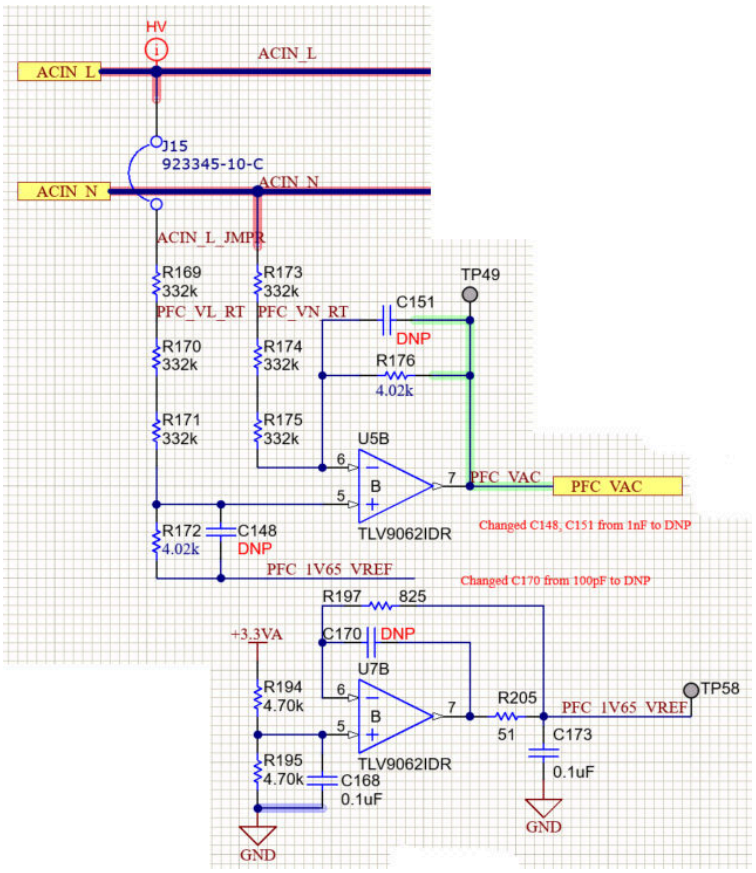


图 2-2. 输入交流电压检测电路

2.4.1.4 直流链路电压检测

同样，直流总线电压检测电路用于将直流链路电压信号转换为低压信号，由电阻分压器网络进行检测，如图 2-3 所示。

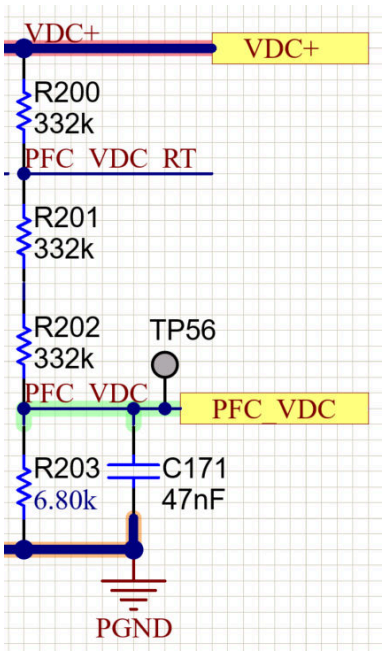


图 2-3. 直流总线电压检测电路

2.4.1.5 总线电流检测

电流检测电路用于将 PFC 电感电流信号转换为低压信号，然后对检测到的信号进行调节以满足 ADC 输入端口的要求。作为 IPFC 内部电流环路控制的一部分，交流电流检测至关重要。为此，该设计在二极管桥的返回点执行电流检测。它是交流电流的整流版本。该方法具有相对于系统地的电流检测优势。电流检测电路包含两个部分，如图 2-4 所示：

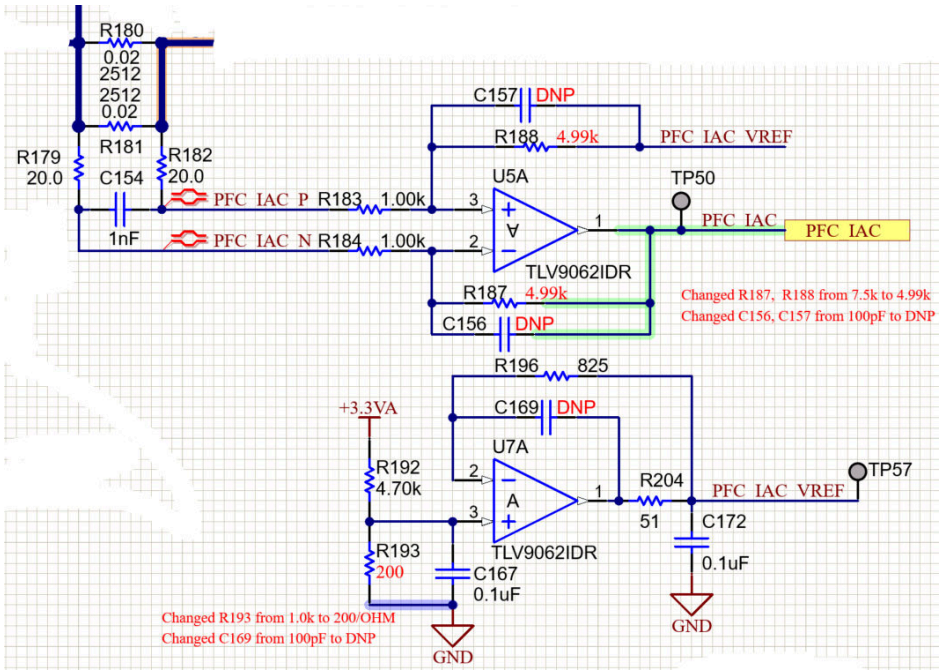


图 2-4. PFC 总线电流检测电路

- 分流电阻器计算

  分流电阻器上的建议电压确定为 0.2 至 0.25V，以优化功率消耗并实现更佳的噪声性能。分流器上的功率消耗已限制在总功率的 0.05% 以下，满量程电压的精确值是试错值，具体取决于可用的标准分流电阻器。

$$V_{SHUNT\_PFC} = I_{SHUNT\_PFC} \times R_{SHUNT\_PFC} \times G_i \quad (7)$$

其中 Gi = 4.892，如图 2-4 所示

$$R_{SHUNT\_PFC} = 0.02\Omega \parallel 0.02\Omega \quad = \quad 0.01\Omega \quad (8)$$

$$I_{SHUNT\_PFC} \quad = \quad 2 \quad \times \quad I_{L(MAX)} \quad (9)$$

2.4.1.6 直流链路电容器额定值

- TIDM02010 使用铝电解电容器来实现直流链路。虽然寿命显著延长的聚合物电解电容器已证明就有高可靠性，但在压缩机室外机等高振动环境中并不是最佳选择。与铝电解电容器中的液体电解质不同，聚合物电解电容器由于其固态聚合物结构而不能吸收振动。为了吸收直流链路中的高频分量，还在直流链路中使用了金属化薄膜电容器。这些电容器放置在非常靠近升压二极管的位置，以减小高频环路。
- 电容值计算
  - 基于保持电荷的直流链路电容器容值

$$C_{OUT(MIN)} \geq \frac{2 \times P_{OUT} \times \frac{1}{f_{LINE}}}{V_{OUT}^2 \quad - \left(\frac{V_{OUT}}{2}\right)^2} \quad (10)$$

  - 基于电压纹波要求的直流链路电容器容值

$$C_{OUT(MIN)} = \frac{1}{4} \times \frac{\left(\frac{P_{OUT}}{V_{OUT} \times \eta \times 0.637}\right)}{V_{RIPPLE} \times 0.8 \times \pi \times \mathrm{f_{LINE}}} \quad (11)$$

  - 电容器降额

    实际电容器容值会因初始电容器容差、温度和老化等因素而变化。每个因素考虑 20%。

$$C_{OUT} = \frac{C_{OUT(MIN)}}{\left(1 - \eta_{tolerance}\right) \times \left(1 - \eta_{temp}\right) \times \left(1 - \eta_{aging}\right)} \quad (12)$$

- 电容电压额定值
  - 考虑峰值纹波电压

$$WV_{DC\_BUS\_CA)P\_MIN} \quad = \quad V_{DCBUS} + \frac{\Delta V_{PP}}{2} \quad (13)$$

  - 考虑 10% 的瞬态和过压安全裕度

$$WV_{DC\_BUS\_CA)P} \quad = \quad 1.1 \times WV_{DC\_BUS\_CA)P\_MIN} \quad (14)$$

- 电容电流 RMS 值

  估算电容器 RMS 电流并不简单，需要执行以下步骤。

  - 角频率由下式给出

$$\omega = 2 \times \pi \times \mathrm{f_{LINE}} \quad (15)$$

  - 迭代次数

$$iteration = \frac{F_{SW\_PFC}}{2 \times F_{LINE}} \quad (16)$$

  - 一个阶跃的值（以秒为单位）：

$$Step \quad = \quad \frac{\frac{1}{f_{LINE}}}{Iteration} \quad (17)$$

  - 线电压是正弦函数，可由下式确定

$$V_{IN}(t) = \sqrt{2} \times V_{IN(RMS)} \times \sin(\omega t) \quad (18)$$

- 占空比也是输入电压的函数，可通过以下公式进行计算：

$$D(t) = \frac{V_{OUT} - V_{IN}(\omega t)}{V_{OUT}} \quad (19)$$

- 在采用功率因数校正时，输入电流也是正弦函数

$$I_{IN}(t) = \frac{\sqrt{2} \times P_{OUT}}{\eta_{System} \times V_{IN(MIN)}} \times \sin(\omega t) \quad (20)$$

$$I_{Crms} = \sqrt{\frac{1}{Iteration} \times \sum_{n=1}^{Iteration} \left\{ [I_{IN}(n \times Step)]^2 \times \left[\frac{1}{2} \times \sqrt{(2 - 2 \times D(n \times Step)) - (2 - 2 \times D(n \times Step))^2} \right]^2 \right\}} \quad (21)$$

- 电容器 ESR 额定值

$$ESR_{MAX} \geq \frac{V_{RIPPLE} \times 0.2}{\left(\frac{P_{OUT} \times \sqrt{2}}{V_{IN(MIN)} \times \eta} \right)} \quad (22)$$

2.4.1.7 MOSFET 额定值

- MOSFET 额定电压

  当 MOSFET 处于关断状态时，它必须阻断直流总线电压，该电压最高可上升至 400V。因此，考虑到要留出 50% 的余量作为安全系数，该设计选择了 600V MOSFET。
- MOSFET 额定电流

  当 MOSFET 导通时，它会在电感处于最低值时传导电感器电流峰值

$$I_{FET(PEAK)} = I_{L(PEAK)} \quad (23)$$

- MOSFET 中的功率耗散

  MOSFET 的总功率耗散包括导通损耗、关断损耗、开通损耗、C_{OSS} 损耗和栅极驱动损耗。可以使用以下公式估算该值

$$P_{FET} = P_{R_{DS(ON)}} + P_{SWITCH(t_{ON})} + P_{SWTICH(t_{OFF})} + P_{C_{OSS}} + P_{GATE} \quad (24)$$

- MOSFET 导通损耗

  MOSFET 中的导通损耗可以通过以下公式进行计算

$$P_{R_{DS(ON)}} = I_{FET(RMS)}^2 \times R_{DS(ON)} \quad (25)$$

  其中 FET RMS 电流可以使用以下公式进行计算：

$$I_{FET(RMS)} = \sqrt{f_{LINE} \times \sum_{n=1}^{Iteration} \left\{ \int_0^{\left[\frac{D(n \times Step)}{f_{SW}} \right]} \left[\frac{I_{IN}(n \times Step)}{2} \right]^2 dt \right\}} \quad (26)$$

- MOSFET 开通损耗
  - 由于开通操作而产生的 MOSFET 开关损耗

$$P_{SWITCH(t_{ON})} = f_{LINE} \times \sum_{n=1}^{Iteration} \left[I_{IN_{t_{ON}}}(n \times Step) \times V_{OUT} \times t_{ON(delay)} \right] \quad (27)$$

  - 其中导通状态下的电流的计算公式为

$$I_{IN_{t_{ON}}}(t) = \left(I_L - \frac{\Delta I_{L\_MAX}}{2}\right)\sin(\omega \times t) \quad (28)$$

- MOSFET 开通延迟时间的计算公式为

$$t_{ON(delay)} = \frac{Q_{GS(miller)}}{I_{GATE(ON)}} \quad (29)$$

- $I_{GATE(ON)}$ 是开通操作期间的平均 FET 栅极驱动电流

$$I_{GATE(ON)} = \frac{V_{GS(MAX)}}{2 \times R_{GATE(ON)}} \quad (30)$$

其中，$R_{GATE(ON)}$ = R70 = R71。这是开通操作期间栅极电流路径中的电阻。

- MOSFET 关断
  - 由于关断操作而产生的 MOSFET 开关损耗

$$P_{SWITCH(t_{OFF})} = f_{LINE} \times \sum_{n=1}^{Iteration}\left[I_{IN_{t_{OFF}}}\left(n \times Step\right) \times V_{OUT} \times t_{OFF(delay)}\right] \quad (31)$$

  - 其中关断状态下的电流的计算公式为

$$I_{IN_{t_{OFF}}}(t) = \left(I_L + \frac{\Delta I_{L\_MAX}}{2}\right)\sin(\omega \times t) \quad (32)$$

  - MOSFET 关断延迟时间的计算公式为

$$t_{OFF(delay)} = \frac{Q_{GS(miller)}}{I_{GATE(OFF)}} \quad (33)$$

  - $I_{GATE(OFF)}$ 是关断操作期间的平均 FET 栅极驱动电流。该模式下的电流流经二极管和与二极管串联的电阻器。因此，考虑到正向压降，$I_{GATE(OFF)}$ 的计算公式为：

$$I_{GATE(OFF)} = \frac{V_{GS(MAX)} - V_{f\_Diode}}{2 \times R_{GATE(OFF)}} \quad (34)$$

- 总 FET 损耗的一部分是源于 PWM 开关周期内充电和放电期间的影响因素 C_{OSS} ($C_{OSS(AVG)}$)，其计算公式为

$$P_{C_{OSS}} = \frac{1}{2} \times C_{OSS(AVG)} \times V_{OUT}^2 \times f_{SW\_PFC} \quad (35)$$

其中 C_{OSS} 随线电压的变化而变化，其关系不是线性函数。FET 数据表中的以下公式和信息可用于计算 $C_{OSS(AVG)}$。$C_{OSS(SPEC)}$ 是在指定电压 $V_{DS(SPEC)}$ 下测得的典型 C_{OSS}。

$$C_{OSS(AVG)} = 2 \times C_{OSS(SPEC)} \times \sqrt{\frac{V_{DS(SPEC)}}{V_{OUT}}} \quad (36)$$

- MOSFET 栅极损耗可通过以下计算公式来确定

$$P_{GATE} = Q_{GS} \times V_{GS(MAX)} \times f_{SW\_PFC} \quad (37)$$

2.4.1.8 二极管额定值

- 二极管类型选择
  - 在该设计中，碳化硅 (SiC) 肖特基二极管用作升压二极管。
  - 与标准的超快速硅功率二极管相比，硅肖特基器件正向压降更低并且正向和反向恢复特性更低，因此具有更高的性能。但其低击穿电压限制了其在低压应用中的使用。
  - 碳化硅以其高反向击穿电压和快速恢复克服了这些限制。
- 二极管反向电压额定值
  - 二极管反向电压类似于 MOSFET 阻断电压。当二极管反向偏置时，它必须阻断直流总线电压，该电压最高可达 400V。将 50% 的余量作为安全系数，选择 600V 反向阻断额定值。
- 二极管电流额定值
  - 当二极管导通时，通过二极管的电流的平均值和峰值可通过以下公式确定：

$$I_{DIODE(AVG)} = \frac{P_{OUT}}{2 \times V_{OUT(MIN)} \times \eta_{DIODE}} \quad (38)$$

$$I_{DIODE(PEAK)} = I_{L(MAX)} \quad (39)$$

- 二极管中的功率耗散
  - 二极管整流器中有两种损耗，即导通损耗和反向恢复损耗。主要损耗是由正向压降造成的，可以通过以下公式得出：

$$P_{DIODE} = I_{DIODE(AVG)} \times V_F \quad (40)$$

2.4.2 三相 PMSM 驱动器

永磁同步电机 (PMSM) 具有一个绕线定子、一个永磁转子组件和用于检测转子位置的内部或外部器件。感测器件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。定子绕组通电后可产生旋转电磁场。为了控制旋转的磁场，有必要控制定子电流。
- 根据机器的功率范围和额定速度，转子的实际结构会有所不同。永磁体适合于范围高达几千瓦的同步机器。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。转子的机械结构是针对所需磁极的数量和所需的磁通梯度进行设计的。
- 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，因此转子的旋转将产生一个有用的机械输出，如图 2-5 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。这会减少最优扭矩产出量，并且在机器部件上产生过多的机械抖动、噪声和机械应力。此外，如果转子因惯性而不能对这些摆动做出响应，那么转子的转动会偏离同步频率，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° 以获得最高的互扭矩产出量。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。

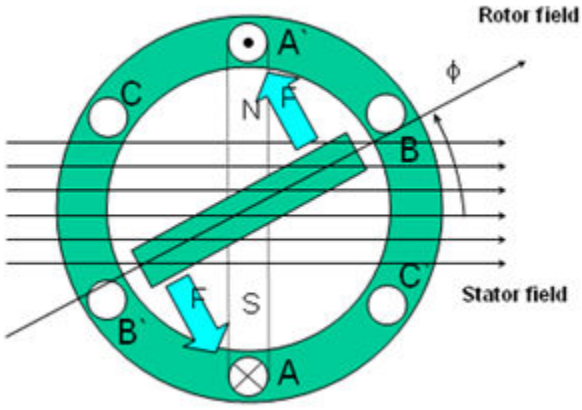


图 2-5. 旋转的定子磁通和转子磁通之间的相互作用产生扭矩

2.4.2.1 PM 同步电机的磁场定向控制

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 PM 电机。借助微控制器提供的数学处理能力，我们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，或简称为磁场定向控制 (FOC)。

在直流电机中，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 2-6 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最佳情况。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。

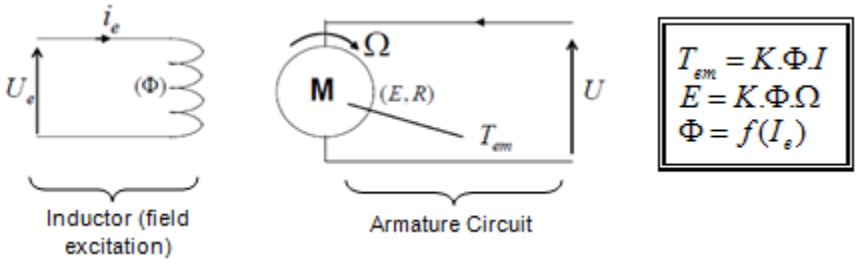


图 2-6. 在直流电机模型中磁通和扭矩是独立控制的

同步和异步电机上的 FOC（也称为矢量控制）旨在分别控制扭矩产生分量和磁化通量分量。利用 FOC 控制，我们能够解耦定子电流的扭矩分量和磁化通量分量。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。为了去耦合扭矩和磁通，有必要采用几个数学变换，而这是最能体现微控制器价值的地方。微控制器提供的处理能力可非常快速地执行使这些数学变换。反过来，这意味着控制电机的整个算法可以高速率执行，从而实现了更高的动态性能。除了去耦合，现在一个电机的动态模型被用于很多数量的计算，例如转子磁通角和转子速度。这意味着，它们的影响被计算在内，并且总体控制质量更佳。

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 41 所示。

$$\tau_{em} = \vec{B}_{stator} \times \vec{B}_{rotor} \tag{41}$$

该表达式表明，如果定子和转子磁场正交，则扭矩最大，这意味着我们需要将负载保持在 90 度。如果我们能够始终确保满足这一条件，并且能够正确地对磁通进行定向，将减少扭矩纹波并确保实现更好的动态响应。然而，您需要了解转子的位置：这可以通过位置传感器（诸如递增编码器）实现。对于无法接近转子的低成本应用，采用不同的转子位置观察器策略可无需使用位置传感器。

简而言之，目标是使转子和定子磁通保持正交：例如，目标是将定子磁通与转子磁通的 q 轴对齐，从而与转子磁通正交。为了实现这个目的，控制与转子磁通正交的定子电流分量以产生命令规定的扭矩，并且直接分量被设定为零。定子电流的直接分量可用在某些磁场减弱的情况下，这有抗拒转子磁通的作用，并且减少反电动势，从而实现更高速的运行。

磁场定向控制包括控制由矢量表示的定子电流。该控制基于将三相时间和速度相关系统变换为两坐标（d 和 q 坐标）时不变系统的投影。这些投影形成了一个与直流电机控制结构相似的结构。磁场定向控制 (FOC) 电机需要两个常数作为输入基准：扭矩分量（与 q 坐标对齐）和磁通分量（与 d 坐标对齐）。由于磁场定向控制只是基于这些投影，因此控制结构将处理瞬时电量。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传统方案存在的问题：

- 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 42 所示。

$$\tau_{em} \propto \psi_R \times i_{sq} \tag{42}$$

通过将转子磁通（ψ_R）的振幅保持在一个固定值，扭矩和扭矩分量（i_{Sq}）之间存在线性关系。然后我们可以通过控制定子电流矢量的扭矩分量来控制扭矩。

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 是定子的三相即时电流，则复数定子电流矢量的定义如方程式 43 所示。

$$\bar{i}_s = i_a + \alpha i_b + \alpha^2 i_c \quad (43)$$

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

图 2-7 所示为定子电流的复数空间矢量。

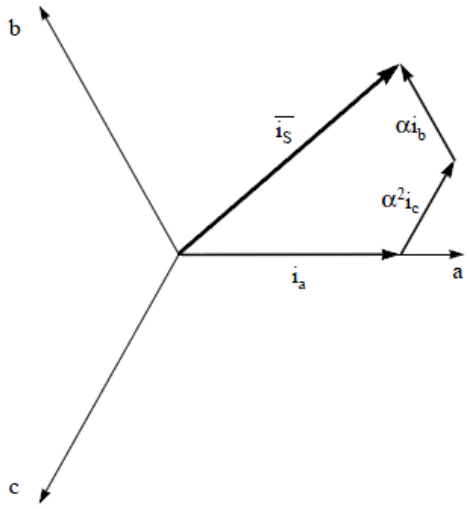


图 2-7. 定子电流空间矢量及其以 (a,b,c) 坐标系表示的分量

其中 (a,b,c) 是三相系统轴。这个电流空间矢量对三相正弦系统进行了描述，但仍需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

$(a,\ b) \Rightarrow (\alpha, \beta)$ 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，我们可以得到下面图 2-8 所示的矢量图。

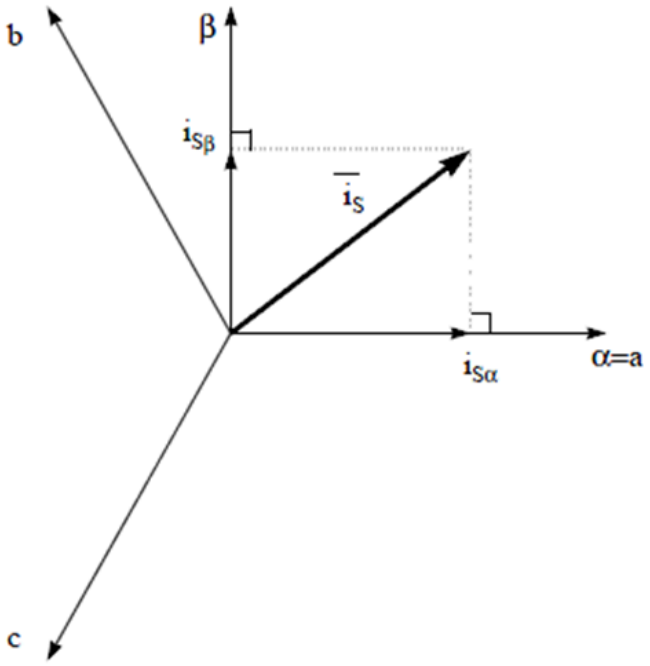


图 2-8. 静止坐标系中的定子电流空间矢量

将三相系统修改为 (α , β) 二维正交系统的投影如方程式 44 所示。

$$i_{s\alpha} = i_a$$
$$i_{s\beta} = \frac{1}{\sqrt{3}} i_a + \frac{2}{\sqrt{3}} i_b \quad (44)$$

两相 (α , β) 电流仍取决于时间和速度。

$(\alpha, \beta) \Rightarrow (d, q)$ **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)。如果我们考虑 d 轴与转子磁通对齐，那么图 2-9 显示了来自该二维坐标系的电流矢量的关系。

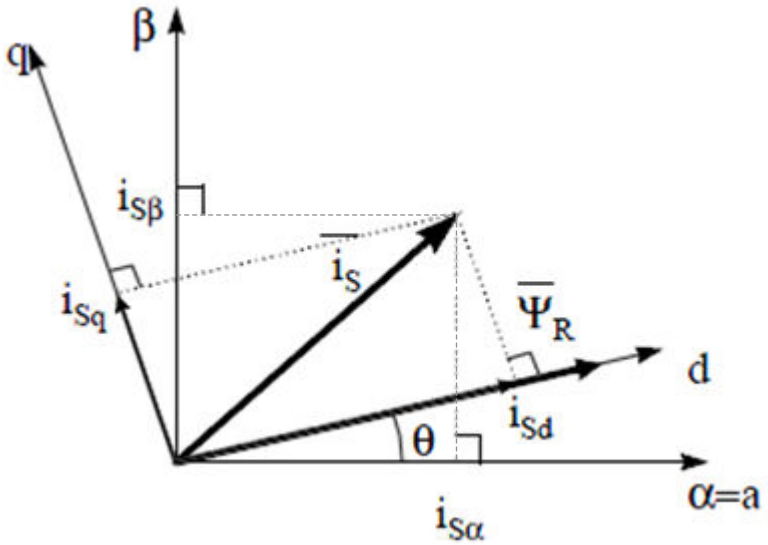


图 2-9. d,q 旋转坐标系中的定子电流空间矢量

电流矢量的磁通和扭矩分量由方程式 45 决定。

$$i_{sd} = i_{s\alpha}\cos(\theta) + i_{s\beta}\sin(\theta)$$
$$i_{sq} = -i_{s\alpha}\sin(\theta) + i_{s\beta}\cos(\theta) \quad (45)$$

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；如果我们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。

交流电机 FOC 基本配置方案

图 2-10 总结了使用 FOC 进行扭矩控制的基本配置方案：

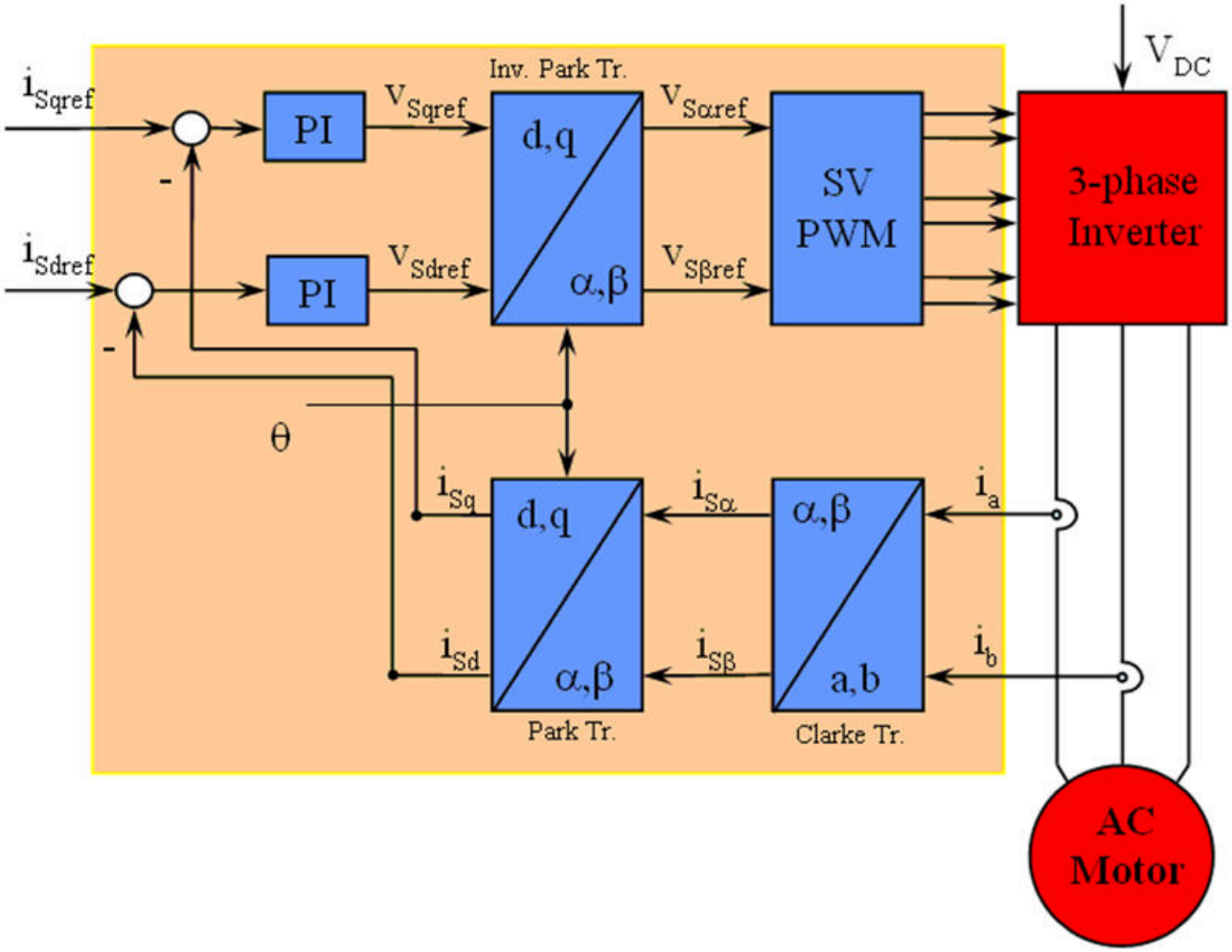


图 2-10. 交流电机 FOC 基本配置方案

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，该变换给出了 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。在这一点上，这个控制结构显示了一个有意思的优势：它可被用来控制同步或感应机器，采用的方法就是简单地改变磁通基准并获得转子磁通位置。与在同步永磁电机中一样，转子磁通是固定的，并由磁体确定；所以无需产生转子磁通。因此，当控制一个 PMSM 时，i_{sdref} 应被设定为 0。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。当我们使用转速 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 Vsdref 和 V_{sqref}；它们进行 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。

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 2-11 显示了 (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。

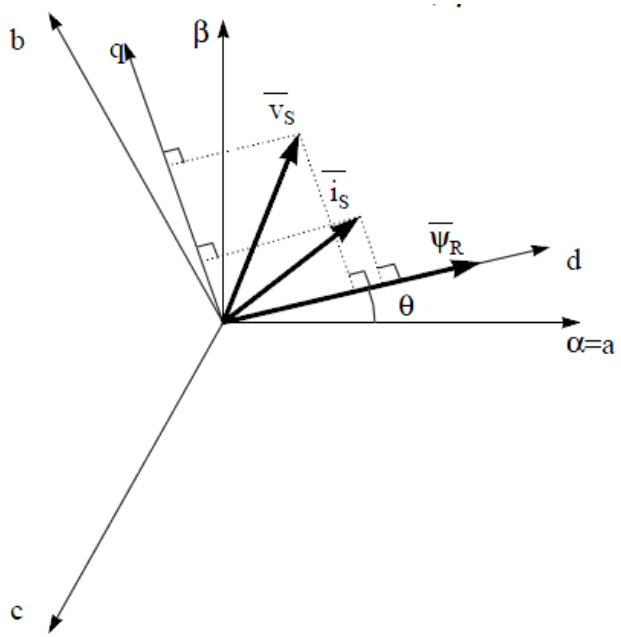


图 2-11. (d, q) 旋转坐标系中的电流、电压和转子磁通空间矢量

如果我们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，利用适用于 PMSM 驱动的磁场定向控制，可以使用磁通实现对电机扭矩的单独控制，这与直流电机的运行类似。换句话说，扭矩和磁通互相之间去耦合。从静止基准框架到同步旋转基准框架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 轴电流将控制扭矩，而 d 轴电流被强制设置为零。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 2-12 显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 2-13 显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 2-14 显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 2-15 显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。

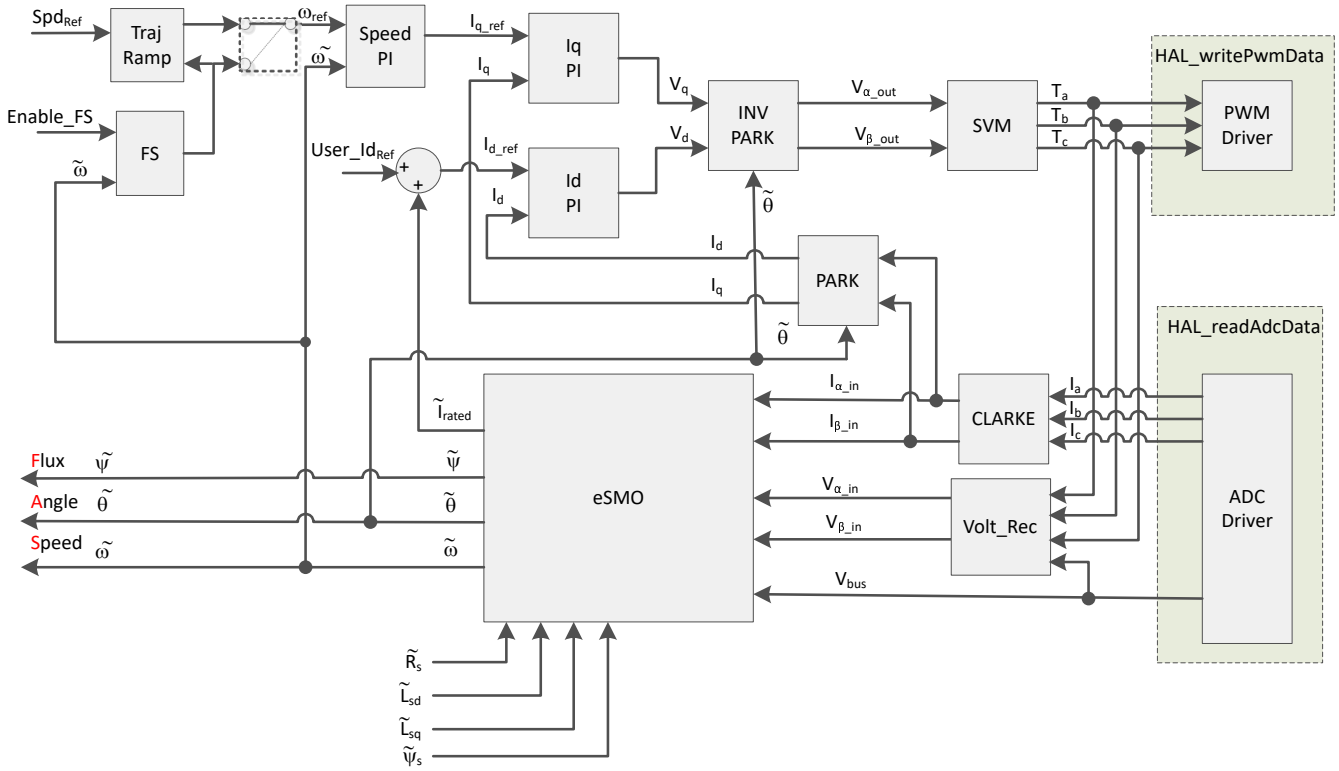


图 2-12. 使用 eSMO 并具有快速启动 (FS) 功能的风扇 PMSM 的无传感器 FOC

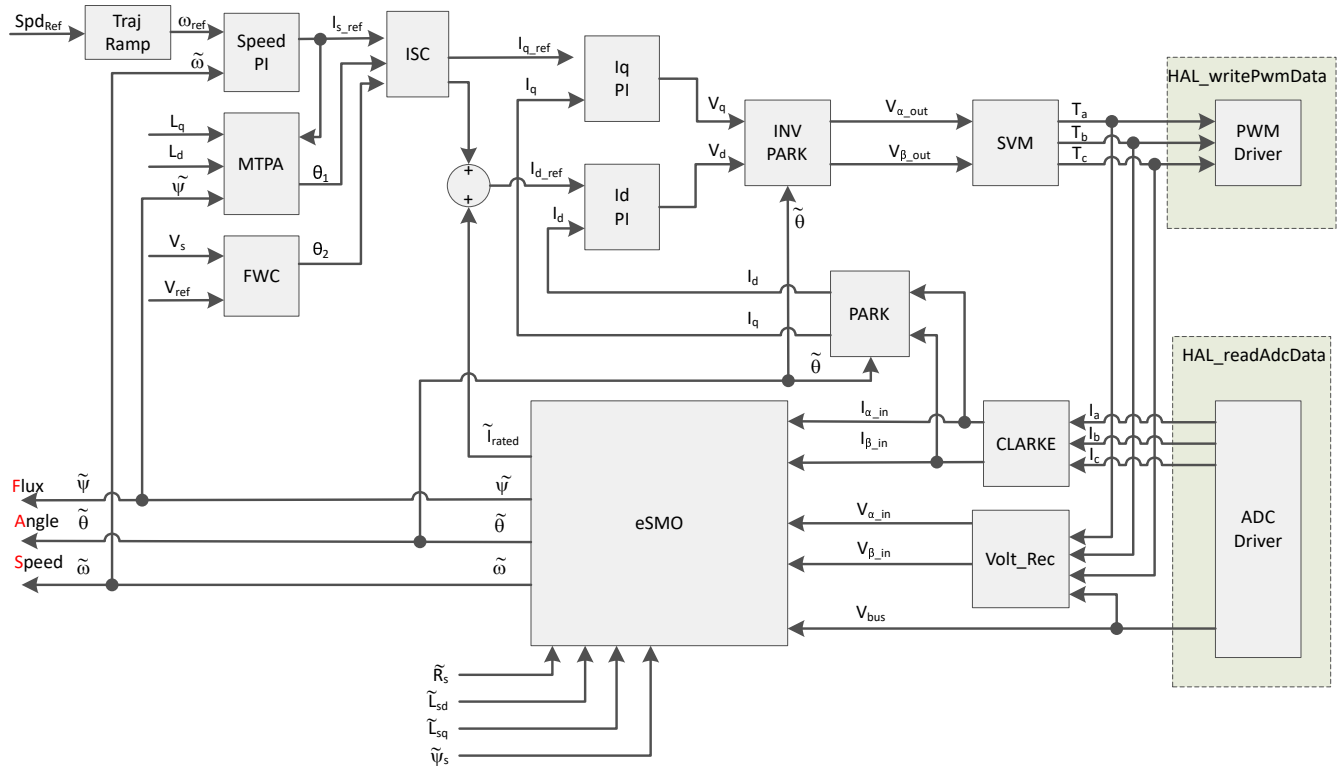


图 2-13. 使用 eSMO 并具有 FWC 和 MTPA 功能的压缩机 PMSM 的无传感器 FOC

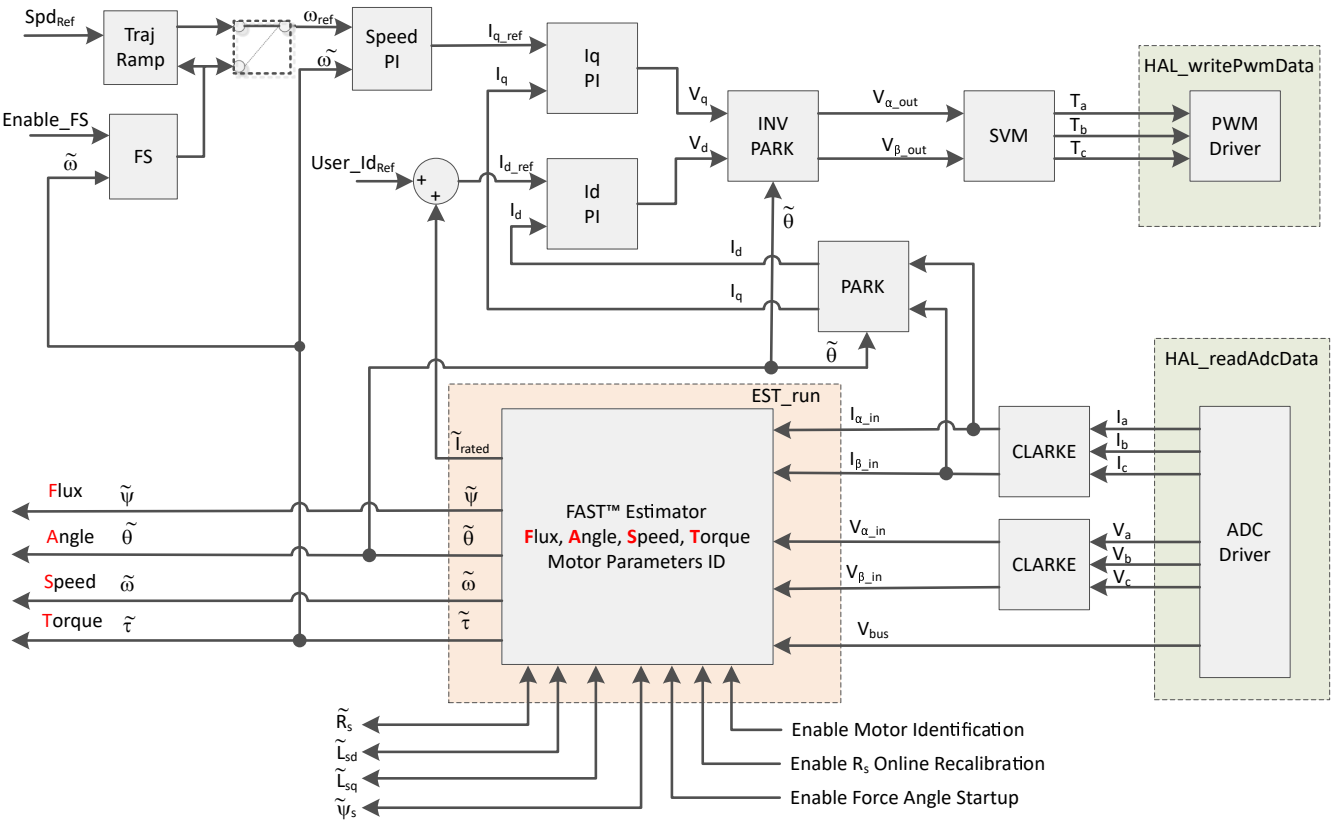


图 2-14. 使用 FAST 并具有快速启动 (FS) 功能的风扇 PMSM 的无传感器 FOC

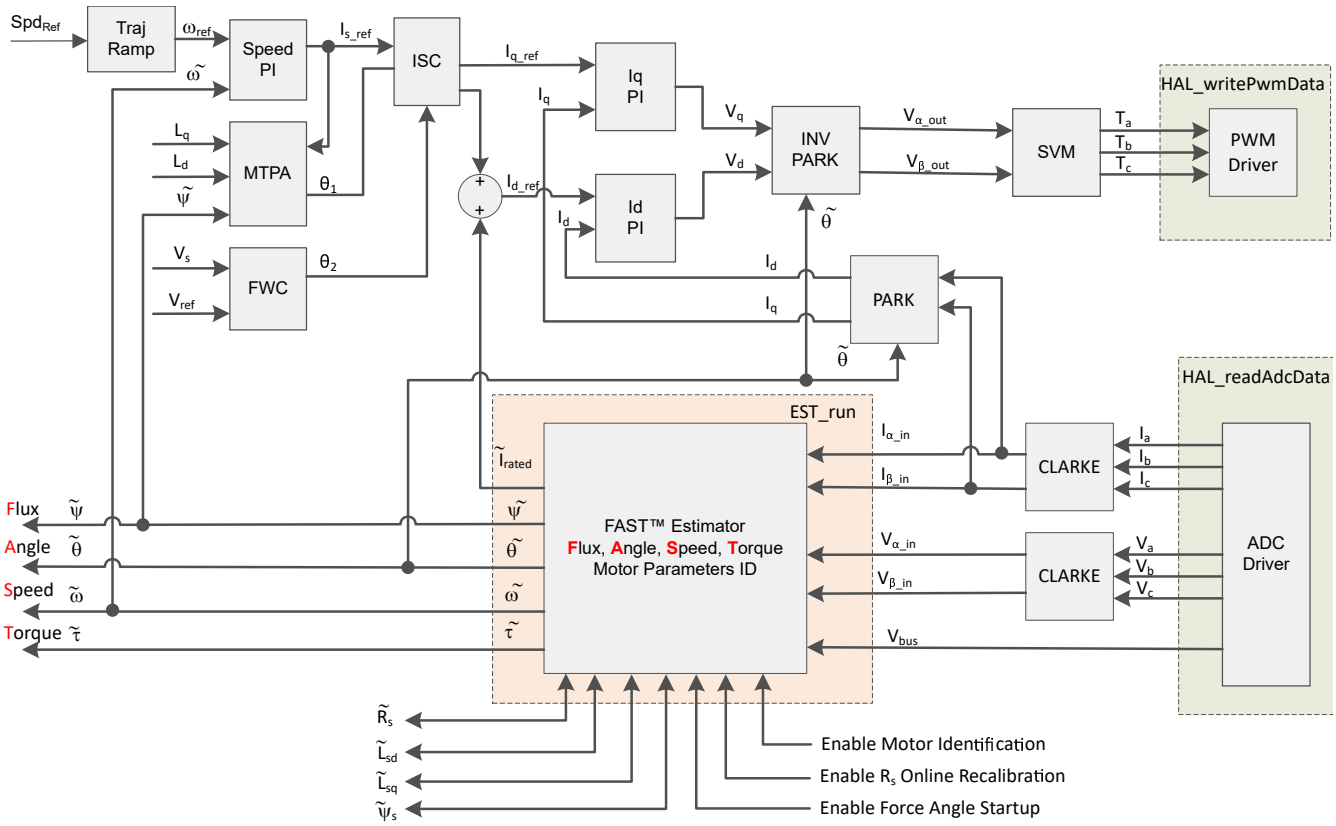


图 2-15. 使用 FAST 并具有 FWC 和 MTPA 功能的压缩机 PMSM 的无传感器 FOC

2.4.2.2 PM 同步电机的无传感器控制

在家用电器应用中，如果使用机械传感器，将会导致成本、尺寸和可靠性问题增加。为了克服这些问题，无传感器控制方法应运而生，它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。

2.4.2.2.1 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。

2.4.2.2.1.1 IPMSM 的数学模型和 FOC 结构

IPMSM 的无传感器 FOC 结构如图 2-16 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。

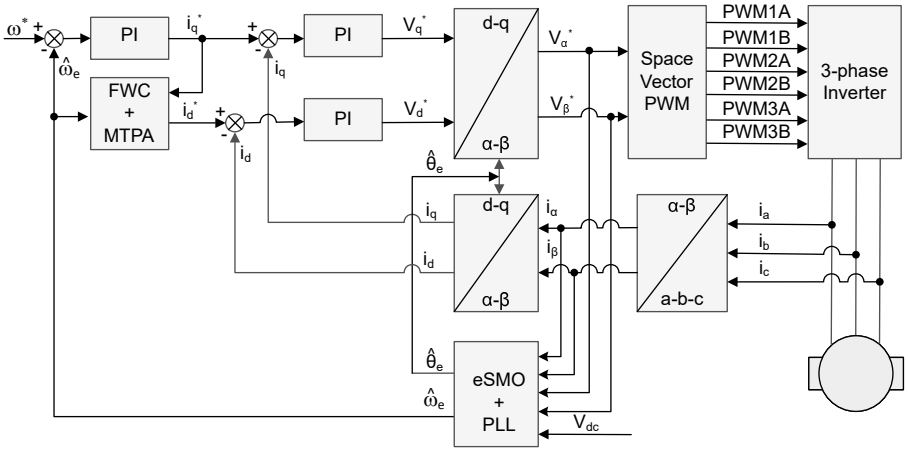


图 2-16. IPMSM 系统的无传感器 FOC 结构

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 α - β 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 46 所示。通用 PMSM 的动态模型可以在 d-q 转子坐标系中写为：

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} R_s + pL_d & -\omega_e L_q \\ \omega_e L_d & R_s + pL_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_e \lambda_{pm} \end{bmatrix} \quad (46)$$

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。

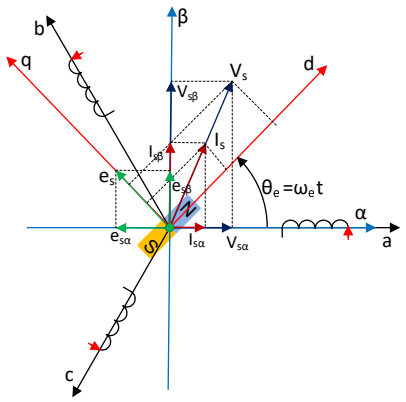


图 2-17. PMSM 建模坐标系的定义

通过使用如图 2-17 所示的 Park 逆变换，PMSM 的动力学可以在 α - β 静止坐标系中建模为：

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \begin{bmatrix} R_s + pL_d & \omega_e\left(L_d - L_q\right) \\ -\omega_e\left(L_d - L_q\right) & R_s + pL_q \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \begin{bmatrix} e_\alpha \\ e_\beta \end{bmatrix} \tag{47}$$

其中 e_α 和 e_β 是 α - β 轴上扩展电动势 (EEMF) 的分量，可以定义为：

$$\begin{bmatrix} e_\alpha \\ e_\beta \end{bmatrix} = \left(\lambda_{pm} + \left(L_d - L_q\right) i_d\right) \omega_e \begin{bmatrix} -\sin(\theta_e) \\ \cos(\theta_e) \end{bmatrix} \tag{48}$$

根据方程式 47 和方程式 48，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 47 改写为状态公式：

$$\begin{bmatrix} \dot{i}_\alpha \\ \dot{i}_\beta \end{bmatrix} = \frac{1}{L_d} \begin{bmatrix} -R_s & -\omega_e\left(L_d - L_q\right) \\ \omega_e\left(L_d - L_q\right) & -R_s \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \frac{1}{L_d} \begin{bmatrix} V_\alpha - e_\alpha \\ V_\beta - e_\beta \end{bmatrix} \tag{49}$$

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：

$$S(x) = \begin{bmatrix} \hat{i}_\alpha - i_\alpha \\ \hat{i}_\beta - i_\beta \end{bmatrix} = \begin{bmatrix} \tilde{i}_\alpha \\ \tilde{i}_\beta \end{bmatrix} \tag{50}$$

其中 $\hat{i}_\alpha$ 和 $\hat{i}_\beta$ 是估算的电流，上标 **^** 表示变量为估算值，上标 “~ ” 表示变量为变量误差，即观测值与实际测量值之间的差异。

2.4.2.2.1.2 IPMSM 的 ESMO 设计

图 2-18 显示了集成在 SMO 中的传统 PLL。

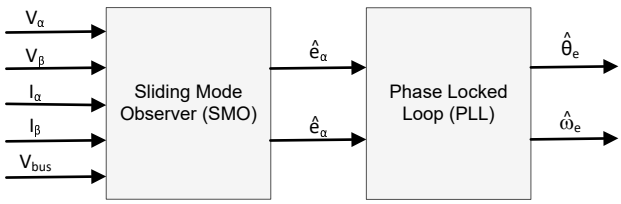


图 2-18. 包含用于 PMSM 的 PLL 的 eSMO 方框图

构建了传统的降阶滑模观测器，其数学模型如方程式 51 所示，方框图如图 2-19 所示。

$$\begin{bmatrix}\dot{\hat{i}}_\alpha \\ \dot{\hat{i}}_\beta\end{bmatrix} = \frac{1}{L_d}\begin{bmatrix}-R_s & -\hat{\omega}_e\left(L_d - L_q\right) \\ \hat{\omega}_e\left(L_d - L_q\right) & -R_s\end{bmatrix}\begin{bmatrix}\hat{i}_\alpha \\ \hat{i}_\beta\end{bmatrix} + \frac{1}{L_d}\begin{bmatrix}V_\alpha - \hat{e}_\alpha + z_\alpha \\ V_\beta - \hat{e}_\beta + z_\beta\end{bmatrix} \tag{51}$$

其中 z_α 和 z_β 是滑模反馈分量，其定义为：

$$\begin{bmatrix}z_\alpha \\ z_\beta\end{bmatrix} = \begin{bmatrix}k_\alpha sign\left(\hat{i}_\alpha - i_\alpha\right) \\ k_\beta sign\left(\hat{i}_\beta - i_\beta\right)\end{bmatrix} \tag{52}$$

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，以保证 SMO 的稳定运行， k_α 和 k_β 应足够大，以保持 $k_\alpha > \max(|e_\alpha|)$ 和 $k_\beta > \max(|e_\beta|)$。

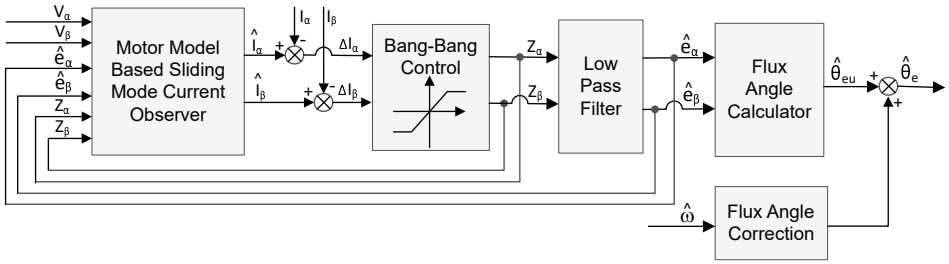


图 2-19. 传统滑模观测器的方框图

α - β 轴上的 EEMF 估算值 $(\hat{e}_\alpha, \hat{e}_\beta)$ 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_α:

$$\begin{bmatrix}\hat{e}_\alpha \\ \hat{e}_\beta\end{bmatrix} = \frac{\omega_c}{s + \omega_c}\begin{bmatrix}z_\alpha \\ z_\beta\end{bmatrix} \tag{53}$$

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，其定义如下：

$$\hat{\theta}_e = -\tan^{-1}\left(\frac{\hat{e}_\alpha}{\hat{e}_\beta}\right) \tag{54}$$

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，定义为：

$$\Delta\theta_e = -\tan^{-1}\left(\frac{\omega_e}{\omega_c}\right) \tag{55}$$

这样使用 SMO 方法估算的转子位置就为：

$$\hat{\theta}_e = -\tan^{-1}\left(\frac{\hat{e}_\alpha}{\hat{e}_\beta}\right) + \Delta\theta_e \tag{56}$$

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 51 的时间离散系统矩阵由方程式 57 给出：

$$\begin{bmatrix}\hat{i}_\alpha\left(n+1\right) \\ \hat{i}_\beta\left(n+1\right)\end{bmatrix} = \begin{bmatrix}F_\alpha \\ F_\beta\end{bmatrix}\begin{bmatrix}\hat{i}_\alpha\left(n\right) \\ \hat{i}_\beta\left(n\right)\end{bmatrix} + \begin{bmatrix}G_\alpha \\ G_\beta\end{bmatrix}\begin{bmatrix}V_\alpha^*(n) - \hat{e}_\alpha(n) + z_\alpha(n) \\ V_\beta^*\left(n\right) - \hat{e}_\beta\left(n\right) + z_\beta\left(n\right)\end{bmatrix} \tag{57}$$

其中矩阵 $[F]$ 益 $[G]$ 由方程式 58 和方程式 59 给出：

$$\begin{bmatrix} F_\alpha \\ F_\beta \end{bmatrix} = \begin{bmatrix} e^{-\frac{R_S}{L_d}} \\ e^{-\frac{R_S}{L_q}} \end{bmatrix} \tag{58}$$

$$\begin{bmatrix} G_\alpha \\ G_\beta \end{bmatrix} = \frac{1}{R_S}\begin{bmatrix} 1-e^{-\frac{R_S}{L_d}} \\ 1-e^{-\frac{R_S}{L_q}} \end{bmatrix} \tag{59}$$

方程式 53 的时间离散形式由方程式 60 给出：

$$\begin{bmatrix} \hat{e}_\alpha(n+1) \\ \hat{e}_\beta(n+1) \end{bmatrix} = \begin{bmatrix} \hat{e}_\alpha(n) \\ \hat{e}_\beta(n) \end{bmatrix} + 2\pi f_c \begin{bmatrix} z_\alpha(n) - \hat{e}_\alpha(n) \\ z_\beta(n) - \hat{e}_\beta(n) \end{bmatrix} \tag{60}$$

2.4.2.2.1.3 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 2.4.2.2.1.2 中说明了与 SMO 配合使用的 PLL 结构。反电动势估算 $\hat{e}_\alpha$ 和 $\hat{e}_\beta$ 可与 PLL 模型配合使用来估算电机角速度和位置，如图 2-20 所示。

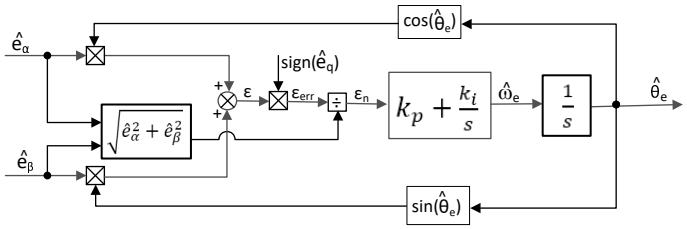


图 2-20. 锁相环位置跟踪器的方框图

由于 $e_\alpha = E\cos(\theta_e)$, $e_\beta = E\sin(\theta_e)$ 和 $E = \omega_e\lambda_{pm}$，位置误差可定义为：

$$\varepsilon = \hat{e}_\beta\cos\left(\hat{\theta}_e\right) - \hat{e}_\alpha\sin\left(\hat{\theta}_e\right) = E\sin(\theta_e)\cos\left(\hat{\theta}_e\right) - E\cos(\theta_e)\sin\left(\hat{\theta}_e\right) = E\sin\left(\theta_e - \hat{\theta}_e\right) \tag{61}$$

其中 E 是 EEMF 的幅度，与电机转速成正比 ω_e.当 $\left(\theta_e - \hat{\theta}_e\right) < \frac{\pi}{2}$，方程式 61 可以简化为

$$\varepsilon = E\left(\theta_e - \hat{\theta}_e\right) \tag{62}$$

可以进一步得到 EEMF 归一化后的位置误差：

$$\varepsilon_n = \theta_e - \hat{\theta}_e \tag{63}$$

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 2-21 所示。PLL 的闭环传递函数可表示为：

$$\frac{\hat{\theta}_e}{\theta_e} = \frac{k_p s + k_i}{s^2 + k_p s + k_i} = \frac{2\xi\omega_n s + \omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{64}$$

其中 k_p 和 k_i 是标准 PI 调节器的比例增益和积分增益，其固有频率 ω_n 和阻尼比 ξ 已给定：

$$k_p = 2\xi\omega_n : \qquad k_i = \omega_n^2 \tag{65}$$

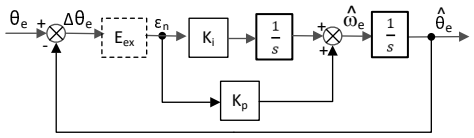


图 2-21. 锁相环位置跟踪器的简化方框图

2.4.2.3 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、高效率和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 66 和方程式 67 所示。

$$v_d = L_d\frac{di_d}{dt} + R_s i_d - p\omega_m L_q i_q \tag{66}$$

$$v_q = L_q\frac{di_q}{dt} + R_s i_q + p\omega_m L_d i_d + p\omega_m \psi_m \tag{67}$$

图 2-22 显示了 IPM 同步电机的动态等效电路。

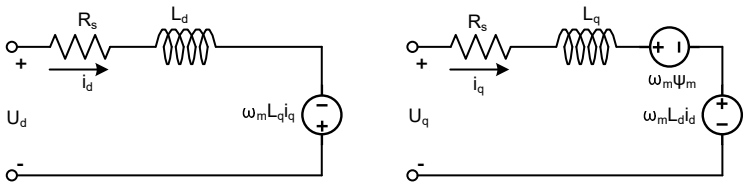


图 2-22. IPM 同步电机的等效电路

IPMSM 产生的总电磁扭矩可以由方程式 68 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。

$$T_e = \frac{3}{2}p\left[\psi_m i_q + \left(L_d - L_q\right)i_d i_q\right] \tag{68}$$

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 69 和方程式 70 进行表示。

$$I_a = \sqrt{i_d^2 + i_q^2} \le I_{max} \tag{69}$$

$$V_a = \sqrt{v_d^2 + v_q^2} \le V_{max} \tag{70}$$

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 71 中所示的值。

$$\sqrt{v_d^2 + v_q^2} \le v_{max} = \frac{v_{dc}}{\sqrt{3}} \tag{71}$$

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 72，如下所示。

$$\sqrt{L_d^2\left(i_d+\frac{\psi_{pm}}{L_d}\right)^2+L_q^2 i_q^2} \le \frac{V_{max}}{\omega_m} \tag{72}$$

方程式 69 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 71 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 2-23 所示。

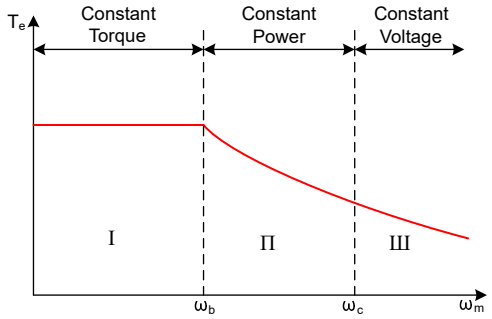


图 2-23. IPMSM 控制工作区域

1. 恒定扭矩区域：可以在该工作区域内实施 MTPA，以确保产生最大扭矩。
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 68，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之间的差值的乘积成正比。

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 IPMSM 将利用电机的磁阻扭矩，也应控制 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。

$$I_s=\sqrt{i_d^2+i_q^2} \tag{73}$$

$$I_d=I_s\cos\beta \tag{74}$$

$$I_q=I_s\sin\beta \tag{75}$$

其中 β 是同步 (d-q) 坐标系中的定子电流角度。方程式 68 可以表示为方程式 76，其中 I_s 替换了 i_d 和 i_q。

方程式 76 表明电机扭矩取决于定子电流矢量的角度，因此

$$T_e=\frac{3}{2}pI_s\sin\beta\left[\psi_m+(L_d-L_q)\,I_s\cos\beta\right] \tag{76}$$

，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 77 所示）时，可以找到 MTPA 点。

$$\frac{dT_e}{d\beta}=\frac{3}{2}p\left[\psi_m I_s\cos\beta+(L_d-L_q)\,I_s^2\cos 2\beta\right]=0 \tag{77}$$

接下来，可以通过方程式 78 得出 MTPA 控制的电流角度。

$$\beta_{mtpa}=\cos^{-1}\frac{-\psi_m+\sqrt{\psi_m^2+8*\left(L_d-L_q\right)^2*I_s^2}}{4*\left(L_d-L_q\right)*I_s} \tag{78}$$

因此，可以使用 MTPA 控制的电流角度通过方程式 79 和方程式 80 来表示有效的 d 轴和 q 轴基准电流。

$$I_d = I_s{*}\cos\beta_{mtpa} \tag{79}$$

$$I_q = I_s{*}\sin\beta_{mtpa} \tag{80}$$

不过，如方程式 78 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最佳 MTPA 点。为了提高电机驱动器的效率，应在线估算 d 轴和 q 轴电感，但参数 L_d 和 L_q 不易于在线测量，并且受饱和效应的影响。稳健的查找表 (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其 d-q 电流的函数，如方程式 81 和方程式 82 所示。

$$L_d = f_1\left(i_d, i_q\right) = f_1\left(i_d\right) \tag{81}$$

$$L_q = f_2\left(i_q, i_d\right) = f_2\left(i_q\right) \tag{82}$$

为了通过简化方程式 78 来减轻 ISR 计算负担，基于电机参数的常数 K_{mtpa} 改为用方程式 84 表示，其中 K_{mtpa} 在后台循环中使用更新的 L_d 和 L_q进行计算。

$$K_{mtpa} = \frac{\psi_m}{4{*}\left(L_q - L_d\right)} = 0.25{*}\frac{\psi_m}{\left(L_q - L_d\right)} \tag{83}$$

$$\beta_{mtpa} = cos^{-1}\left(K_{mtpa}/I_s - \sqrt{\left(K_{mtpa}/I_s\right)^2 + 0.5}\right) \tag{84}$$

第二个中间变量 G_{mtpa} 由方程式 85 进行表示，用于进一步简化计算。使用 G_{mtpa}，MTPA 控制的角度 β_{mtpa} 可以通过方程式 86 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}。

$$G_{mtpa} = K_{mtpa}/I_s \tag{85}$$

$$\beta_{mtpa} = cos^{-1}\left(G_{mtpa} - \sqrt{G_{mtpa}^2 + 0.5}\right) \tag{86}$$

在所有情况下，都可以通过作用于直轴电流 i_d来减弱磁通量以扩大可达到的转速范围。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，可通过 d 轴电枢反应引起的退磁效应来削弱气隙磁通。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 69 和方程式 70 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 72 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 2-24 显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。在电压幅度达到其限制之前，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。

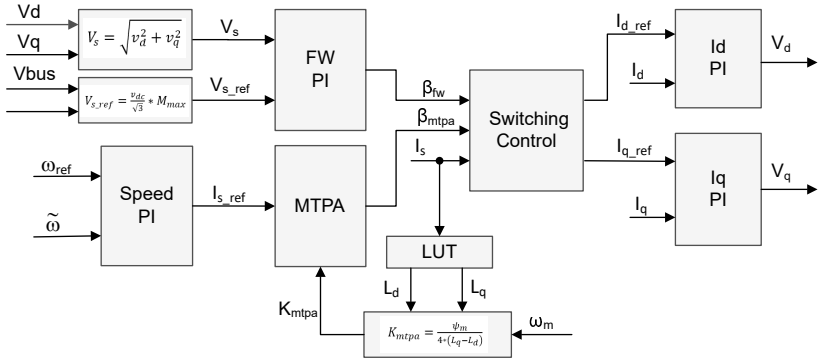


图 2-24. 弱磁和每安培最大扭矩控制的方框图

图 2-13 和图 2-15 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 2-25 所示。

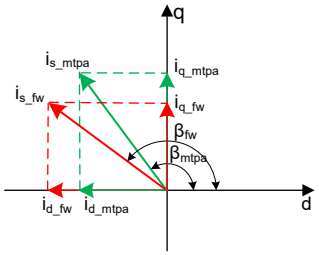


图 2-25. FW 和 MTPA 期间 IPMSM 的电流相量图

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 74 和方程式 75 所示。可以根据下面的方程式 87 和方程式 88 来选择电流角度。

$$\beta = \beta_{fW}\, if\ \beta_{fW} > \beta_{mtpa} \tag{87}$$

$$\beta = \beta_{mpta}\, if\ \beta_{fW} < \beta_{mtpa} \tag{88}$$

图 2-26 是显示在主循环和中断中运行采用 FW 和 MPTA 的 InstaSPIN-FOC 所需步骤的流程图。

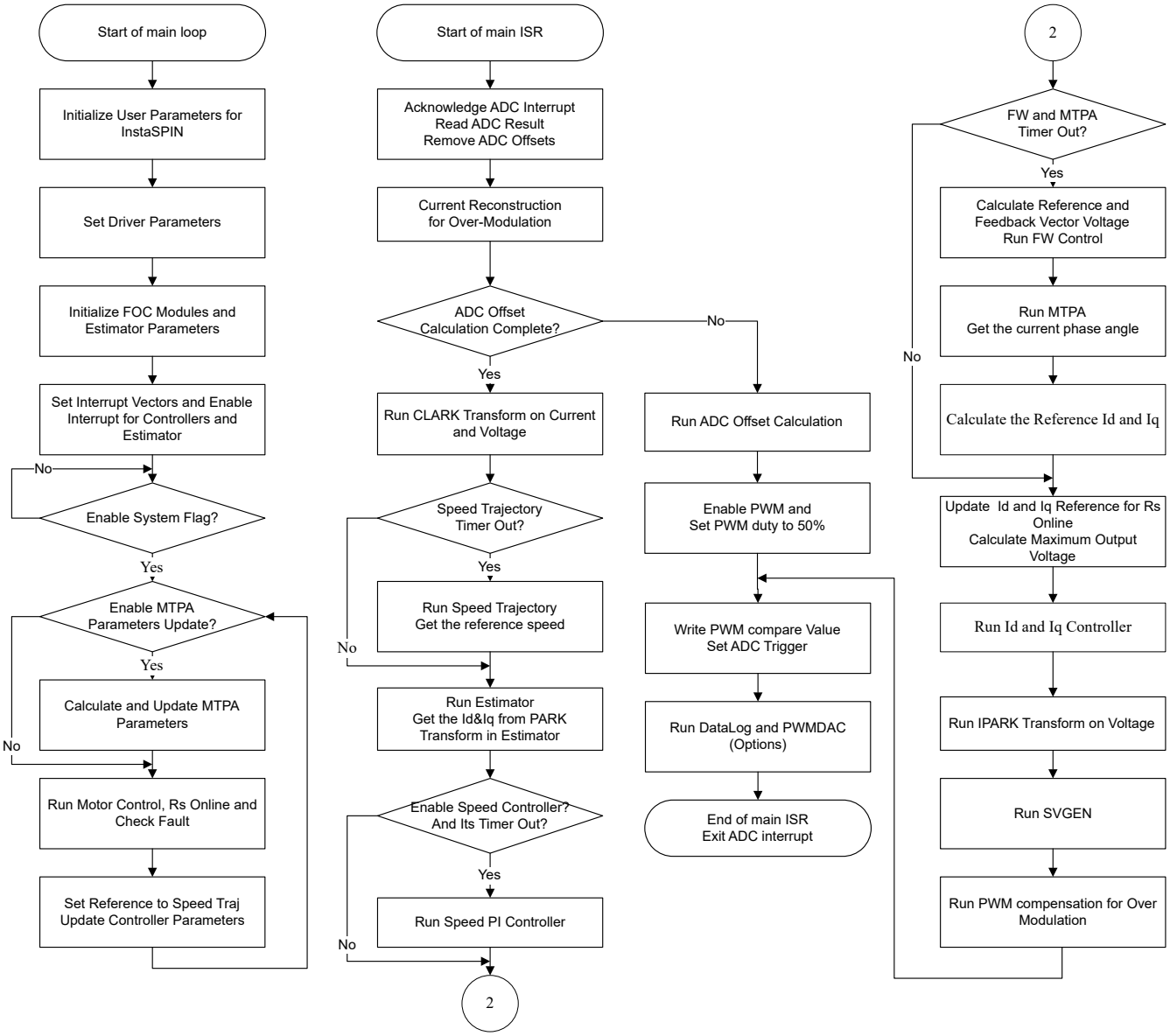


图 2-26. 采用 FW 和 MTPA 的 InstaSPIN-FOC 工程的流程图

2.4.2.4 具有自动振动补偿功能的压缩机驱动器

振动和噪声可能成为空调压缩机应用中的一个问题，因为它们会导致不良的最终用户体验，以及由于应力而产生的机械故障。压缩机应用包含脉动负载，这取决于机械角度（如图 2-27 所示），可能会导致电机振动和可闻噪声。产生振动和噪声的原因有多种，主要原因是负载特性产生的振动。本指南还将介绍一种新的动态和自适应补偿方法，详细说明其工作原理和所需的最小调整。

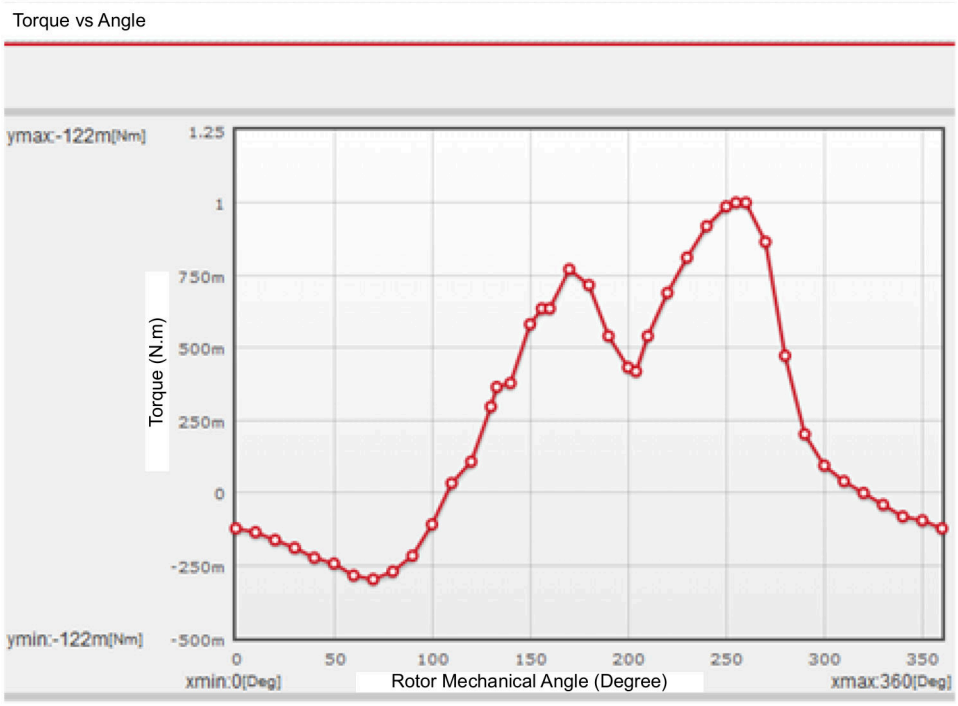


图 2-27. 负载扭矩波形

振动补偿算法在电机运行时学习负载曲线，当转速控制器尝试纠正这些负载变化时以及学习负载后，该算法将用于提取与机械角度相关的负载信息，并使用该信息作为转速控制器中的前馈。如图 2-28 所示，在 FOC 系统中添加了一个称为*动态振动补偿*的新块，用于学习扭矩负载，以允许向转速控制器添加前馈项（采用转速控制器所生成输出的求和点形式）。

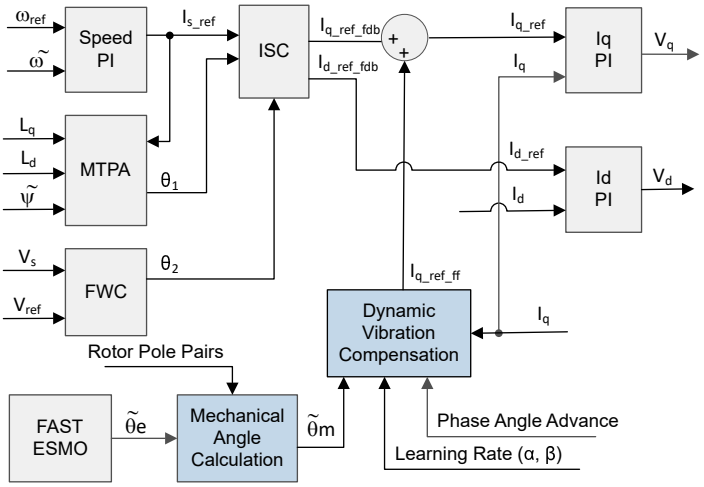


图 2-28. 具有振动补偿功能、基于 FOC 的电机驱动器方框图

该算法需要四个主要块才能工作：

1. 具有前馈输入的转速控制器
2. 用于保存学习曲线的表
3. 一种基于索引提取该表的特定成员的方法
4. 计算用于更新学习曲线的索引和用于从该表中提取值的高级索引

该算法能够根据两个输入动态学习负载曲线：

1. 电角度信息。从图 2-28 中显示的左下角开始，振动补偿模块所需的第一个输入是机械角度。这是根据电角度和极对数计算得出的。机械角度不需要与电角度同步。例如，机械角度零在物理上不必是电角度零。这是因为振动补偿模块将根据提供的机械角度学习负载，而与相对于轴物理位置的机械角度大小无关。
2. 测量的电流值（对于 FOC 为 Iq），决定电机的扭矩。

然后实现振动补偿模块。在该实现中，模块需要四个参数。

1. 相位超前。这是相对于机械角度访问所学负载的方式。如果相位超前为零，则 IqRef_ff 上的加载值将对应于提供的机械角度。如果相位超前为 10，则 IqRef_ff 上的加载值将对应于机械角度加上 10 度机械角度（范围为 0 至 360 度）。
2. 学习速度。这个参数值的范围为 0 至 1，它本质上反映负载学习有多快（抗噪性能较低）或者有多慢（抗噪性能较高）。
3. 点数。这是用于学习曲线的补偿表点数。
4. 极对。这是电机的极数。

然后是转速控制器和 Iq 控制器之间的求和点。这时使用振动补偿模块的输出，以帮助转速控制器使用该项。该技术也称为前馈，因为可以根据提供的机械角度预先知道负载。

振动补偿模块获知负载后，转速控制器将校正负载变化的瞬态，这与自然机械负载和机械角度之间的关系无关，振动补偿模块已对其进行了补偿。为了说明振动补偿模块如何提供帮助，让我们看看下图，其中显示了禁用振动补偿时的转速控制器输出。很明显，转速控制器增益需要足够高，以跟踪电机在每个周期旋转时的负载变化。

调整学习速度

可以根据两个因素来调整学习速度。第一考虑因素是用户想要多快地学习曲线，第二个考虑因素是学习曲线的输入中有多少噪声。第二个考虑因素很重要，因为噪声不仅来自电流检测方法本身，还来自系统中的微小机械扰动，它们不是周期性的，我们希望将其滤除，而不将其包含在我们的补偿表中。如果学习速度过低，那么对于特定的应用而言，学习时间可能会过长，因此需要做出权衡。

调整相位角

在离散系统中，在向电机输出电压时存在多个延迟，并且还存在与检测电流相关的延迟。例如，在处理器中实现 FOC 系统时，输出电压通常会通过具有延迟的脉宽调制器 (PWM)。利用该相位超前参数，可以通过从学习表中提供以表位置为单位的数字来微调该延迟，以便可以将适当的输出应用于转速控制器的前馈输入。调整该参数的最简单方法是在应用动态补偿后查看转速变化，然后调整该值以实现最小的转速变化。

2.4.2.5 具有快速启动功能的风扇驱动器

快速启动功能使驱动器能够确定旋转电机的转速和方向，并以该转速和方向开始输出电压和频率。如果没有快速启动功能，那么驱动器将以零伏和零转速开始其输出，并尝试增加至命令的速度。如果负载的惯性或旋转方向要求电机产生很大的扭矩，则可能会导致过大的电流，并且可能会在驱动器上发生过流跳闸。这些问题可以通过快速启动加以解决。

快速启动是指以除**零**以外的任何速度启动控制能力，这是空调应用中用于风扇驱动的重要功能。

当电机以正常模式启动时，控制器最初应用 0Hz 的频率并增加至所需的频率。如果驱动器在该模式下启动，并且电机已经以非零频率旋转，则会产生大电流。如果电流限制器反应不够快，则可能会导致过流跳闸。即使电流限制器足够快，可以防止过流跳闸，发生同步和电机达到其所需频率也可能需要不可接受的时长。此外，会在应用上施加更大的机械应力。

在快速启动模式下，驱动器对启动命令的响应是与电机的转速（频率和相位）和电压同步。然后电机加速至命令的频率。该过程可防止过流跳闸并显著减少电机达到其命令的频率所需的时间。由于驱动器以其旋转速度与电机同步并增加至适当的转速，因此几乎不存在机械应力或不存在机械应力。

快速启动功能实现了一种搜索转子转速的算法。该算法搜索与施加到电机上的励磁电流相对应的电机电压。

当电机旋转时，可以通过 BEMF 电压估算转速和位置信息。由于在 InstaSPIN 驱动器中测量定子电压，因此通过开关逆变器可以轻松获得转速和位置。向电机施加零扭矩电流并测量产生的电流和定子电压，然后 InstaSPIN-FOC 模块使用这些信号来估算转子位置和转速。

图 2-29 显示了具有快速启动功能的 FOC 的方框图，快速启动模块输出一个标志来启用或禁用转速闭环控制。设置了零基准扭矩电流，在快速启动功能运行时速度 PI 控制器输出被禁用。

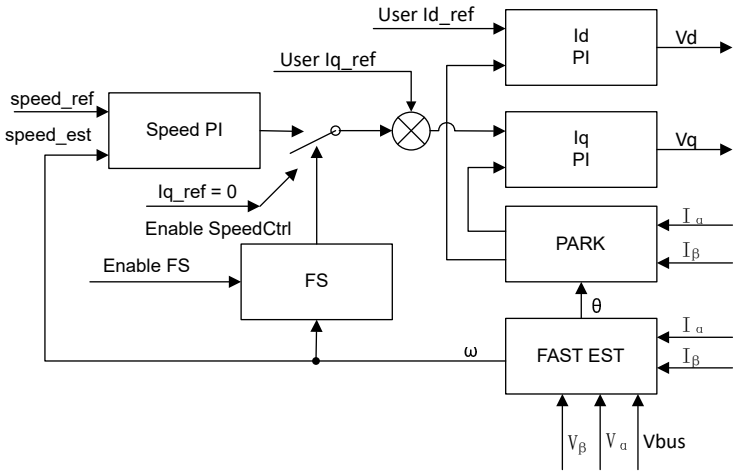


图 2-29. 快速启动控制方框图

如图 2-30 所示，模块例程禁用转速闭环控制，将基准 Iq 设置为零，并在启动期间启用 FOC 模块运行电机。在测量相电流和电压后，该例程运行 InstaSPIN-FOC 并且可以估算实际的电机转速。程序重新启用速度闭环控制，并在快速启动完成后设置转速基准值。重启期间的电流波形如图 3-32 所示。

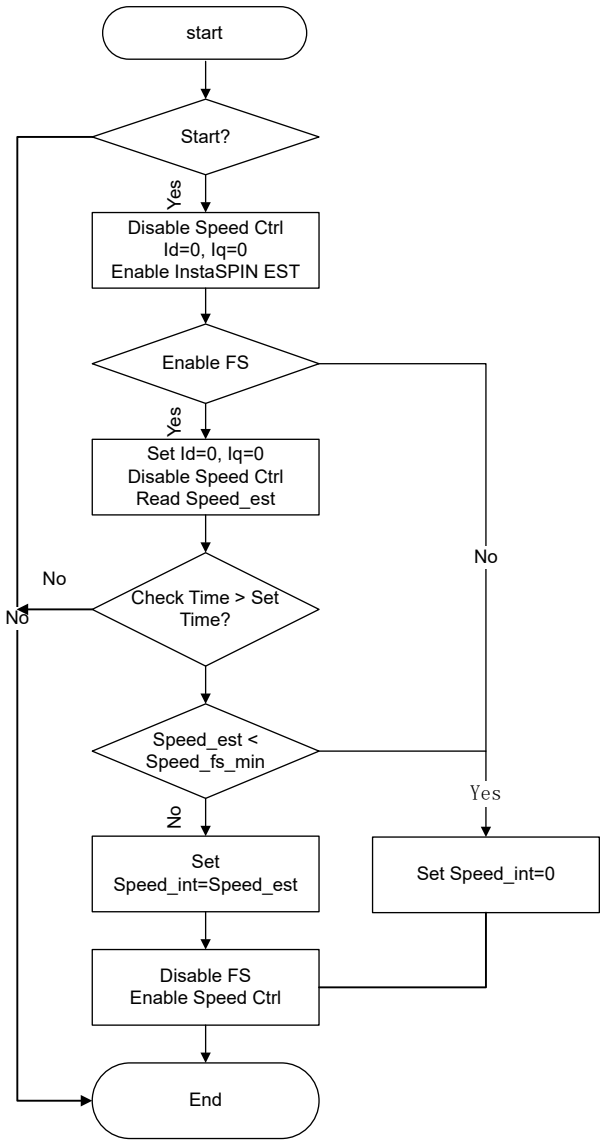


图 2-30. 快速启动模块程序流程图

2.4.2.6 电机驱动器的硬件必要条件

用于控制电机的算法利用电机条件的采样测量值，包括直流总线电源电压、每个电机相位上的电压、每个电机相位的电流。需要正确设置一些与硬件相关的参数才能正确识别电机并使用磁场定向控制 (FOC) 有效地运行电机。以下各节说明如何计算采用 FAST 或 eSMO 的压缩机和风扇电机控制的电流标度值、电压标度值和电压滤波器极点。

2.4.2.6.1 电机电流反馈

支持两种技术来测量电机的相电流。

- 单分流器电流检测
- 三分流器电流检测

可以在工程的构建配置中选择这两种电流检测技术中的任何一种。“HVAC_REV3P2_3SC_LIB”构建配置支持三分流器电流检测方法，“HVAC_REV3P2_1SC_LIB”支持单分流器电流检测方法，如节 3.2.2 中所述。

2.4.2.6.1.1 采用三分流器的电流检测

在每一个（压缩机）或三个（风扇）PWM 周期内，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。为了测量电机相位的双向电流（即正负电流），以下电路需要 1.65V 的基准电压。该基准偏移由电压跟

随器生成，如图 2-31 所示。该 1.65V 基准电压用于风扇和压缩机电机相位电流以及 PFC 交流电压反馈检测电路。在该版本的硬件中，压缩机（电机 1）、风扇（电机 2）和 PFC 分别具有偏移基准，但是这三个采样电路可以在设计中共享相同的偏移基准以节省成本。

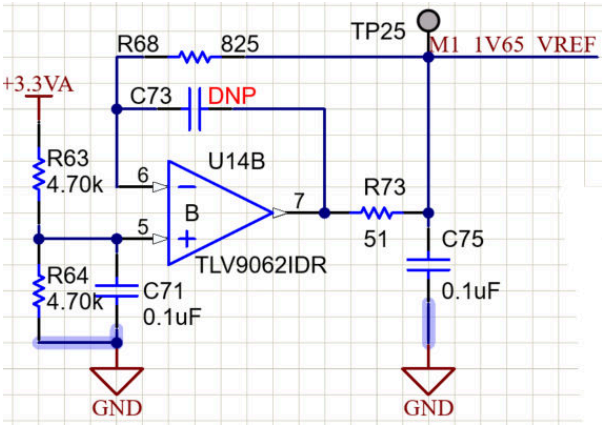


图 2-31. 3.3V 输入电路提供的 1.65V 基准

图 2-32 显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 ADC 输入范围中心的偏移。该电路用于压缩机和风扇的三相 PMSM 的每一相。方程式 89 给出了该电路的传递函数。

$$V_{OUT} = V_{OFFSET} + (I_{IN} \times R_{SHUNT} \times G_i) \tag{89}$$

其中 R_{shunt} = 0.01（Ω）和 V_{offset} = 1.65 (V)

利用计算出的电阻值，可得到图 2-35 所示的检测电路，G_i 由方程式 90 给出。

$$G_i = \frac{R_{fb}}{R_{in}} = \frac{R45}{(R54 + R39)} = \frac{7.5K}{(20 + 825)} = 8.876 \tag{90}$$

微控制器可测量的最大峰峰值电流由方程式 91 给出。

$$I_{scale\_max} = \frac{V_{ADC\_max}}{R_{SHUNT} \times G_i} = \frac{3.3}{0.01 \times 8.876} = 37.18A \tag{91}$$

即峰峰值为 ±18.59A。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         (37.18f)
```

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。在该硬件板配置中，分流电阻器的负引脚接地，同时与运算放大器的同相引脚连接。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 motor1_drive.c 中的以下代码片段所示：

```
// define the sign of current feedback based on hardware board
adcData[MTR_1].current_sf = -userParams[MTR_1].current_sf;
```

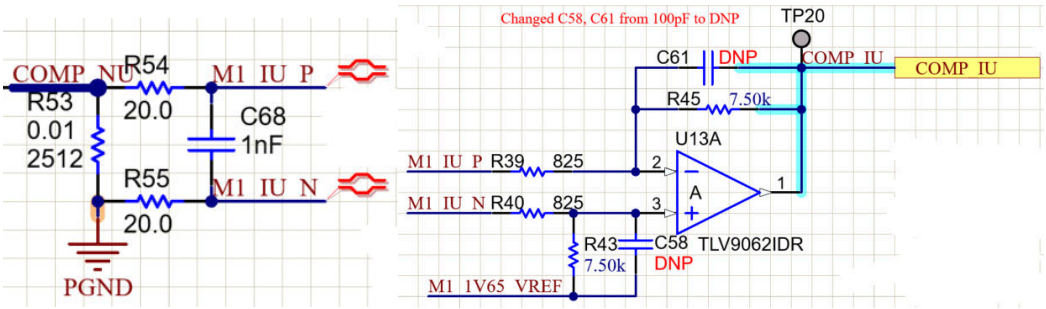


图 2-32. 采用三分流器的电机电流检测

对风扇电机执行相同的计算步骤，并在 user_mtr2.h 文件中设置标度值。

2.4.2.6.1.2 采用单分流器的电流检测

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。应用手册使用单一直流链路分流器的 PMSM 无传感器 FOC (spract7) 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 2-33 所示。

1. 移除分流电阻器 R53 和 R65，只保留一个分流电阻器 R56 来检测压缩机驱动器的和直流链路电流。
2. 使用粗导线将 NU、NV 和 NW 引脚连接在一起。

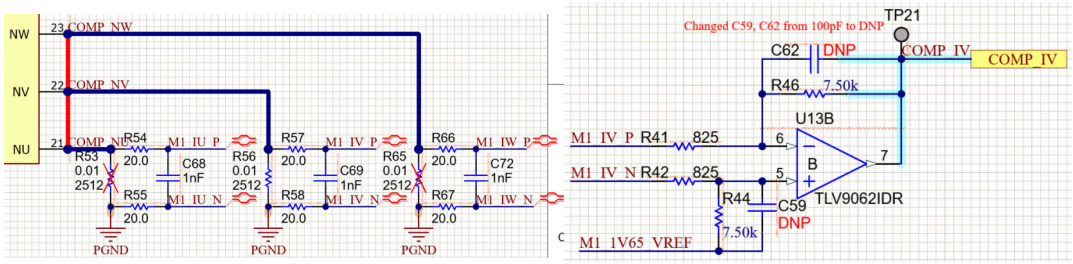


图 2-33. 采用单分流器的电机电流检测电路

直流链路电流不是双向信号，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 2-34 所示。将 R64 从 4.70k (Ω) 更改为 47.5K (Ω) (电阻器精度为 1%) ，以实现基准电压。

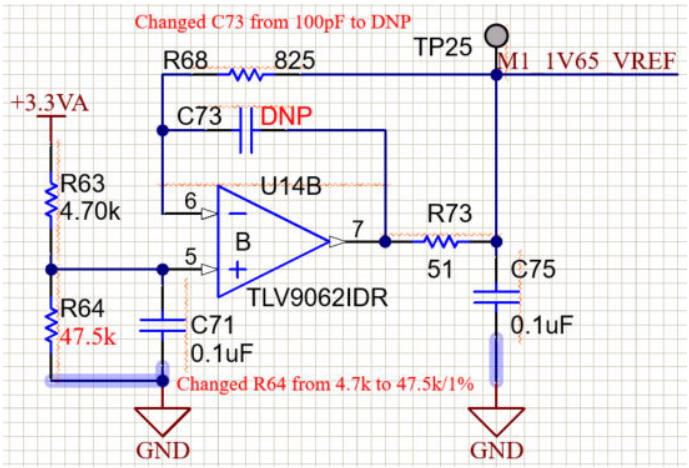


图 2-34. 3.3V 输入电路提供的直流偏移基准

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。

2.4.2.6.2 电机电压反馈

FAST 估算器需要电压反馈，以在最宽的速度范围内实现最佳性能，相电压直接从电机相位测量，而不是使用软件估算。eSMO 依靠软件估算值来表示电压相位，而不使用电机相位电压检测电路。此软件值 (USER_ADC_FULL_SCALE_VOLTAGE_V) 取决于感测电机相电压反馈的电路。图 2-35 显示了如何使用基于电阻分压器的电压反馈电路对电机电压进行滤波和调整以适应 ADC 输入范围。类似的电路用于测量全部三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通过方程式 92 进行计算。

$$V_{FS} = V_{ADC\_FS} \times G_v = 3.3V \times 122.46 = 404.13V \tag{92}$$

其中 G_v 是衰减因子，可通过以下公式进行计算

$$G_V = \frac{(R59 + R60 + R61 + R62)}{R62} = \frac{(332K + 332K + 332K + 8.2K)}{8.2K} = 122.46 \tag{93}$$

对于该电压反馈电路，在 user_mtr1.h 中进行以下设置：

```
//! \brief Defines the maximum voltage at the AD converter
#define USER_M1_ADC_FULL_SCALE_VOLTAGE_V         (404.13f)
```

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。滤波器的电压应足够低，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通过下面的方程式 94 进行计算：

$$f_{filter\_pole} = \frac{1}{(2 \times \pi \times R_{Parallel} \times C)} = 405.15 \quad Hz \tag{94}$$

$$Where,$$

$$C = 47nF$$

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

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           (405.15f)
```

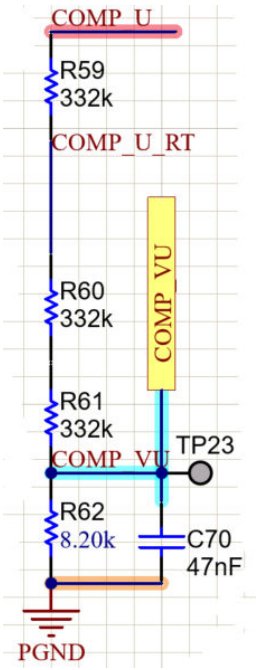


图 2-35. 电机电压检测电路

3 硬件、软件、测试要求和测试结果

3.1 入门硬件

本节详细介绍了设计电路板和软件测试和检验所需的设备、测试装置和过程说明。

3.1.1 硬件板概述

图 3-1 概述了典型的双电机控制，其中 PFC 系统由交流电源供电。PFC 级可对输入交流电流进行波形整形，并为压缩机和风扇电机驱动器的三相逆变器提供可调直流电源。

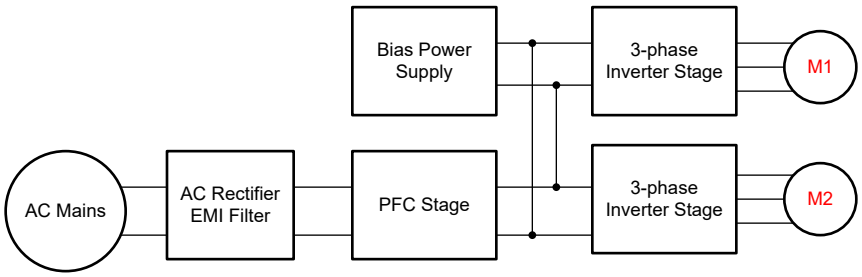


图 3-1. TIDM-02010 的硬件板方框图

电机控制板具有可实现完整电机驱动系统的功能组。以下是电路板上的块及其功能的列表，图 3-2 显示了电路板顶视图和 TIDM-02010 PCB 的不同块。

- 电源线输入滤波器
  - 截止频率为 2.2kHz 的双共模滤波器。
- 数字交错式 PFC
  - 最大功率高达 1.5kW。
  - 具有 72kHz 开关频率的两相交错式升压 PFC
  - 具有 UCC27517A 高速栅极驱动器的功率 MOSFET
- 用于电机 1（压缩机）的三相逆变器
  - 功率高达 1.5kW 的三相逆变器支持 PMSM 或 IPM
  - 6 kHz 开关频率

  - 用于电流检测的 3 分流器
- 用于电机 2（风扇）的三相逆变器
  - 功率高达 150W 的三相逆变器支持 PMSM 或 IPM
  - 12kHz 或 18kHz 开关频率
  - 用于电流检测的 3 分流器
- 控制
  - 采用 64 引脚 LQFP 封装的单个 TMS320F280025C 或 TMS320F28039C 系列 MCU
  - 具有 FPU 和 TMU 的 100MHz 32 位 CPU
  - 模拟信号的放大和输入滤波器
- 用于系统功能的阀门和继电器驱动器
  - 用于空调阀门的继电器驱动器
  - 用于电子膨胀阀的步进电机驱动器
- 辅助电源
  - 板载电源 +3.3V、+12V 和 +15V

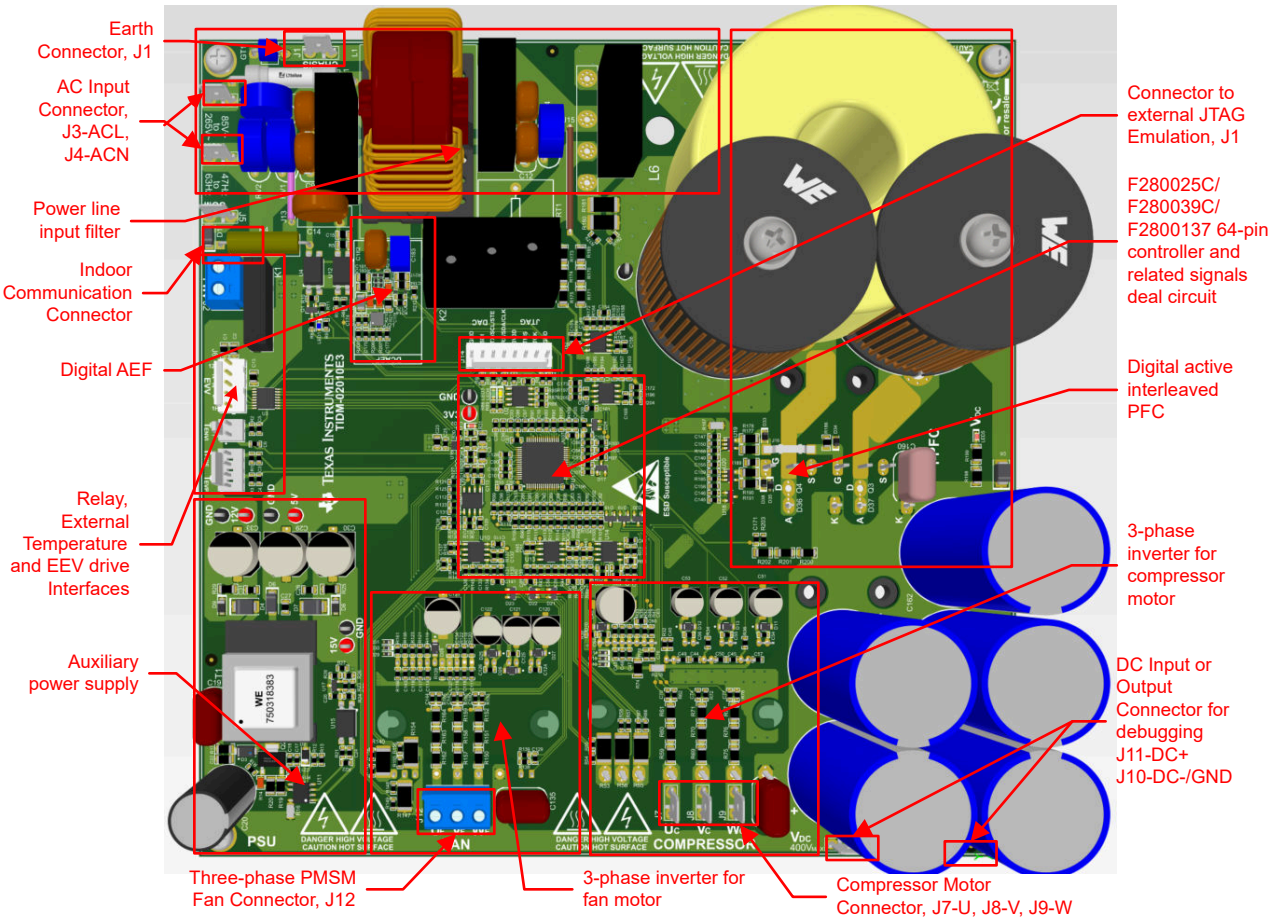


图 3-2. 采用 PFC 的双电机控制参考设计板的布局

备注

F28002x、F28003x 和 F280013x 在该 64 引脚封装上完全引脚对引脚兼容，因此同一 PCB 板可用于这三款 MCU 之一。

TI 建议在使用该板时采取以下预防措施：

- 电路板通电时，请勿触摸电路板的任何部分或连接到电路板的元件。
- 使用交流电源/墙上电源为套件供电。TI 建议使用隔离交流电源。

- 通电时请勿触摸电路板、套件或其组件的任何部分。（尽管电源模块散热器与电路板隔离，但高电压开关会在散热器器身上产生一些电容耦合电压。）
- 控制接地可能很热。

3.1.2 测试条件：

供用户测试参考设计软件

- PFC 操作
  - 对于输入端，电源电压范围必须为 165V VAC 至 265V VAC。将输入交流电源的输入电流限制设置为 10A 以进行全功率测试，但在初始电路板启动期间先使用较低的电流限制。
  - 对于输出端，使用电子可变负载或可变电阻负载，其额定电压不得低于 400V，负载电流必须在 0mA 至 5 A 范围内变化。
- 电机 1（压缩机）驱动操作
  - 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设置为 10A 或将直流电源设置为 6.5A，但在初始电路板启动期间先使用较低的电流限制。
  - 对于输出，采用带测力计的三相 PMSM
- 电机 2（风扇）驱动操作
  - 对于输入，如果使用交流电源，则电源电压 (VIN) 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设置为 3A 或将直流电源设置为 1.5A，但在初始电路板启动期间先使用较低的电流限制。
  - 对于输出，采用带测力计的三相 PMSM

3.1.3 电路板检验所需测试设备

- 隔离式交流源
- 单相功率分析仪
- 数字示波器
- 万用表
- 电子或电阻负载
- 直流电源
- 1.5Kw 和 150W 三相 PM 同步电机
- 测力计
- 三相功率分析仪

3.1.4 测试设置

图 3-2 显示了这些块和连接器在板上的位置。按照以下步骤设置硬件。

1. 将 JTAG 仿真器连接至连接器 J14 以对 C2000 器件进行调试或编程。

   在该 HVAC 硬件板中，外部仿真器必须使用具有 2 个引脚（TMS 和 TCK）的 cJTAG 模式。
2. 完成第一个增量构建步骤后，将压缩机电机导线连接至端子 J7、J8 和 J9。
3. 完成第一个增量构建步骤后，将风扇电机导线连接至端子 J12。
4. 按照节 3.3 中的分步过程说明，通过将电源连接至 J3 和 J4，向逆变器施加直流总线电源、交流电源或交流市电。
   a. 直流电源的最大输出为 380V VDC。
   b. 交流电源的最大输出为 265V VAC（频率为 50/60Hz）。
   c. 交流市电为 220V VAC（频率为 50/60Hz）。
5. 按照节 3.3 中的分步过程说明，将输出端子 J10 和 J11 连接至电子负载（保持正确的极性）或阻性负载。J11 是 VDC 输出，J10 是 GND 端子。
6. 连接万用表、示波器探头和其他测量设备，以根据需要探测或分析各种信号和参数。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。

备注

如果在测试时外部仿真器出现连接问题，请在 JTAG 信号和 USB 电缆上添加铁氧体磁珠。此外，使连接线路尽可能短。

WARNING

两个电源域的接地平面可以相同或不同，具体取决于硬件配置。在将任何测试设备与电路板连接之前，应满足适当的隔离要求，以确保您和您的设备的安全。在为电路板供电之前，请检查 GND 连接。如果测量设备连接至电路板，则需要隔离器。

3.2 固件入门

通过 TI 提供的链接下载 C2000WARE-MOTORCONTROL-SDK v4.01.00.00 或更高版本的软件，并在其默认文件夹中安装该 MotorControl SDK 软件。

C2000WARE-MOTORCONTROL-SDK 中提供了该参考设计的软件。软件工程将保留在 C2000Ware MotorControl SDK 文件夹 `<install_location>\solutions\tidm_02010_dmpfc\` 中。按照以下步骤使用不同的增量构建来构建和运行该代码。

3.2.1 下载并安装电路板测试所需的软件

1. 从 Code Composer Studio (CCS) 集成开发环境 (IDE) 工具文件夹下载 Code Composer Studio 并进行安装。建议使用版本 12.0 或更高版本。
2. 通过以下两种方式之一安装 C2000WARE-MOTORCONTROL-SDK：
   - 通过 C2000Ware MotorControl SDK 工具文件夹下载软件
   - 转至 CCS 的“View”→“Resource Explorer”下。在 TI Resource Explorer 下，转至“Software”→“C2000Ware_MotorControl_SDK”，然后点击安装按钮。
3. 安装完成后，关闭 CCS 并创建一个新的工作区以导入工程。

备注

该参考设计支持 SysConfig 配置器件引脚并在易于使用的图形界面中初始化器件外设。该功能仅供当前版本参考。用户可以下载 SysConfig，并参阅 C2000 SysConfig 软件指南以实施 SysConfig，从而将参考设计迁移至自己的板以配置器件。

3.2.2 打开 CCS 内的工程

F28002x、F28003x 和 F280013x 各有一个工程，基于 F28002x 的参考设计的 projectspec 文件位于 `<install_location>\solutions\tidm_02010_dmpfc\f28002x\ccs\sensorless_foc` 目录下，基于 F28003x 的参考设计的 projectspec 文件位于 `<install_location>\solutions\tidm_02010_dmpfc\f28003x\ccs\sensorless_foc` 目录下，基于 F280013x 的参考设计的 projectspec 文件位于 `<install_location>\solutions\tidm_02010_dmpfc\f280013x\ccs\sensorless_foc` 目录下

在 CCS 中导入工程并通过右键点击工程名称来选择合适的构建配置，如图 3-3 所示。为 HVAC 参考设计选择合适的构建配置。“HVAC_REV3P2_3SC_LIB”构建配置支持三分流器电流检测方法，“HVAC_REV3P2_1SC_LIB”支持单分流器电流检测方法。

使用以下两种方式之一配置工程以选择工程中的支持功能：

- 对于 F28002x，导航至 `<install_location>\solutions\tidm_02010_dmpfc\f28002x\ccs\sensorless_foc`；对于 F28003x，导航至 `<install_location>\solutions\tidm_02010_dmpfc\f28003x\ccs\sensorless_foc`；对于 F280013x，导航至 `<install_location>\solutions\tidm_02010_dmpfc\f280013x\ccs\sensorless_foc`。使用文

本编辑器实用程序打开工程规范文件，为系统选择合适的预定义符号，如图 3-4 所示。通过在名称中删除或添加“_N”，可以激活或禁用预定义符号。例如，将“MOTOR1_FWC_N”中的“_N”删除（使其变为“MOTOR1_FWC”）可启用弱磁控制，而将“MOTOR1_FWC”符号名称更改为“MOTOR1_FWC_N”可禁用电机 1（压缩机）的弱磁控制功能。在“tidm_02010_dmpfc_002x/003x/0013x.projectspec”中设置预定义符号之后，依次点击“Project”、“Import CCS Projects”，在相关文件夹中选择基于硬件板的工程，从而导入工程。

- 依次点击“Project”、“Import CCS Projects”，对于基于 F28002x 的参考设计，转至 `<install_location>\solutions\tidm_02010_dmpfc\f28002x\ccs\sensorless_foc` 以将工程导入 CCS，对于基于 F28003x 的参考设计，转至 `<install_location>\solutions\tidm_02010_dmpfc\f28003x\ccs\sensorless_foc`，而对于基于 F280013x 的参考设计，转至 `<install_location>\solutions\tidm_02010_dmpfc\f280013x\ccs\sensorless_foc`，然后右键点击导入的工程名称，点击“Properties”命令为工程设置预定义符号，如图 3-5 所示。

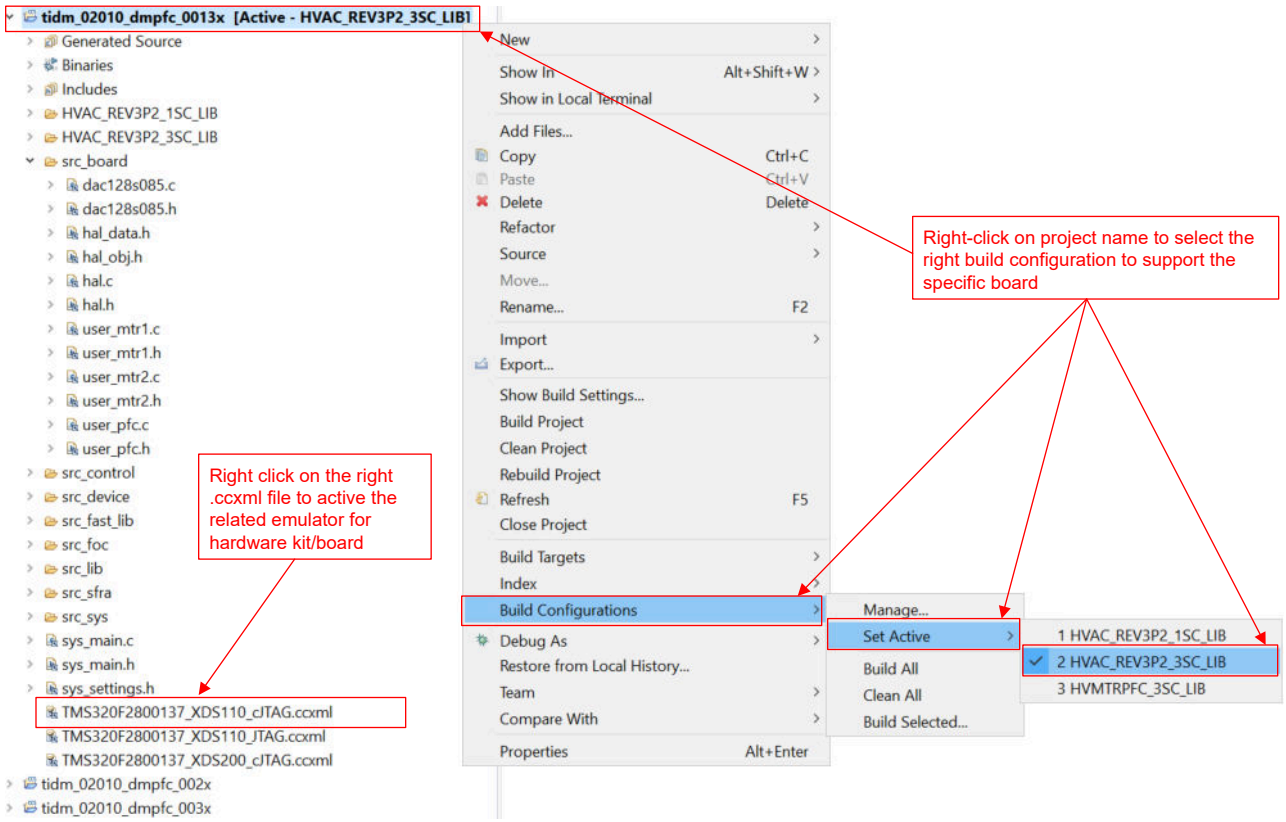


图 3-3. 选择合适的构建配置

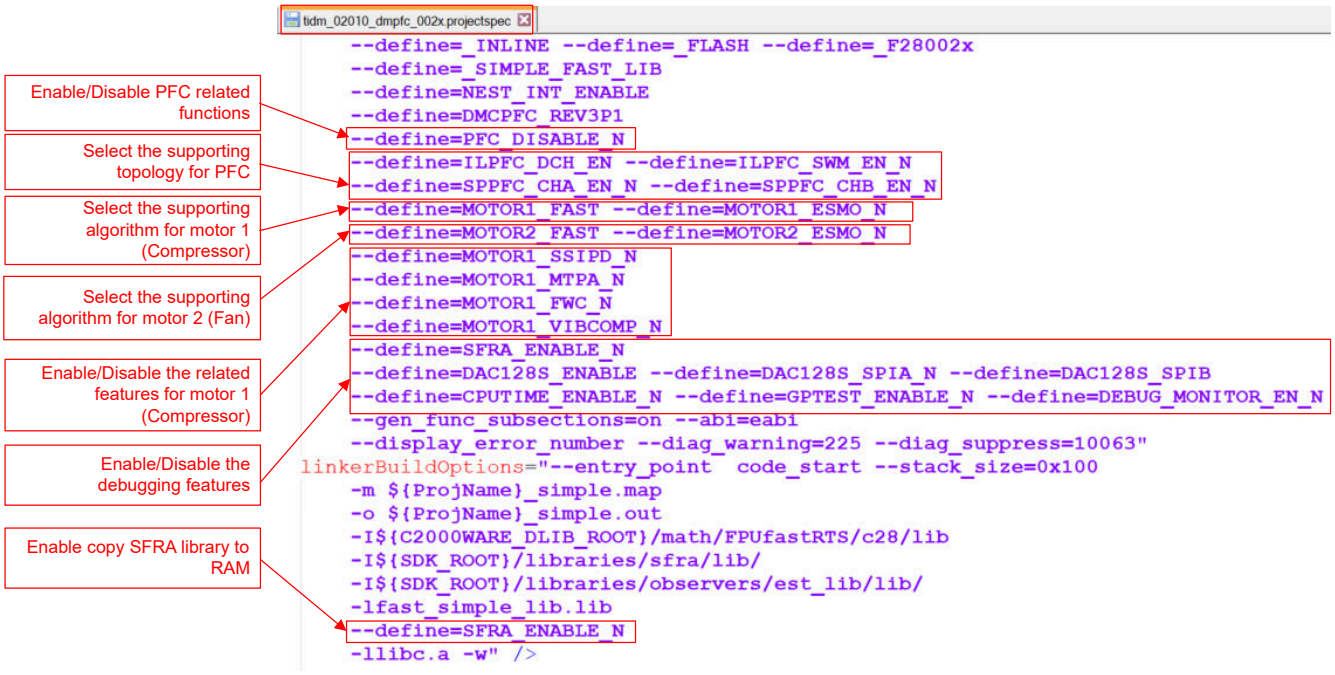


图 3-4. 在 ProjectSpec 文件中选择合适的预定义符号

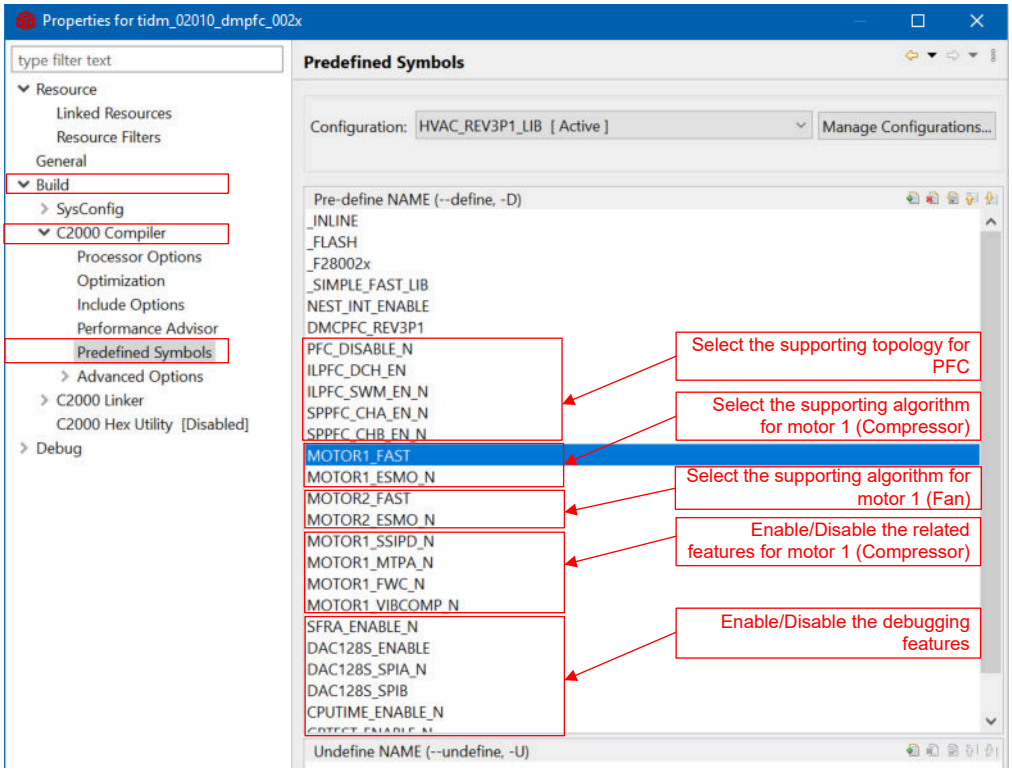


图 3-5. 在工程属性中选择合适的预定义符号

3.2.3 工程结构

图 3-6 中显示了工程的总体结构。器件外设配置基于 C2000Ware driverlib。如果用户需要将参考设计软件迁移到自己的电路板上，则只需更改 *hal.c* 和 *hal.h* 中的代码和定义以及 *user_mtr1/mtr2/pfc.h* 中的参数。

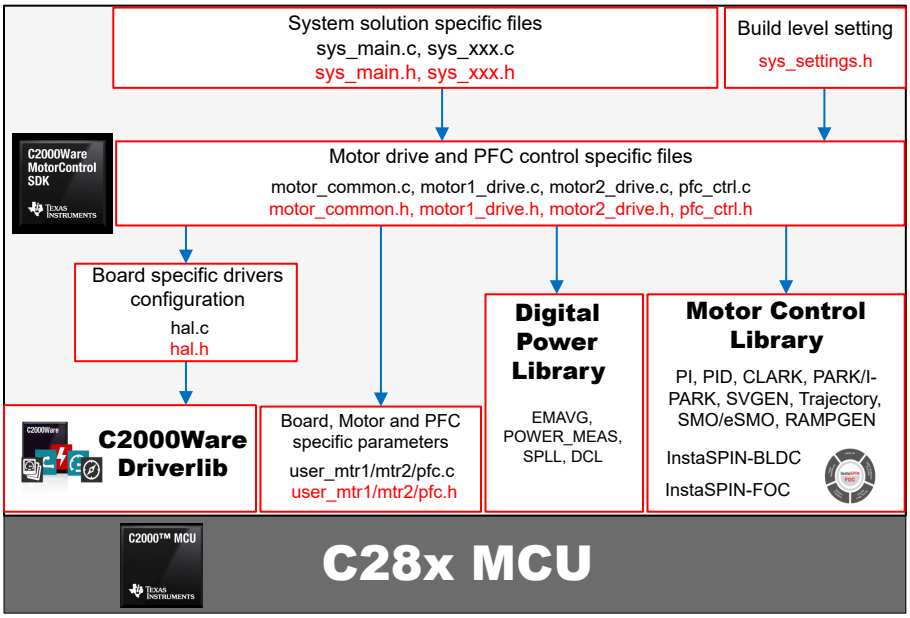


图 3-6. 工程结构概览

导入工程后，CCS 中将显示工程浏览器，如图 3-7 所示。

文件夹 *src_foc* 包含典型的 FOC 模块，其中包括变换、PID 和由电机驱动算法组成的估算器，独立于特定器件和电路板。

文件夹 *src_lib* 包含 PFC 的数字电源控制模块、InstaSPIN-FOC 库和独立于器件和电路板的数学库。

文件夹 *src_sys* 包含为系统控制保留的一些文件，这些文件独立于特定的器件和电路板。

文件夹 *src_control* 包含电机驱动和 PFC 控制文件，这些文件在中断服务例程和后台任务中调用电机和 PFC 控制核心算法函数。

F28002x、F28003x 和 F280013x 工程共享所有控制算法文件。特定于电路板、特定于电机和特定于器件的文件位于 *src_board* 文件夹中，这些文件包含用于运行解决方案、特定于器件的驱动程序。F28002x、F28003x 和 F280013x 分别具有专用的 *hal.c*、*hal.h*、*user_mtr1.h*、*user_mtr2.h* 和 *user_pfc.h* 因此，如果用户希望将工程迁移到不同的电路板或器件，则只需根据板的器件外设使用情况对这些头文件 *hal.c*、*hal.h*、*user_mtr1.h*、*user_mtr2.h* 和 *user_pfc.h* 进行一些更改。

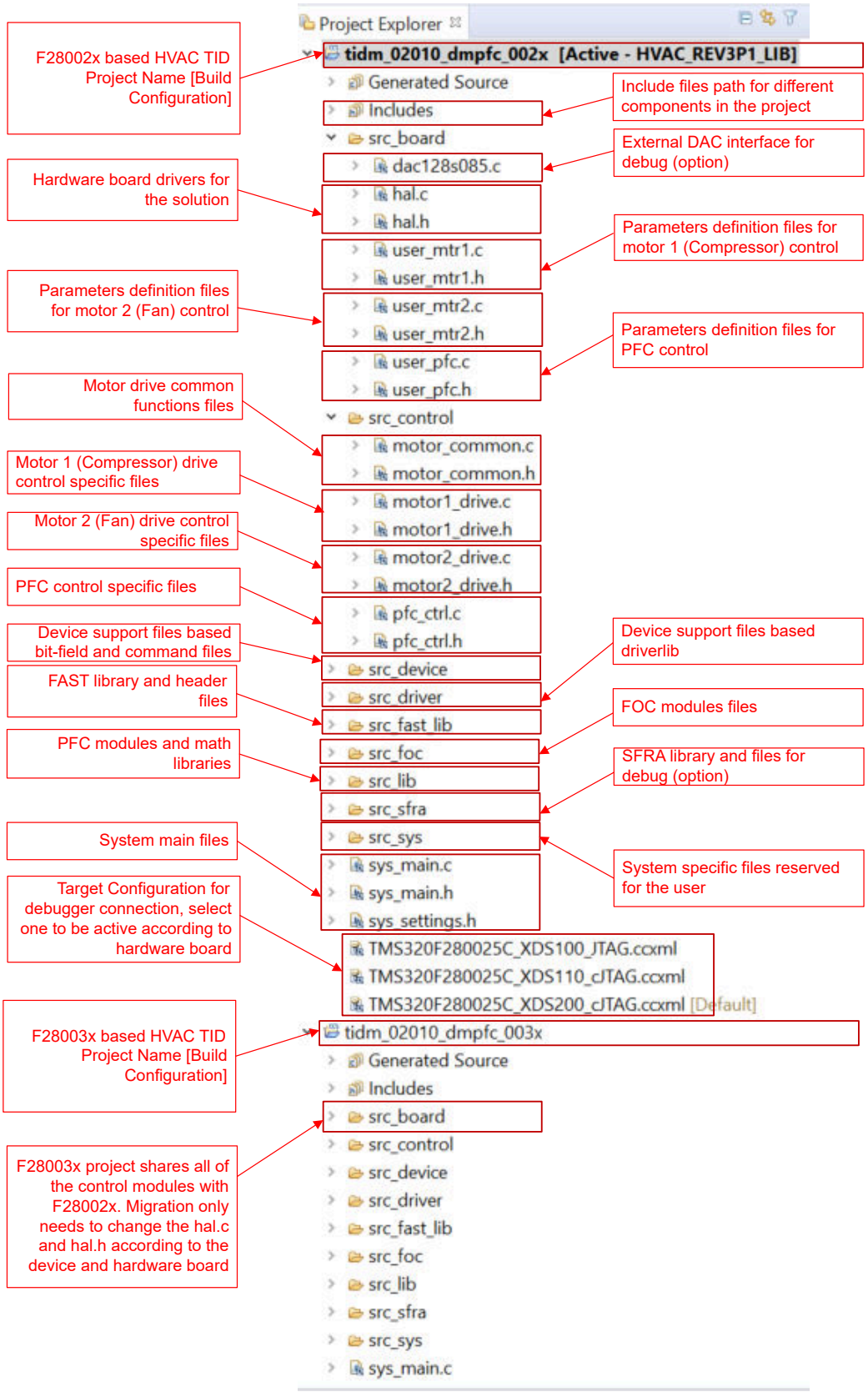


图 3-7. 双电机控制和 PFC 工程的工程浏览器视图

图 3-8 显示了固件的工程软件流程图，其中包括三个用于 PFC 和电机控制的 ISR、一个用于 PFC 的主循环和后台循环中的电机控制参数更新。

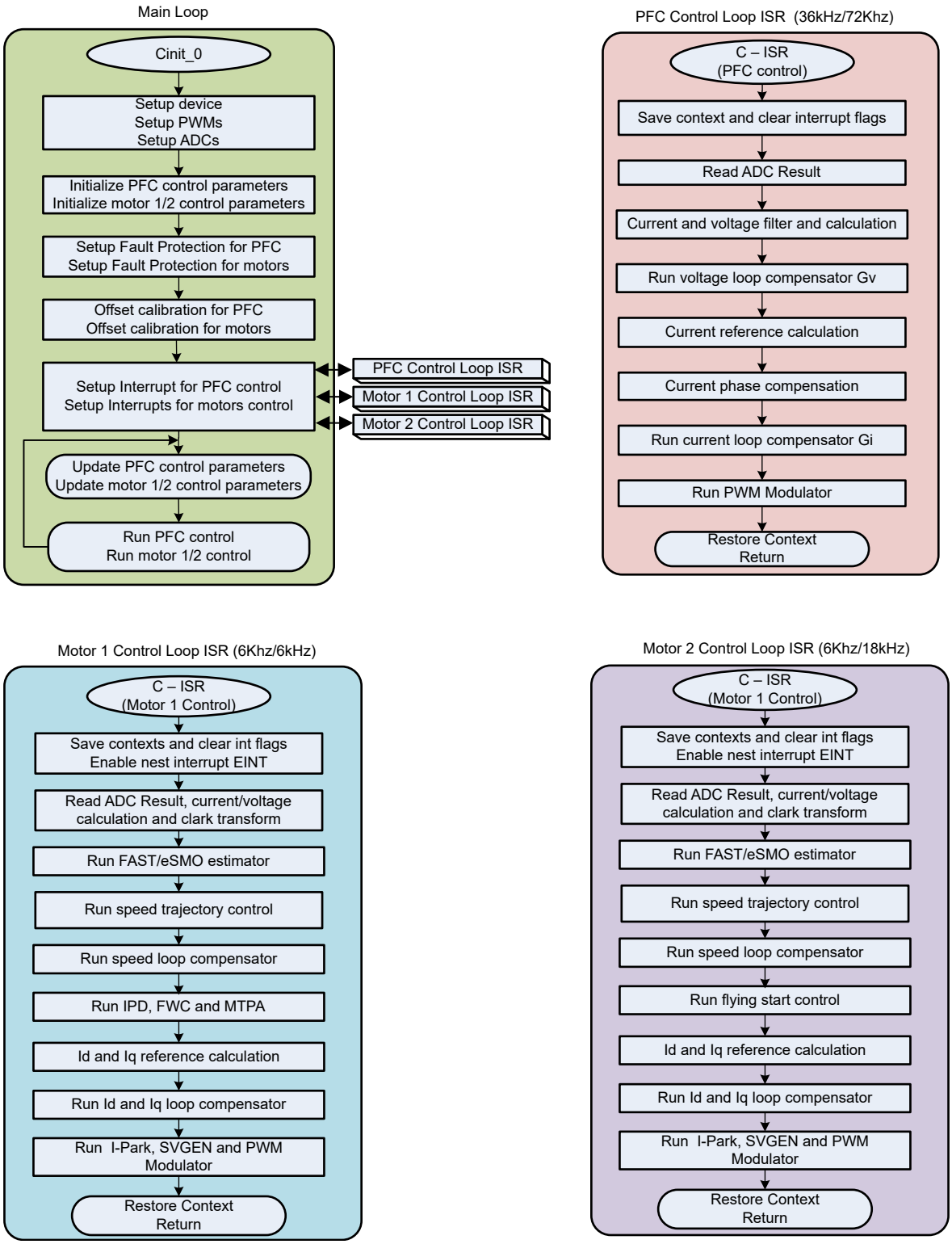


图 3-8. 固件的工程流程图

工程包含三个中断服务例程，每个 PWM 周期或每两个 PWM 周期调用一次。一些后台任务以轮询方式进行调用，可用于运行不要求绝对计时精度的缓慢任务（如 PFC 控制参数更新、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（压缩机），该电机以 USER_M1_ISR_FREQ_Hz 的频率定期触发。

motor2CtrlISR 被保留，用于调用电机驱动控制算法以旋转电机 2（风扇），该电机以 USER_M2_ISR_FREQ_Hz 的频率定期触发。

pfcCtrlISR 被保留，用于调用数字电源控制算法以实现以 USER_PFC_ISR_FREQ_Hz 的频率定期触发的 PFC。

为了简化系统开发和设计，该参考设计的软件在四个实验室中通过增量构建（PFC_BUILDLEVEL 和 DMC_BUILDLEVEL）进行了组织，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 显示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 3.3 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
PFC	PFC_LEVEL_1	50% PWM 占空比输出，验证 ADC 失调电压校准、PWM 输出和相移
	PFC_LEVEL_2	开环控制，用于检查 PFC 的电流和电压检测信号
	PFC_LEVEL_3	闭合电流环路
	PFC_LEVEL_4	闭合电流和电压环路
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	开环控制，用于检查电机 1 和 2 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/eSMO 运行

3.3 测试步骤

CAUTION

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。

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。

3.3.1 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 1、电机 2、PFC 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 3-9 显示了该构建级别的软件流程。

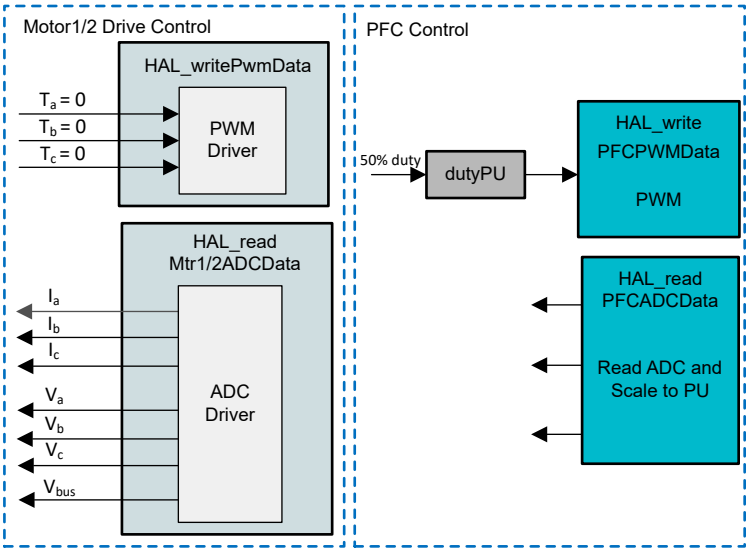


图 3-9. 控制软件方框图：构建级别 1 - 偏移验证

3.3.1.1 启动 CCS 并打开工程

1. 将 +5V、+15V 和 +12V 电源连接至相关测试点，TP7 用于连接 +5V，TP8 用于连接 +15V，TP6 用于连接 +12V，TP5、TP9、TP11 用于连接电路板 GND，为栅极驱动器和控制器提供电源，如图 3-10 所示。
2. 打开 CCSv11.0（或更高版本）。工程包含开发可执行输出文件 (.out)（可在基于 C2000 控制器的硬件上运行）所需的所有文件和构建选项。在菜单栏中，点击 *Project → Import CCS Projects*。在 *Select search-directory:* 下，浏览至 C2000Ware MotorControl SDK 文件夹并选择 `<install_location>\solutions\tidm_02010_dmpfc`，点击 *Finish* 将相关的工程导入 CCS。此工程调用所有需要的工具（编译器、汇编器、链接器）来构建工程。
3. 在左侧的工程窗口中，点击工程左侧的加号符号 (+)。图 3-7 展示了一个示例工程窗口。

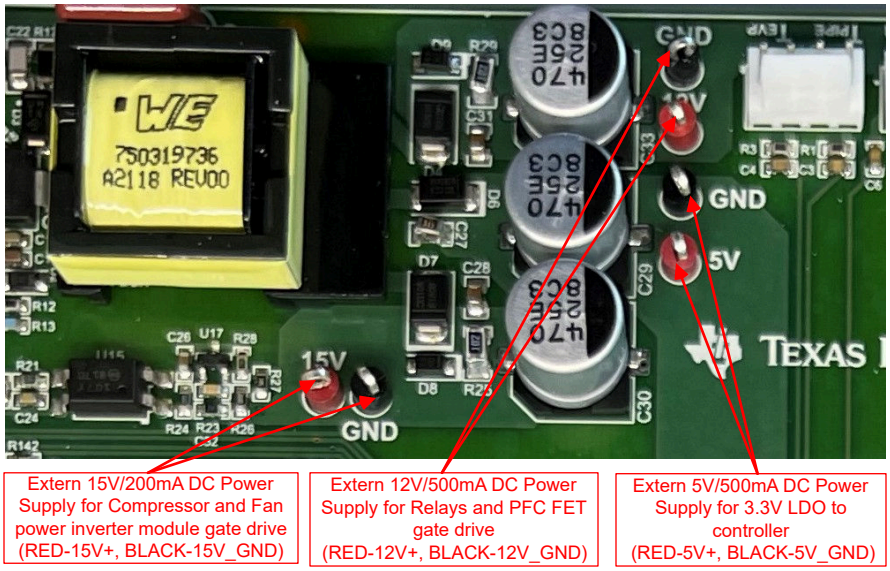


图 3-10. 连接外部直流电源以验证硬件

3.3.1.2 构建和加载工程

1. 打开 sys_settings.h 文件，有两个增量构建选项（PFC_BUILDLEVEL、DMC_BUILDLEVEL），将 PFC_BUILDLEVEL 设置为 PFC_LEVEL_1，将 DMC_BUILDLEVEL 设置为 DMC_LEVEL_1。
2. 如果之前构建了另一个构建选项，则右键点击工程名称并点击 *Clean Project*，然后点击 *Build Project*。观察构建窗口中运行的工具。项目成功构建。

3. 在 Project Explorer 中，确保将正确的目标配置文件设置为“Active”，如图 3-7 所示。
4. 打开工作台直流电源，从而为控制器和栅极驱动器的 LDO 提供 +5V、+12V 和 +15V。点击“Debug”按钮  或点击 *Run → Debug*。此时构建级别 1 代码应该已编译并加载到 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。程序应该在 main() 的开始处停止。

3.3.1.3 设置调试环境窗口

在调试代码时观察局部和全局变量是标准调试做法。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。

1. 点击菜单栏上的 *View → Expressions* 打开一个“Expressions”监视窗口。将鼠标移至“Expressions”窗口以查看工程中使用的变量。将变量添加到“Expressions”窗口，如图 3-11 所示，它使用声明期间与变量关联的数字格式，显示了“Expressions”窗口的一个示例。您可以通过右键点击变量并进行选择来为变量选择所需的数字格式。
2. 或者，可以通过右键点击“Expressions”窗口并点击“Import”将一组变量导入到“Expressions”窗口中，然后浏览至工程目录 `<install_location>\solutions\tidm_02010_hvac_dmpfc\common\debug`，选择 *BuildLevel1.txt* 并点击“OK”以导入图 3-11 中所示的变量。

   请注意，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. 结构体变量 *motorVars[0]*、*motorVars[1]* 和 *pfcVars* 引用了与使用 PFC 控制双电机相关的大多数变量。通过展开该变量，您可以查看所有内容并根据需要进行编辑。
4. 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。请注意，选择过短的刷新间隔可能会影响性能。

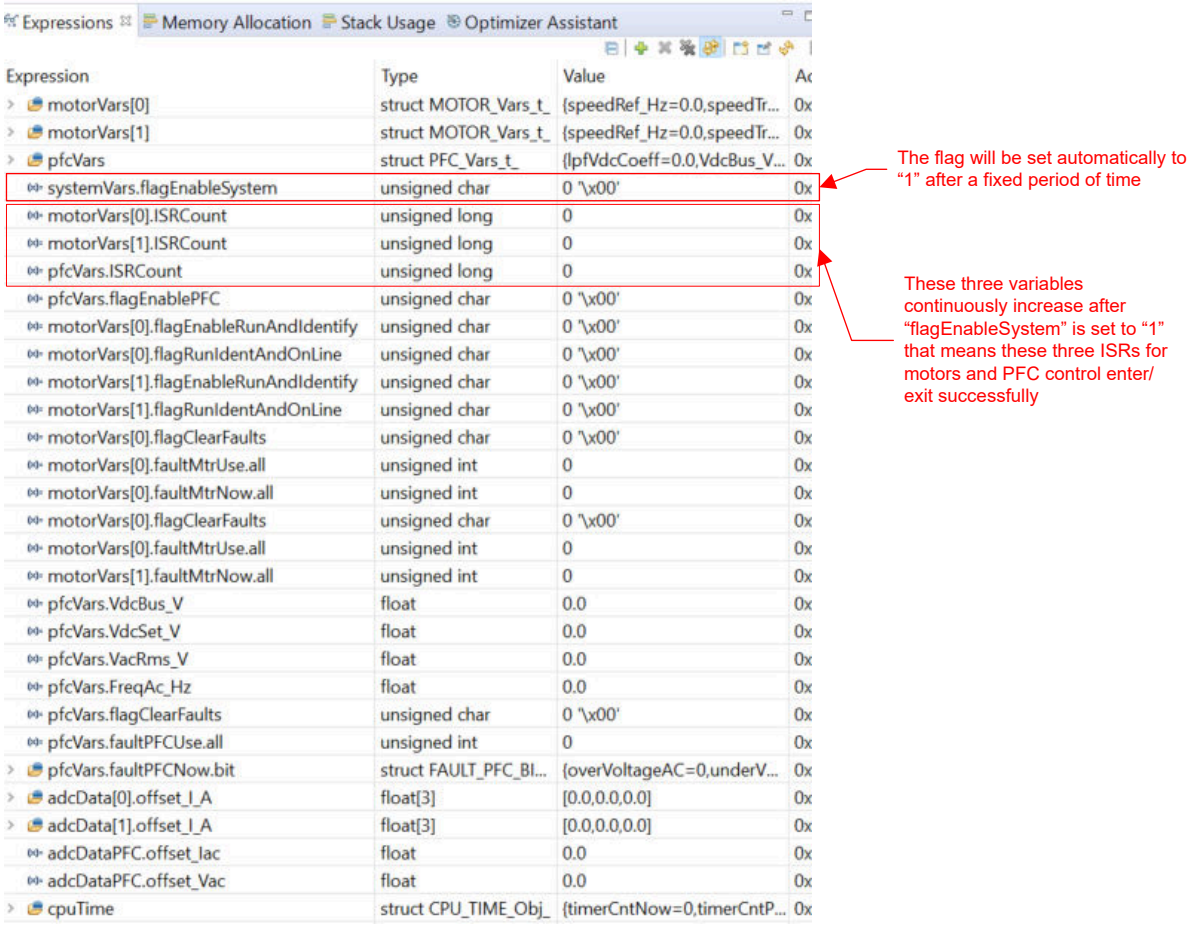


图 3-11. 构建级别 1：重置时的“Expressions”监视窗口

3.3.1.4 运行代码

1. 通过点击按钮 来运行工程，或点击“Debug”选项卡中的 *Run → Resume*。
2. 在“Expressions”窗口中，将变量 *pfcVars.flagEnablePFC*、*motorVars[0].flagEnableRunAndIdentify* 和 *motorVars[1].flagEnableRunAndIdentify* 设置为 1，然后在该监视窗口中 *systemVars.flagEnableSystem* 自动设置为 1。
3. 工程现在应该已运行，在使用该工程时图和表达式窗口中的值应不断更新，如图 3-12 所示。您可能需要根据您自己的偏好来调整窗口大小。
4. 在该监视视图中，变量 *motorVars[0].flagRunIdentAndOnLine* 和 motorVars[1].flagRunIdentAndOnLine 应自动设置为 1。*ISRCount* 应不断增加。
5. 检查双电机和 PFC 的校准偏移，电机相电流检测的偏移值应约等于 ADC 标度电流的一半，如图 3-12 所示。
6. 使用示波器探测双电机和 PFC 驱动控制的 PWM 输出。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 3-13 所示。电机 1 的 PWM 开关频率为 6kHz，电机 2 和 PFC 的 PWM 频率是电机 1 的整数倍，分别为 18kHz 和 72kHz。电机 1、电机 2 和 PFC 之间有固定的延迟，以避免触发 ADC 模块以及 ISR 同时占用 CPU。
7. 现在可以停止控制器，并终止调试连接。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. 通过点击 *Tools* →“On-Chip Flash”，然后点击“On-Chip Flash”选项卡中的 *Erase Flash* 来擦除控制器中的代码以实现下一个构建级别（确保选中了所有闪存组），如图 3-14 所示。该操作将擦除闪存中存储的所有程序代码。（该步骤是可选的，用户可以忽略该步骤以在下一个构建级别加载新的程序代码）

> 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” 或点击 *Run* → *Terminate* 来关闭 CCS 调试会话。

图 3-12. 构建级别 1：运行时的“Expressions”窗口

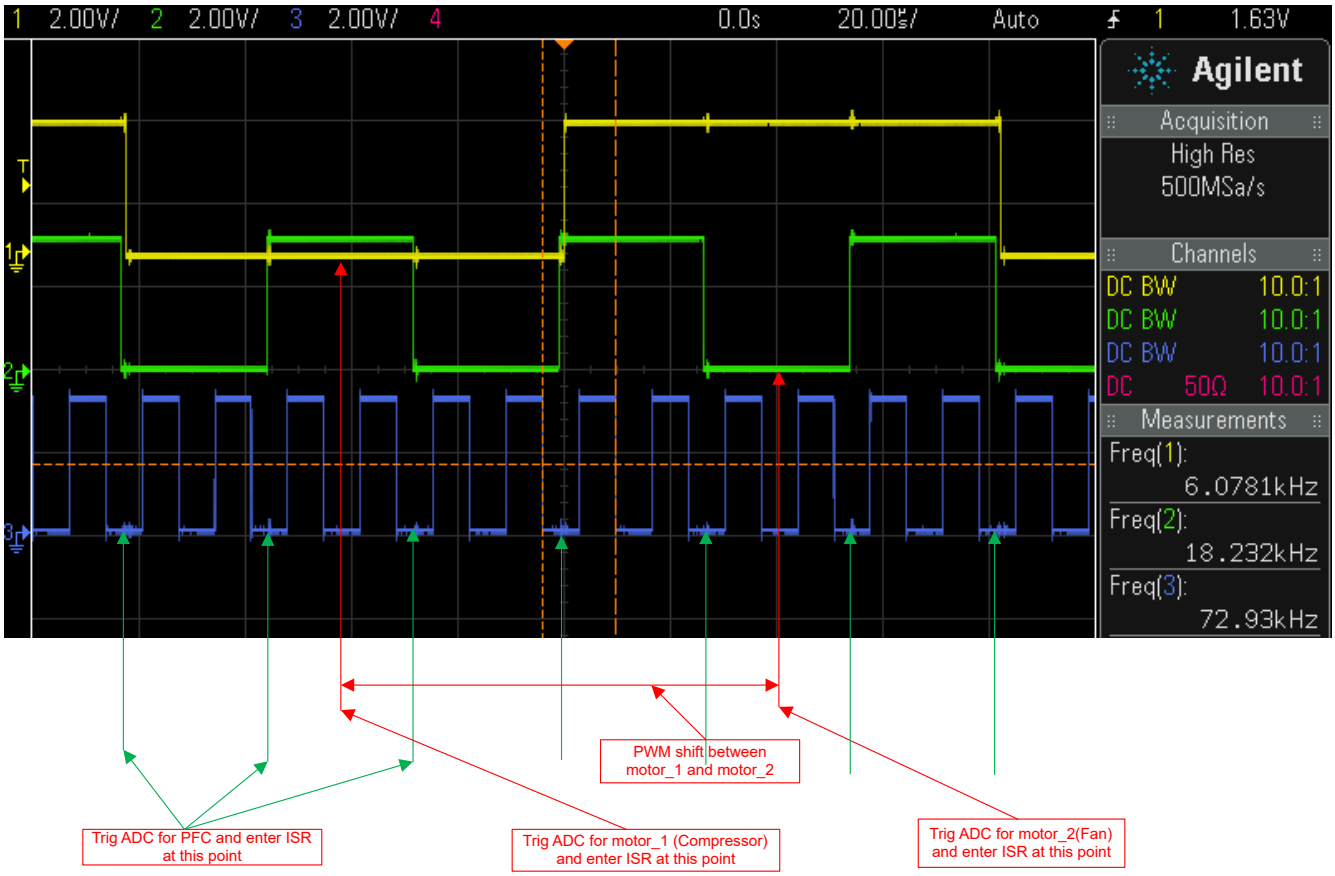


图 3-13. 构建级别 1：双电机和 PFC PWM 输出

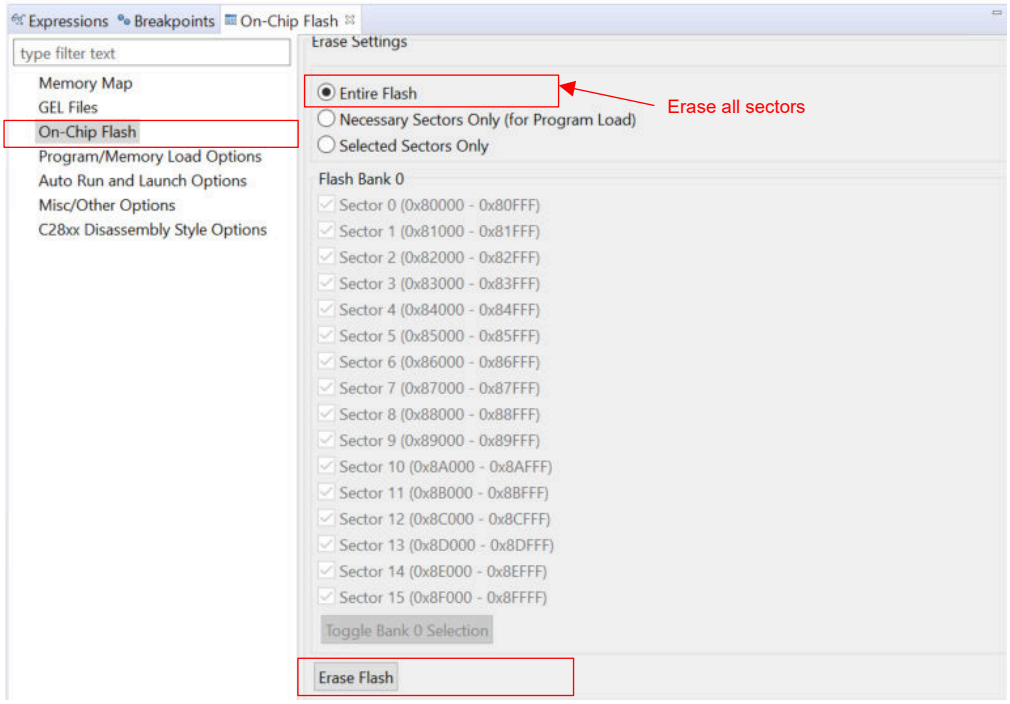


图 3-14. 构建级别 1：擦除闪存中的程序代码以实现下一个构建级别

3.3.2 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 控制以驱动双电机，从而验证电流和电压检测电路以及栅极驱动器电路。
- 实现 PFC 的开环控制，以验证交流电压、直流电压和交流电流检测电路。
- 测试用于电机控制的 InstaSPIN-FOC FAST 或 eSMO 模块。
- 测试用于 PFC 控制的直流电压和电流测量模块。

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 控制器和栅极驱动器的偏置电源。图 3-15 显示了该构建级别的软件流程。

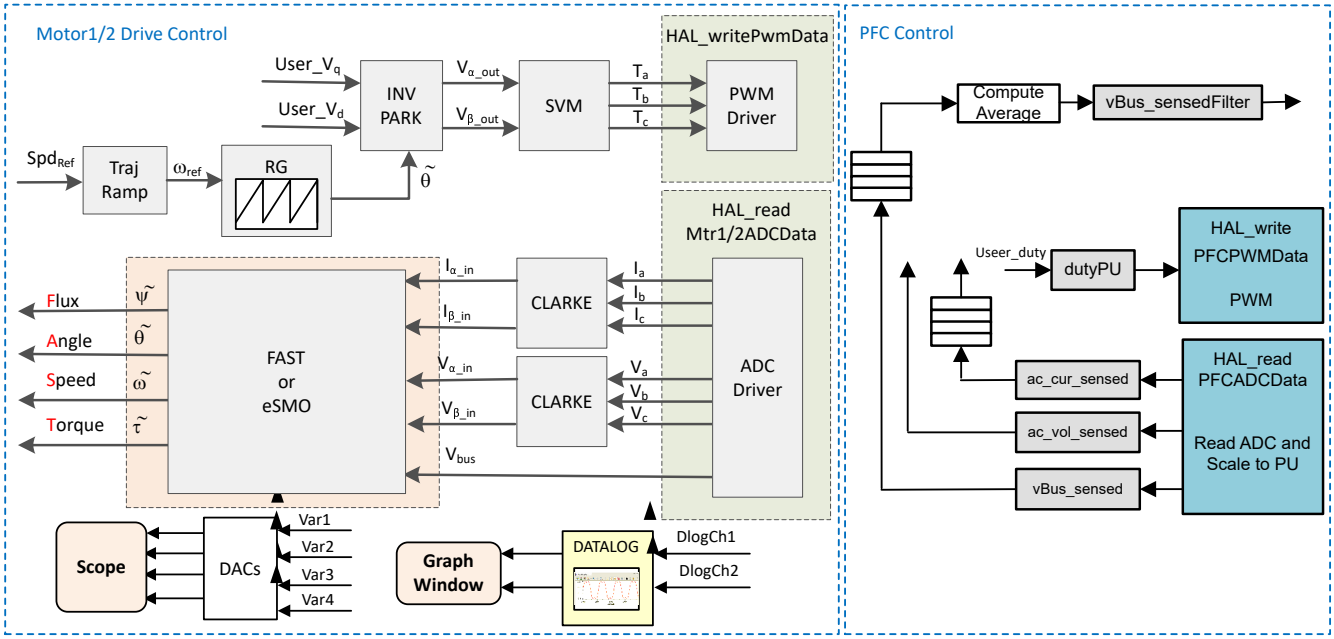


图 3-15. 控制软件方框图：构建级别 2 – 开环控制

3.3.2.1 启动 CCS 并打开工程

1. 将一个能够提供高达 1.6kW 通用输入的可编程隔离式交流电源连接至参考板的输入端子（连接器 J3 和 J4），如图 3-2 所示。将电源电流限制设置为 2A。此时不要打开电源。
2. 在输出端连接一个约 500Ω 的 400V 阻性负载（连接器 J11/Vdc+ 和 J10/Vdc-）。
3. 将电机（压缩机）连接至 J7/J8/J9，将电机（风扇）连接至 J12。
4. 按照节 3.3.1.1 中的步骤 2 和 3 打开工程。

3.3.2.2 构建和加载工程

1. 打开 sys_settings.h 文件，有两个增量构建选项（PFC_BUILDLEVEL、DMC_BUILDLEVEL），将 PFC_BUILDLEVEL 设置为 PFC_LEVEL_2，将 DMC_BUILDLEVEL 设置为 DMC_LEVEL_2。
2. 按照节 3.3.1.2 中的步骤 2 至 4 构建工程并将代码加载到控制器中。

3.3.2.3 设置调试环境窗口

1. 通过选择 *BuildLevel2.txt*，按照节 3.3.1.3 中的步骤 1 至 4 将变量导入“Expressions”窗口。此时将显示“Expressions”窗口，如图 3-16 所示。

3.3.2.4 运行代码

1. 将交流电源输出设置为 0V，打开交流电源，将输出电压从 0V 缓慢增加至 110V 交流。
2. 通过点击按钮  来运行工程，或点击“Debug”选项卡中的 *Run → Resume*。经过固定的时长后，*systemVars.flagEnableSystem* 应设置为 *1*，这意味着偏移校准已完成并且浪涌电源继电器已开启。双电机和 PFC 的故障标志（*motorVars[0].faultMtrUse.all*、*motorVars[1].faultMtrUse.all* 和 *pfcVars.faultPFCUse.all*）应等于 *0*，否则用户必须按照节 3.3.1 中的说明检查电流和电压检测电路。

3. 要验证电机 1 逆变器的电流和电压检测电路，请在“Expressions”窗口中将变量 *motorVars[0].flagEnableRunAndIdentify* 设置为 *1*，如图 3-16 所示。电机 1 应以 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
```

4. 这应该使电机 1 现在以变量 *motorVars[0].speedRef_Hz* 中的设置转速旋转，检查“Expressions”窗口中 *motorVars[0].speed_Hz* 的值，*motorVars[0].speedRef_Hz* 和 *motorVars[0].speed_Hz* 的值应该非常接近，如图 3-16 所示。
5. 将示波器电压和电流探头连接到 PWMDAC 或 DAC128S 板的输出端（电机相电流）以探测角度、电流信号，示波器中的电流和角度波形如图 3-17 所示。请注意，力角发生器的角度与 FAST 或 eSMO 估算器的估算转子角度非常接近，这两个角度之间可能存在一点位移误差。使用 DAC 输出到示波器上的采样电流波形应该与电流探头捕获的相电流波形相同，这意味着电流检测电路非常适合电机控制。
6. 通过减小变量 *motorVars[0].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[0].overCurrent_A* 被设置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[0].flagEnableRunAndIdentify* 将被清除为 *0*，*motorVars[0].faultMtrUse.all* 将被设置为 *0x10*。
7. 按照步骤 3、4 和 5，使用相同的方法通过将变量 *motorVars[1].flagEnableRunAndIdentify* 设置为 *1* 并调整 *user_mtr1.h* 中的 v/f 参数使电机平稳旋转，来测试电机 2 的硬件。
8. 要验证 PFC 的电流和直流链路电压检测电路，请将探头连接至 PWMDAC 或 DAC128S 的输出以对电流和电压进行采样，使用高电压探头和电流探头检测直流链路电压和电流，图 3-18 显示了电压和电流波形。
9. 检查“Expressions”窗口中的变量 *pfcVars.VdcBus_V*，这些变量的值应与交流电源的设置值相同或使用万用表进行测量。非常缓慢地将 pfcVars.dutyOut 从 0.0 增加至 0.2，直流链路电压应该同时增加。
10. 通过减小 *pfcVars.overCurrent_A* 来验证过流保护，如果触发过流故障，PFC 的 PWM 输出将被禁用。
11. 现在可以停止控制器，并终止调试连接。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。
12. 通过点击“Terminate Debug Session”  或点击 *Run* → *Terminate* 来关闭 CCS 调试会话。

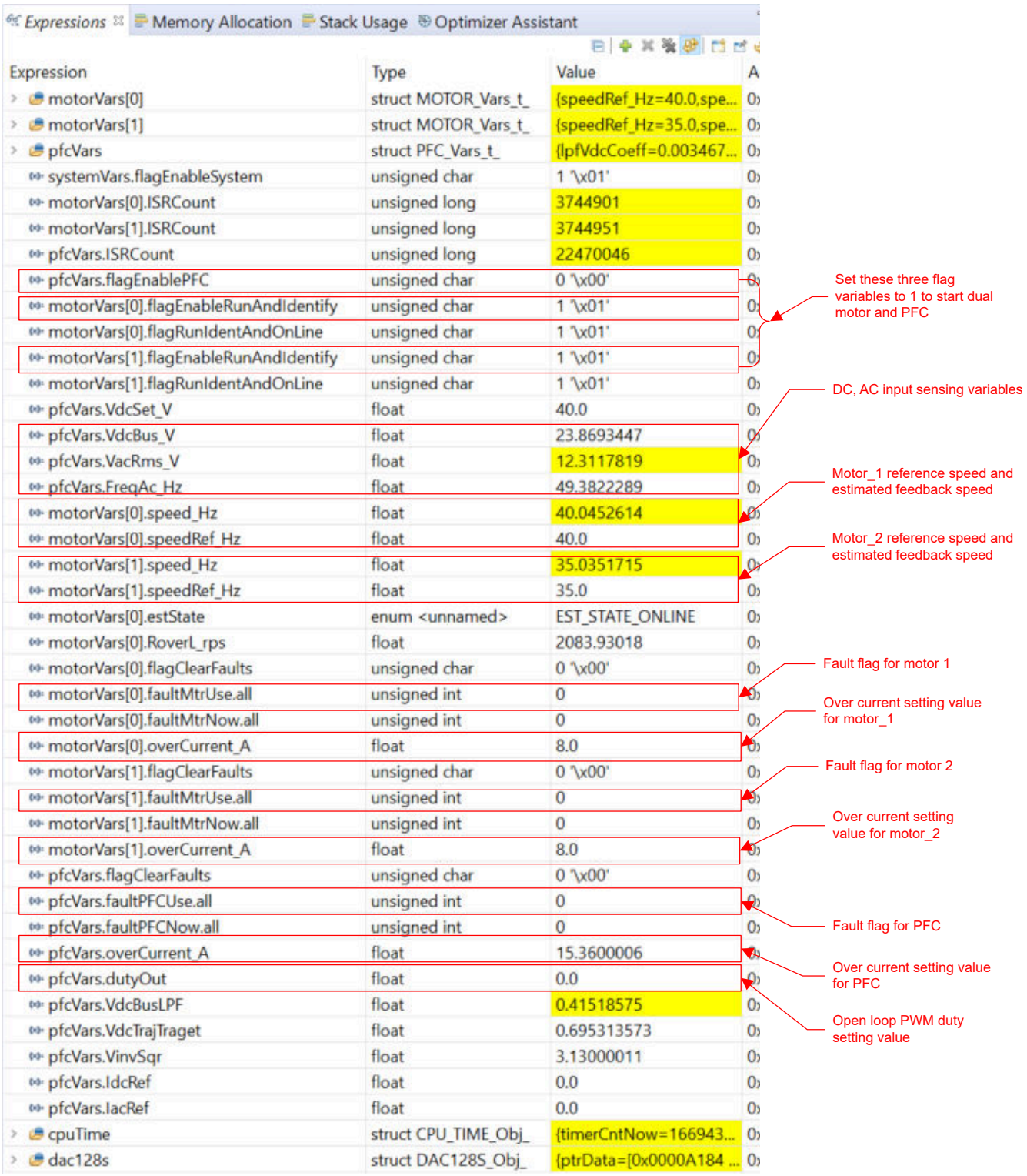


图 3-16. 构建级别 2：运行时的“Expressions”窗口

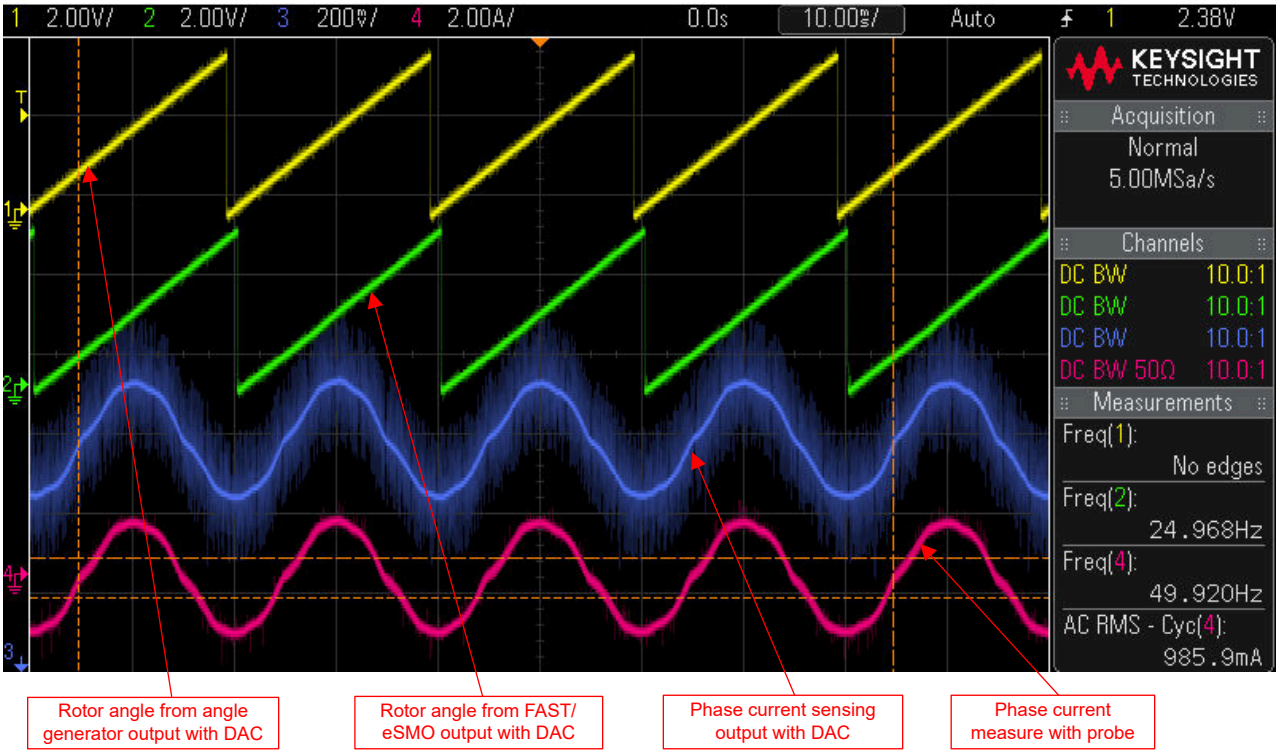


图 3-17. 构建级别 2：电机的转子角度、相电流

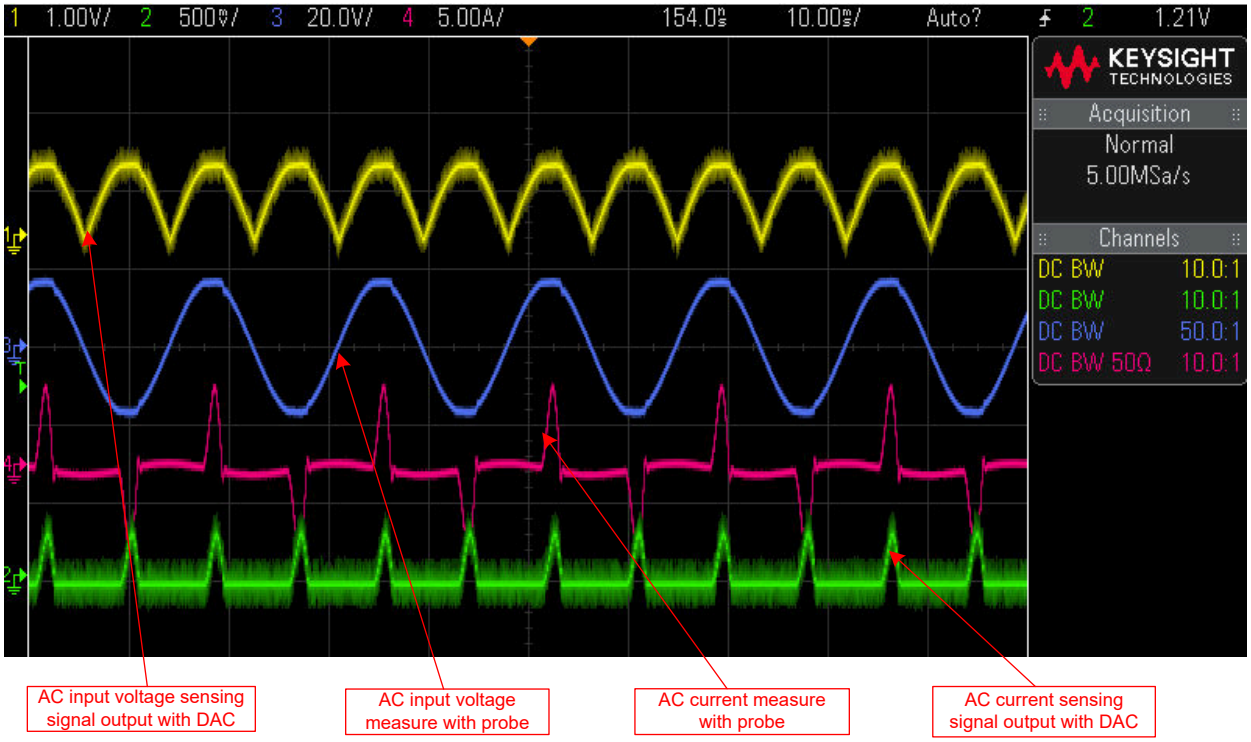


图 3-18. 构建级别 2：PFC 的直流电压、电流、PWM 输出

3.3.3 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 PFC 的闭合电流环路运行。
- 评估双电机的闭合电流环路运行。

在该构建级别中，PFC 的内部电流环路是闭合的，它使用电流补偿器控制电感器电流，通过 i/f 控制来控制电机，转子角度由斜坡发生器模块生成。图 3-19 显示了该构建级别的软件流程。

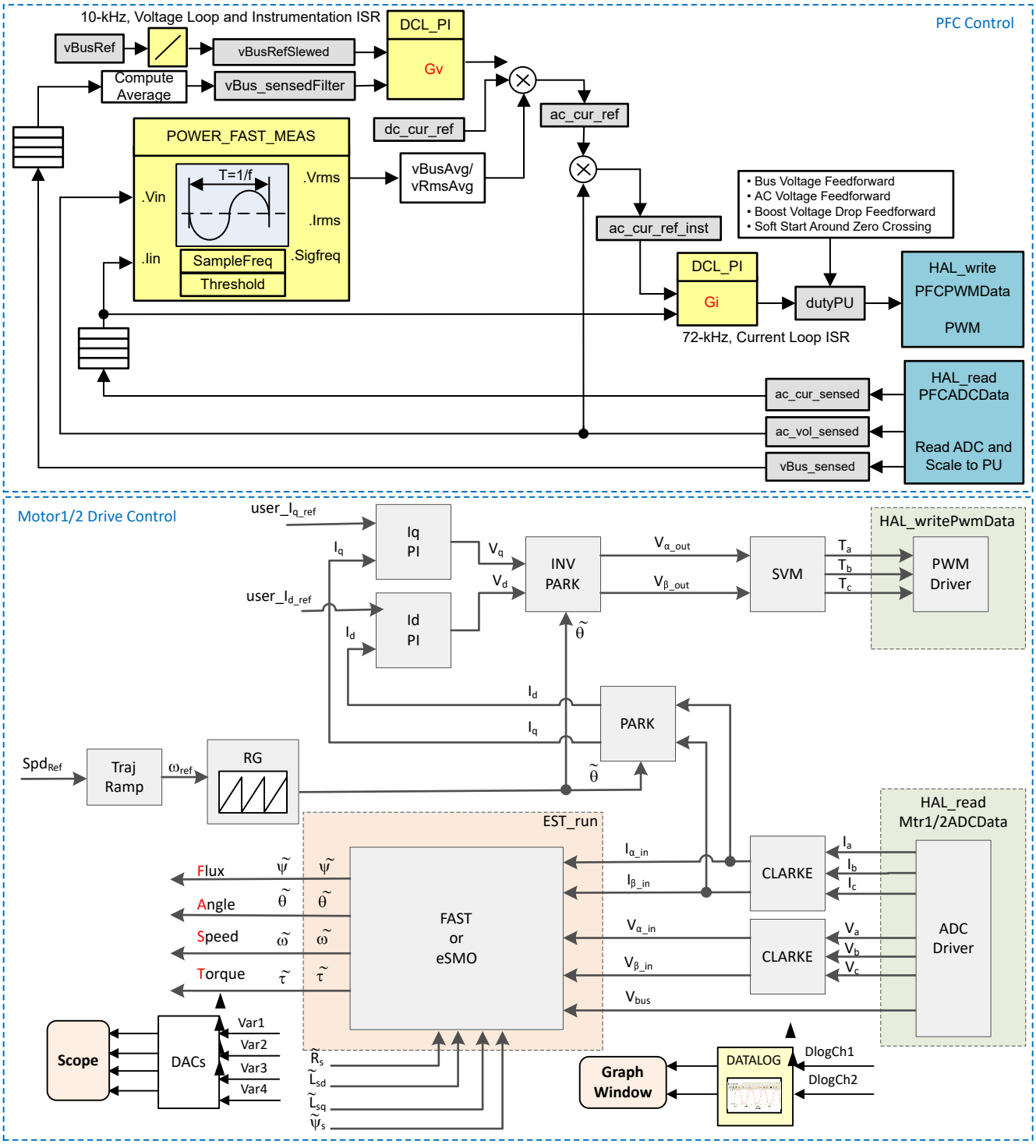


图 3-19. 控制软件方框图：构建级别 3 – 电流闭环控制

3.3.3.1 启动 CCS 并打开工程

1. 将一个能够提供高达 1.6kW 通用交流输入的可编程隔离式交流电源连接至参考板的输入端子（连接器 J3 和 J4），如图 3-2 所示。将电源电流限制设置为 3A，将输出频率设置为 50/60Hz。此时不要打开电源。
2. 在输出端连接一个约 500Ω 的 400V 阻性负载（连接器 J11/Vdc+ 和 J10/Vdc-）。
3. 将电机（压缩机）连接至 J7/J8/J9，将电机（风扇）连接至 J12。

4. 按照节 3.3.1.1 中的步骤 2 和 3 打开工程。

3.3.3.2 构建和加载工程

1. 打开 sys_settings.h 文件，有两个增量构建选项（PFC_BUILDLEVEL、DMC_BUILDLEVEL），将 PFC_BUILDLEVEL 设置为 PFC_LEVEL_3，将 DMC_BUILDLEVEL 设置为 DMC_LEVEL_3。
2. 按照节 3.3.1.2 中的步骤 2 至 4 构建工程并将代码加载到控制器中。

3.3.3.3 设置调试环境窗口

1. 通过选择 *BuildLevel3.txt*，按照节 3.3.1.3 中的步骤 1 至 4 将变量导入“Expressions”窗口。此时将显示“Expressions”窗口，如图 3-16 所示。

3.3.3.4 运行代码

1. 将交流电源输出设置为 0V (50/60Hz)，打开交流电源，将输入电压从 0V 缓慢增加至 110V 交流。
2. 通过点击按钮 来运行工程，或点击“Debug”选项卡中的 *Run* → *Resume*。经过固定的时长后，*systemVars.flagEnableSystem* 应设置为“*1*”，这意味着偏移校准已完成并且浪涌电源继电器已开启。双电机和 PFC 的故障标志（*motorVars[0].faultMtrUse.all*、*motorVars[1].faultMtrUse.all* 和 *pfcVars.faultPFCUse.all*）应等于“*0*”，否则用户必须按照节 3.3.1 中的说明检查电流和电压检测电路。
3. 要验证电机 1 的电流闭环控制，请在“Expressions”窗口中将变量 *motorVars[0].flagEnableRunAndIdentify* 设置为“*1*”，如图 3-20 所示。电机 1 在运行时应该使用来自角度发生器的角度以变量 *motorVars[0].speedRef_Hz* 中的设置速度进行闭环控制，检查“Expressions”窗口中 *motorVars[0].speed_Hz* 的值，两个变量的值应该非常接近。
4. 将示波器探头连接到 DAC 输出端和电机相线以探测角度、电流信号，示波器中的电流和角度波形如图 3-21 所示。更改“Expressions”窗口中的 *Idq_set_A[0].value[1]*，电机相电流应相应地增加。
5. 如果电机无法以电流闭环运行并出现过流故障，则检查是否根据硬件板正确设置了 *adcData[0].current_sf* 的符号和 *userParams[0].current_sf* 的值。
6. 按照步骤 3、4 和 5 使用相同的方法通过将变量 *motorVars[1].flagEnableRunAndIdentify* 设置为“*1*”并调整 *Idq_set_A[1].value[1]* 使电机旋转，来测试电机 2 的硬件。
7. 要通过将 *pfcVars.flagEnablePFC* 设置为“1”来验证 PFC 的电流闭环，请将探头连接至 DAC 的输出以对电流和电压进行采样，使用高电压探头和电流探头检测交流输入电压和电流，图 3-22 显示了电压和电流波形。
8. 检查“Expressions”窗口中的变量 *pfcVars.VdcBus_V*、*pfcVars.VacRms_V* 和 *pfcVars.FreqAc_Hz*，这些变量的值应与交流电源的设置值相同或使用万用表进行测量。
9. 非常缓慢地将 *pfcVars.IdcRef* 从 0.0 增大至 0.05，输出电压应相应地增加。以 0.01 为增量不断增加 *pfcVars.IdcRef*，直到 *pfcVars.IdcRef* 的值增加至 0.1。检测电流值 *pfcVars.IacSen* 可能会快速变化，并且与 *pfcVars.IdcRef* 值不同。之所以会发生这种变化，是因为 *pfcVars.IacSen* 是瞬时输入电流值，而 *pfcVars.IdcRef* 是电流命令的振幅基准。
10. 现在可以在将 *motorVars[0].flagEnableRunAndIdentify*、*motorVars[1].flagEnableRunAndIdentify* 和 *pfcVars.flagEnablePFC* 设置为“0”之前停止控制器并终止调试连接。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。
11. 通过点击“Terminate Debug Session” 或点击 *Run* → *Terminate* 来关闭 CCS 调试会话。

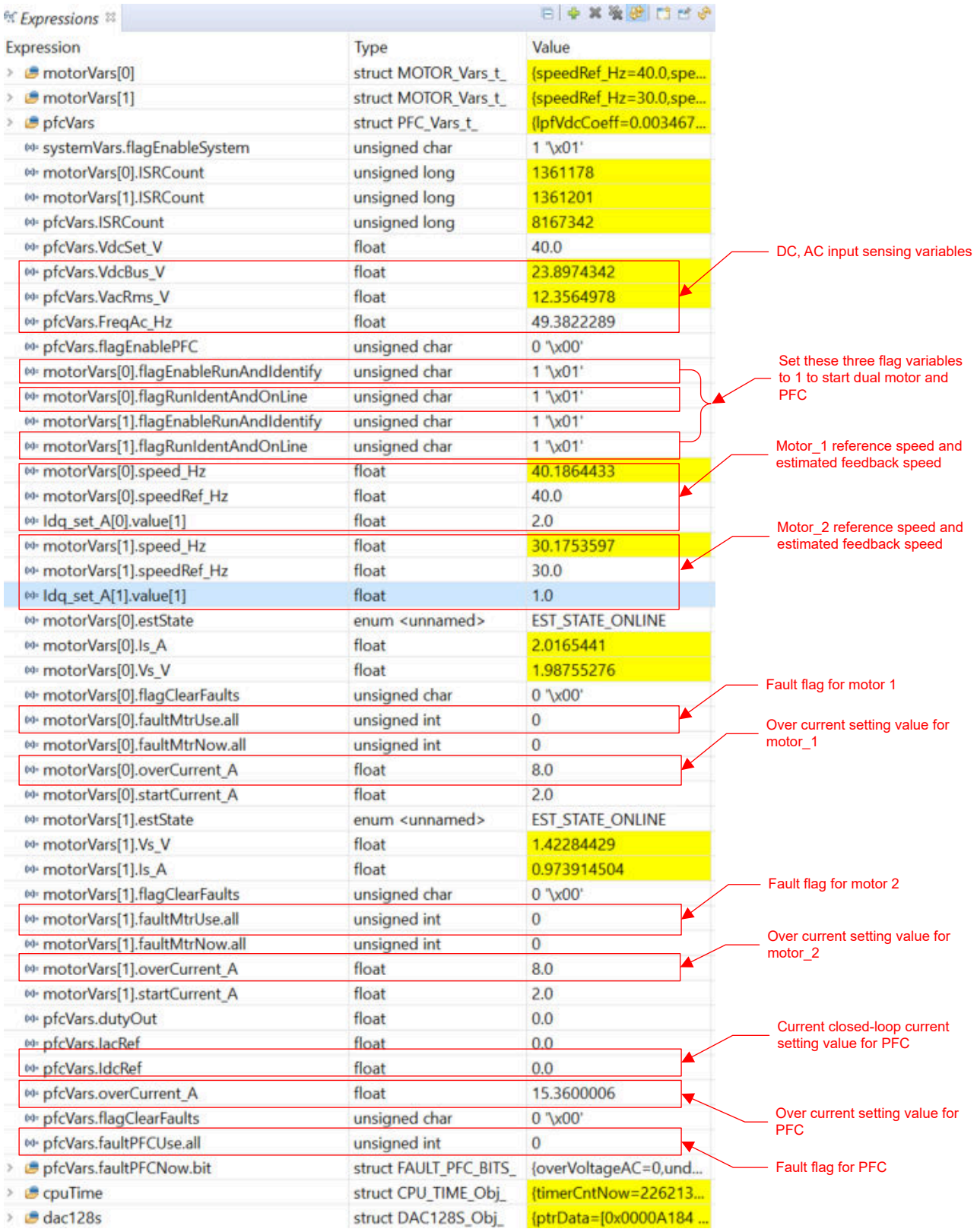


图 3-20. 构建级别 3：运行时的“Expressions”窗口

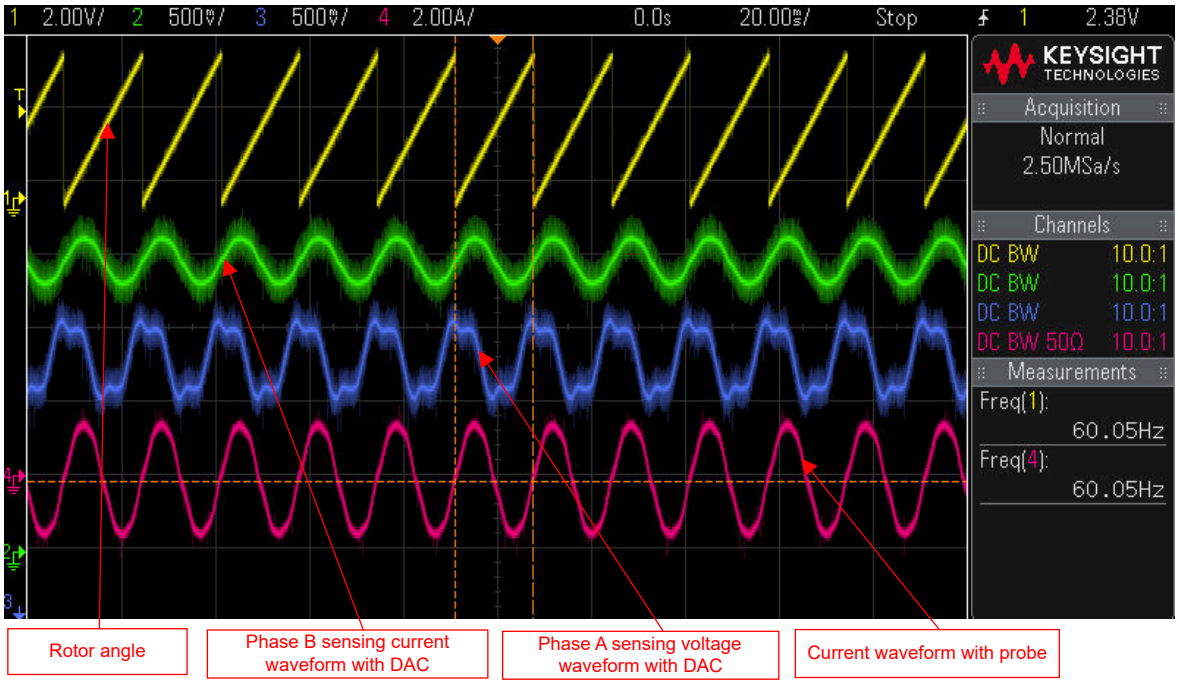


图 3-21. 构建级别 3：电机的转子角度、相电流

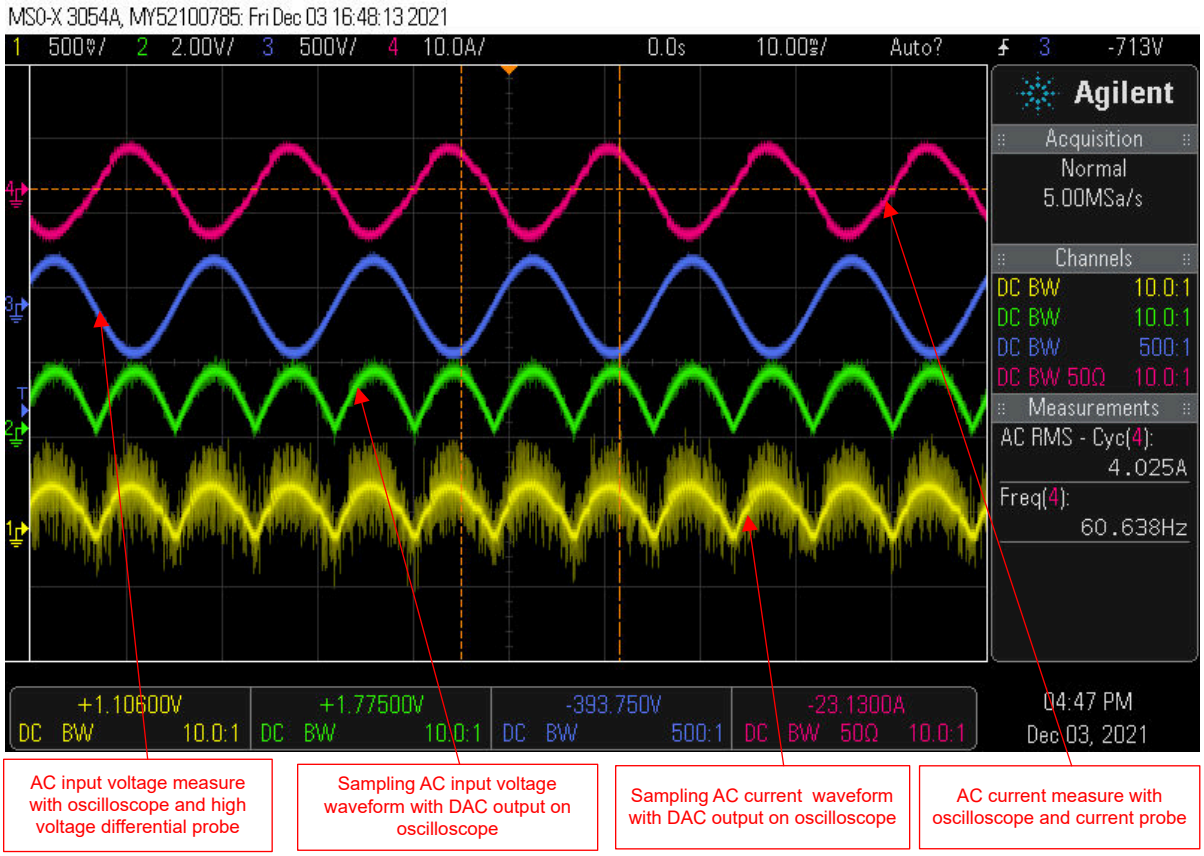


图 3-22. 构建级别 3：PFC 的交流电压、电流

3.3.4 版本级别 4：完全 PFC 和电机驱动控制

了解该构建级别中的目标：

- 评估完整的 PFC 控制。
- 评估压缩机和风扇的完整电机驱动。
- 评估附加功能：弱磁控制、压缩机扭矩补偿、风扇快速启动。
- 评估完成的系统。

在该构建级别中，外侧电压环路是闭合的，内侧电流环路用于 PFC。外侧速度环路是闭合的，内部电流环路用于电机 1 和电机 2，转子角度来自 FAST 或 eSMO 估算器模块。图 3-23 显示了该构建级别的软件流程。

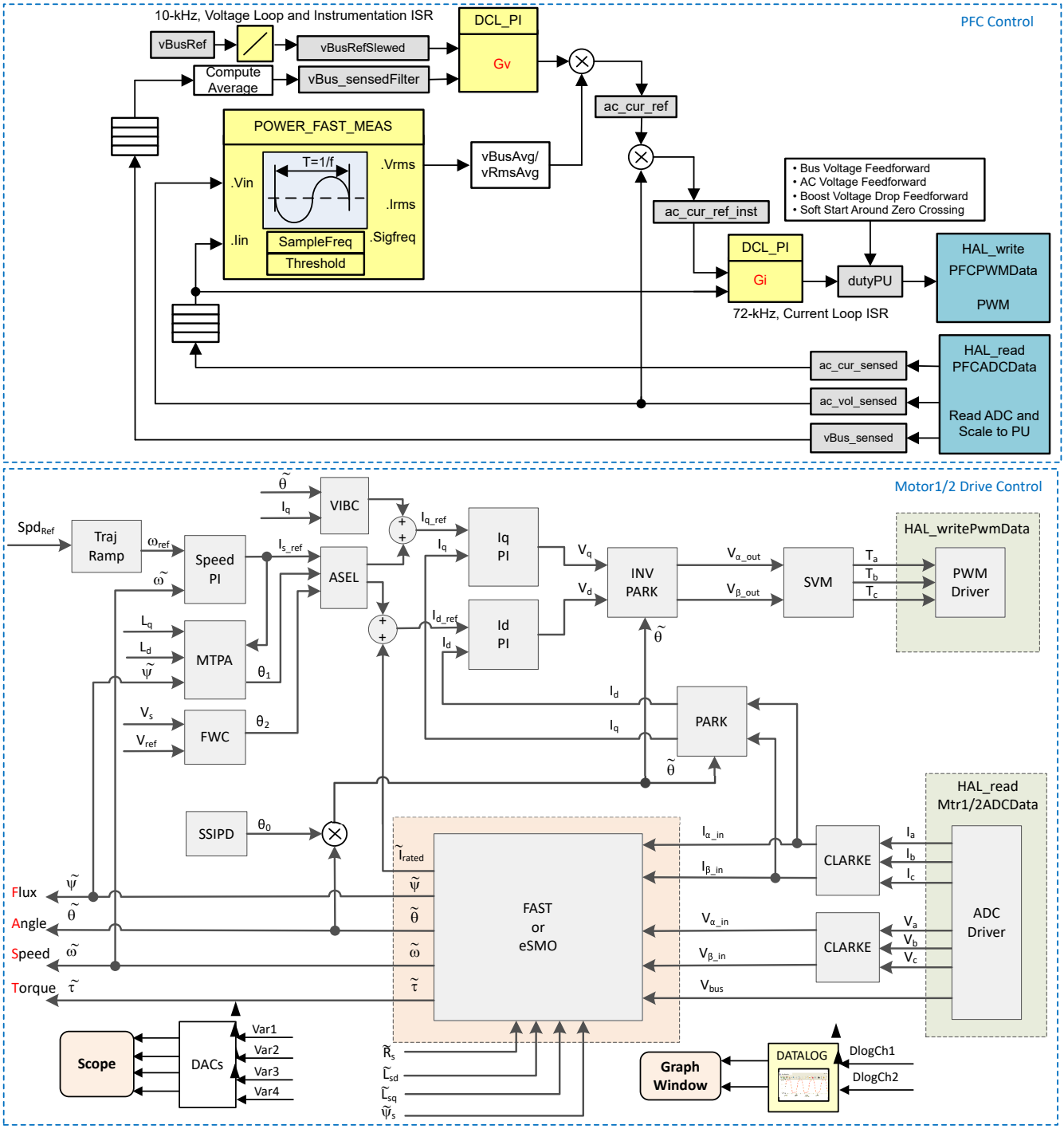


图 3-23. 控制软件方框图：构建级别 4 – 转速和电流闭环控制

3.3.4.1 启动 CCS 并打开工程

1. 将一个能够提供高达 2.0kW 通用交流输入的可编程隔离式交流电源连接至参考板的输入端子（连接器 J3 和 J4），如图 3-2 所示。将交流电源电流限制设置为 10A。此时不要打开电源。
2. 将电机（压缩机）连接至 J7/J8/J9，将电机（风扇）连接至 J12。
3. 按照节 3.3.1.1 中的步骤 2 和 3 打开工程。

3.3.4.2 构建和加载工程

1. 打开 sys_settings.h 文件，有两个增量构建选项（PFC_BUILDLEVEL、DMC_BUILDLEVEL），将 PFC_BUILDLEVEL 设置为 PFC_LEVEL_4，将 DMC_BUILDLEVEL 设置为 DMC_LEVEL_4。
2. 按照节 3.3.1.2 中的步骤 2 至 4 构建工程并将代码加载到控制器中。

3.3.4.3 设置调试环境窗口

1. 通过选择 *BuildLevel4.txt*，按照节 3.3.1.3 中的步骤 1 至 4 将变量导入“Expressions”窗口。此时将显示“Expressions”窗口，如图 3-16 所示。

3.3.4.4 运行代码

1. 将交流电源输出设置为 0V (50/60Hz)，打开交流电源，将输入电压从 0V 缓慢增加至 220V 交流。
2. 必须在头文件（*user_mtr1.h* 和 *user_mtr2.h*）中记录所需的电机参数，如以下示例代码所示。如果用户不太了解电机参数，那么在参考设计中实现了 FAST 估算器的情况下，可以使用电机识别来获得电机参数。

```
#define USER_MOTOR1_Rs_Ohm                  (2.66273594f)
#define USER_MOTOR1_Ls_d_H                  (0.00943629723f)
#define USER_MOTOR1_Ls_q_H                  (0.00943629723f)
#define USER_MOTOR1_RATED_FLUX_VpHz         (0.390171647f)
```

备注

基于 F28002x 的参考设计工程不支持电机识别，因此用户必须从电机制造商处获取电机参数并在 user_mtr1/2.h 文件中设置这些参数。基于 F28003x 的参考设计可支持电机参数识别功能。

3. 将 *userParams[MTR_1].flag_bypassMotorId* 值更改为“false”以启用电机识别，如以下电机 1（压缩机）示例代码所示。

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

4. 根据电动机 1（压缩机）的规格在 *user_mtr1.h* 中设置正确的识别变量值。

```
 #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           (4.5f)      // A - 30~150% of rated current of the
motor
#define USER_MOTOR1_FLUX_EXC_FREQ_Hz        (20.0f)     // Hz - 10~30% of rated frequency of the
motor
```

5. 重新构建工程并将代码加载到控制器中，通过点击按钮  来运行工程，或点击“Debug”选项卡中的 *Run → Resume*。经过固定的时长后，*systemVars.flagEnableSystem* 应设置为 *1*，这意味着偏移校准已完成并且浪涌电源继电器已开启。双电机和 PFC 的故障标志（*motorVars[0].faultMtrUse.all*、*motorVars[1].faultMtrUse.all* 和 *pfcVars.faultPFCUse.all*）应等于“*0*”，否则用户应按照节 3.3.1 中的说明检查电流和电压检测电路。
6. 在“Expressions”窗口中将变量 *motorVars[0].flagEnableRunAndIdentify* 设置为 *1*（如图 3-24 所示），此时将执行电机识别，整个过程需要大约 150s。一旦 *motorVars[0].flagEnableRunAndIdentify* 等于 *0*，就表明已识别了电机参数。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[0].Rs_Ohm 的值
   - USER_MOTOR1_Ls_d = motorVars[0].Ls_d_H 的值
   - USER_MOTOR1_Ls_q = motorVars[0].Ls_q_H 的值
   - USER_MOTOR_RATED_FLUX = motorVars[0].flux_VpHz 的值

7. 如果电机参数未知且需要识别，则对电机 2（风扇）执行相同的过程。
8. 成功识别电机参数后，将 *userParams[MTR_1].flag_bypassMotorId* 和 *userParams[MTR_2].flag_bypassMotorId* 都设置为“true”，重新构建工程并将代码加载到控制器中。
   - 再次将变量 *motorVars[0].flagEnableRunAndIdentify* 设置为 *1* 以开始运行电机 1（压缩机），再次将 *motorVars[1].flagEnableRunAndIdentify* 设置为 *1* 以开始运行电机 2（风扇）。
   - 将变量 *motorVars[0].speedRef_Hz* 和 *motorVars[1].speedRef_Hz* 设置为不同的值，并观察电机轴转速会如何变化。
   - 要更改加速度，请为变量 *motorVars[0].accelerationMax_Hzps* 输入不同的加速度值。
9. 要启用 PFC 控制，请将 *pfcVars.flagEnablePFC* 设置为 *1*，将变量 *pfcVars.VdcSet_V* 设置为不同的值，并观察直流总线电压随设置电压的变化情况。
10. 现在可以在将 *motorVars[0].flagEnableRunAndIdentify*、*motorVars[1].flagEnableRunAndIdentify* 和 *pfcVars.flagEnablePFC* 设置为 *0* 之前停止控制器并终止调试连接。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。
11. 通过点击“Terminate Debug Session” 或点击 *Run* → *Terminate* 来关闭 CCS 调试会话。

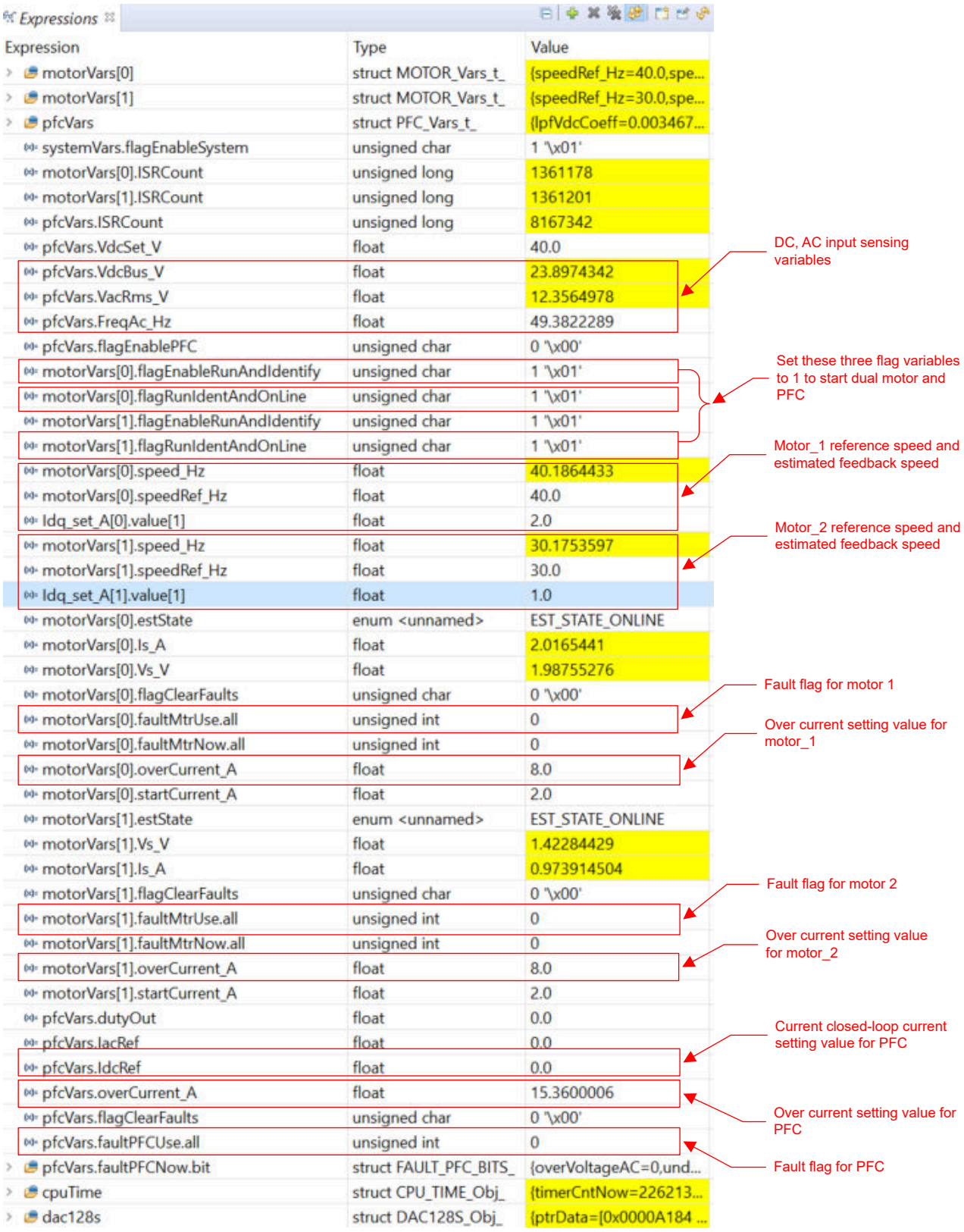


图 3-24. 版本级别 4：运行时的“Expressions”窗口

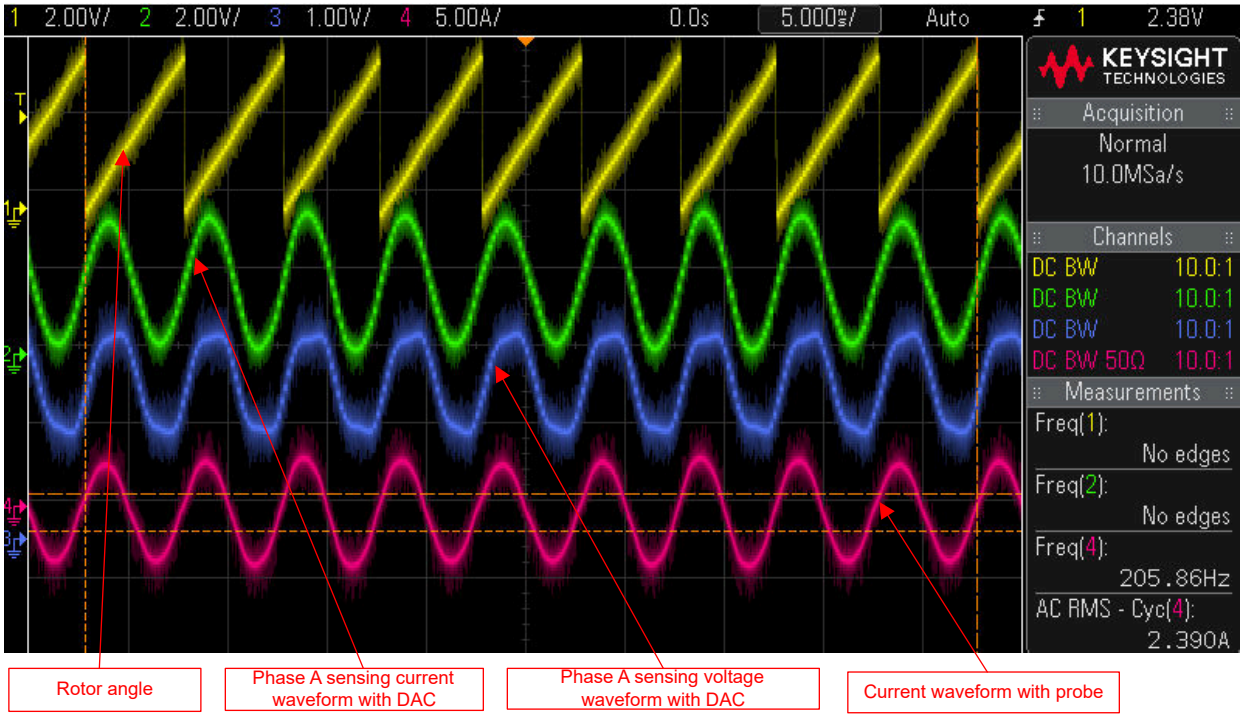


图 3-25. 版本级别 4：电机的转子角度、相电流

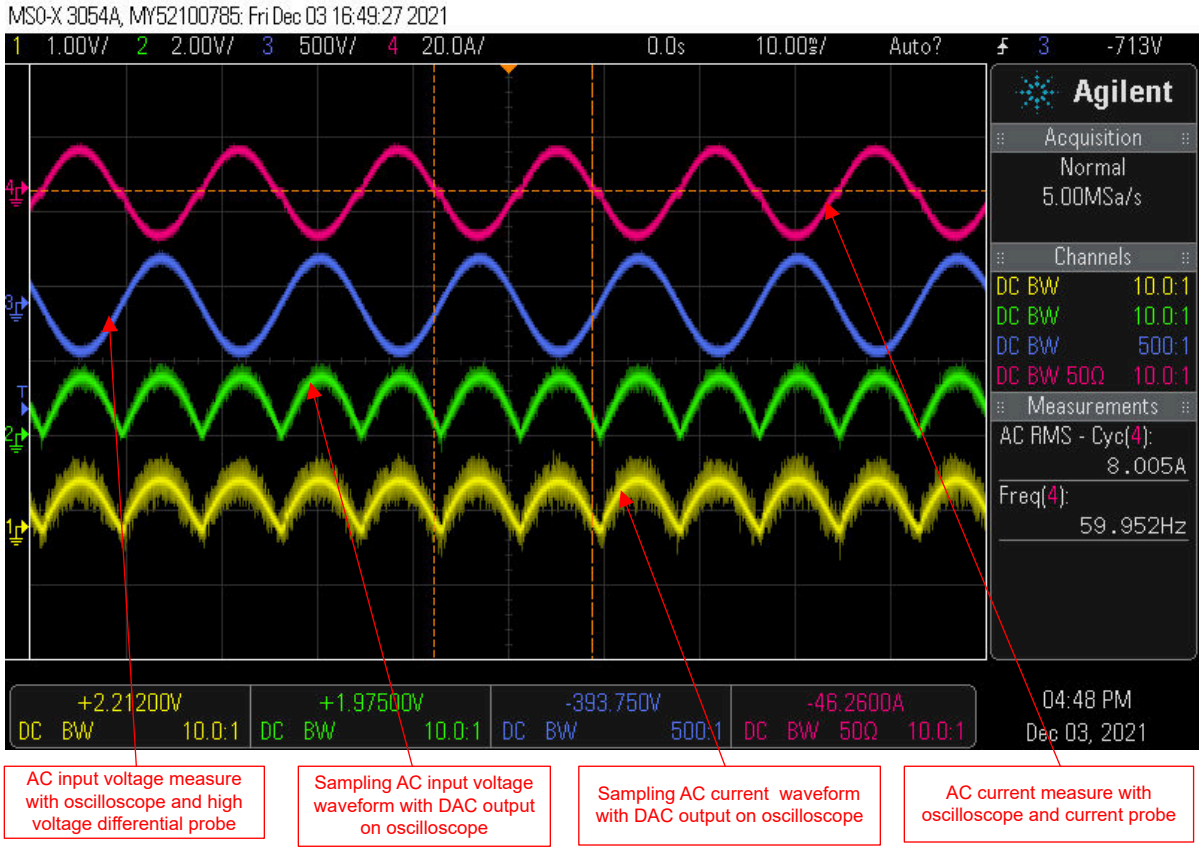


图 3-26. 版本级别 4：PFC 的电压和电流

3.3.4.5 运行系统

1. 将交流电源输出设置为 220V (50/60Hz)，打开交流电源。

2. 右键点击项目名称，然后点击 *Rebuild Project*。项目成功构建。点击 *Run → Debug*，这将启动调试会话。
3. 通过点击按钮  来运行工程，或点击“Debug”选项卡中的 *Run → Resume*。
4. 将变量 *systemVars.flagEnableSynCtrl* 设置为 *true* 以通过更改 PFC 设置电压和电机设置转速来控制系统。
5. 将变量 systemVars.flagEnableRun 设置为“true”。系统将按顺序启动 PFC、电机 2（风扇）和电机 1（压缩机）。
   - 将变量 *systemVars.pfcVdcbusTarget_V*、*systemVars.speedFan_Hz* 和 *systemVars.speedComp_Hz* 设置为不同的值，并观察电机转速和直流总线电压将如何变化。
   - 要更改加速度，请为变量 *systemVars.accelerationFan_Hzps* 和 *systemVars.accelerationComp_Hzps* 输入不同的加速度值。
6. 将变量 systemVars.flagEnableRun 设置为 *false*。系统将按顺序停止 PFC、电机 2（风扇）和电机 1（压缩机）。

3.3.4.6 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 和 G 参数是根据电机参数 Rs 和 Ls 计算得出的，如节 2.4.2.2.1.2 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 的角度以调整其参数。初始参数在 user-mtr1/2.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_Iq、motorVars[0].Ki_Iq、motorVars[0].Kp_Id 和 motorVars[0].Ki_Id。更改压缩机电机驱动器的 PI 增益并记录这些值。
- 在 CCS Debug 透视图中向“Expressions”窗口添加 motorVars[0].Kp_spd、motorVars[0].Ki_spd、motorVars[0].Kp_Iq、motorVars[0].Ki_Iq、motorVars[0].Kp_Id 和 motorVars[0].Ki_Id。更改风扇电机驱动器的 PI 增益并记录这些值。

3.3.4.7 调整 PFC 参数

TI SFRA 库旨在实现仅使用软件对数控电源转换器进行频率响应分析。由于不需要外部连接或设备，该功能可以相对轻松地执行电源转换器的频率响应分析。优化后的库可用于高频功率转换应用，以识别闭环功率转换器的受控对象和开环特性，可用于获取带宽、增益裕度和相位裕度等稳定性信息，以评估控制环路性能。有关更多信息，请参阅 C2000™ SFRA 库和补偿设计器用户指南。

除了 SFRA，该 PFC 设计还支持使用另一种称为补偿设计器的 powerSUITE 工具。补偿设计器工具允许对不同样式的补偿器进行 PFC 设计，以实现所需的闭环性能，这可以使用 SFRA 工具中测量的功率级或受控体数据来完成。必须在器件上编程的系数由补偿设计器生成，可以直接复制到代码中。这些工具可帮助用户评估整个系统，使其适应最终应用程序，并对其进行调整以提高性能。

3.3.4.8 调整弱磁和 MTPA 控制参数

在电机驱动器 ISR 中添加并调用了 FW 和 MTPA 函数来计算电流角，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 3.2.2 所述）以分别启用 FW 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以进行计算。
3. 在 CCS Debug 透视图中向“Expressions”窗口添加变量 VsRef_pu、Kp_fwc 和 Ki_fwc，并根据电机及其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值后，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 Kp 增益

   USER_M1_FWC_KI = Ki_fwc 的值。用于弱磁控制的 PI 稳压器 Ki 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。

3.3.4.9 调整快速启动控制参数

风扇驱动器默认启用快速启动。在 CCS Debug 透视图中向“Expressions”窗口添加变量 flyingStartSpeed_Hz 和 flyingStartTimeDelay 并进行调整，使用 user_mtr2.h 文件中新定义的参数调整和记录监视窗口值。

USER_MOTOR2_SPEED_FLYST_Hz = flyingStartSpeed_Hz 的值

USER_MOTOR2_DELAY_FLYST_s = flyingStartTimeDelay 的值

3.3.4.10 调整振动补偿参数

添加了自动振动补偿功能，并在电机驱动 ISR 中进行调用以计算前馈扭矩电流。

1. 在工程的构建配置中添加预定义符号 *MOTOR1_VIBCOMPA*（如节 3.2.2 所述）以启用振动补偿。
2. 在 CCS Debug 透视图中向“Expressions”窗口添加变量 vibCompAlpha、vibCompGain 和 vibCompIndexDelta，并根据压缩机和空调系统调整这些参数以实现预期的振动补偿性能。
   - vibCompAlpha 用作学习速度。该值越高（最大值为 1.0），学习算法的速度就越慢。不过，取较高的值也有好处，可以提供抗噪性。
   - vibCompIndexDelta 将输出波形提前一点点，以便在机械角达到该点时产生的电流非常接近所需的值。建议使用典型值 10，但最终需要用户进行微调。
   - vibCompGain 是前馈扭矩基准扭矩电流值的增益系数（最大值为 1.0）。
3. 更改转速基准 (motorVars[0].speedRef_Hz) 和转速控制器增益（motorVars[0].Kp_spd 和 motorVars[0].Ki_spd）。该步骤用于使电机由于脉动负载而振动。为了使振动补偿更好地工作，增加转速控制器增益的值。不过，前提是要确保转速控制器保持稳定。
4. 在工程的构建配置中添加预定义符号 *DEBUG_MONITOR_EN*（如节 3.2.2 所述）以启用电机运行转速振动。现在，通过设置标志 vibCompFlagEnable = 1 来启用振动补偿输出。然后使其运行 5 至 10 秒，然后通过设置位 motorVars[0].flagClearRecord = 1 来实现新转速变化。记录转速变化约为 2Hz。
5. 如果振动没有减少，请尝试增加转速控制器增益。此外，还可以尝试通过每次将 vibCompAlpha 的值递减 0.02 来提高振动补偿算法的学习速度，因此可以尝试 0.99、0.97、0.95 等，每次更改 vibCompAlpha 时，使其运行几秒钟并通过重置该计算来读取转速变化：motorVars[0].flagClearRecord = 1。
6. 调整并修正这些变量值后，使用 user_mtr1.h 文件中新定义的参数记录监视窗口值。

   USER_MOTOR1_VIBCOMPA_ALPHA = vibCompAlpha 的值。振动补偿模块的学习速率范围为 0.0 至 1.0。

   USER_MOTOR1_VIBCOMPA_GAIN = vibCompGain 的值。振动补偿模块的增益范围为 0.0 至 1.0。

   USER_MOTOR1_VIBCOMPA_INDEX_DELTA = vibCompIndexDelta 的值。振动补偿模块的相位提前范围为 0 至 360。

基于压缩机扭矩与角度的电机电流控制是对抗转速纹波变化的替代技术。在工程的构建配置中添加预定义符号 *MOTOR1_VIBCOMPT*（如节 3.2.2 所述）以启用该振动补偿方法。

根据滚动活塞角度，可以在转速 PI 控制器输出中添加或减去额外的扭矩电流分量。需要在压缩阶段添加电流，并在排气期间减去电流（它有助于活塞运动，使转速增加），并且可以根据经验计算其幅度以匹配压缩机的扭矩曲线。算法将 360 度机械角分成 3 个扇区，可以通过变量 VibCompAlpha0、120 和 240 分别添加补偿电流。经过粗调补偿后，转速纹波从 200Hz 减小至 100Hz 以下 (1200rpm)。如果更加贴合扭矩曲线进行调整，可以进一步降低转速纹波。通常对介于 1200 - 2000rpm (100Hz) 之间的压缩机转速启用振动补偿，这往往可以减小其影响。

3.3.4.11 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      225           // 2.25us

//! \brief Defines the PWM deadband rising edge delay count (system clocks)
#define MTR1_PWM_DBRED_CNT      245           // 2.45us
```

- 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   (480U)
```

- 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 的开通/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   (455U)
```

3.4 测试结果

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。

3.4.1 性能数据和曲线

表 3-2 显示了各种转速、不同负载下的风扇电机测试数据，电机与 HD-705-6N 测力计耦合以增加负载。

表 3-2. 不同转速和负载下风扇电机和逆变器的功率和效率

$SPEED_{ref}$ (rpm)	$SPEED_{error}$ (rpm)	T_{load} (N.m)	P_{o_motor} (W)	$P_{o\_inverter}$ (W)	η_{motor} (%)	$P_{in\_inverter}$ (W)	$\eta_{inverter}$ (%)
750	3	0.7863	61.8041	89.2126	69.28	95.5376	93.38
1500	3	0.8386	131.8089	168.3758	78.28	182.5367	92.24
2250	4	0.8907	210.0507	256.1420	82.01	276.9516	92.49
1500	3	0.8456	132.8883	168.9123	78.67	183.0125	92.30
2250	4	0.8952	211.1273	256.1164	82.44	275.9552	92.81
3000	3	0.9482	298.1318	352.3146	84.62	368.9679	95.49
2250	4	0.9037	213.0801	257.8850	82.63	277.8515	92.81
1500	4	0.8542	134.2593	169.7001	79.12	183.6163	92.42
750	2	0.8035	63.1298	90.2685	69.94	98.2091	91.91

表 3-3 显示了各种转速、不同负载下的压缩机电机测试数据，电机与 HD-815-6N 测力计耦合以增加负载。

表 3-3. 不同转速和负载下压缩机电机和逆变器的功率和效率

$Speed_{ref}$ (rpm)	$Speed_{error}$ (rpm)	T_{load} (N.m)	P_{o_motor} (W)	$P_{o\_inverter}$ (W)	η_{motor} (%)	$P_{in\_inverter}$ (W)	$\eta_{inverter}$ (%)
750	2	1.9845	155.8340	229.9068	67.74	251.6578	91.36
1500	4	2.3945	376.1113	521.2635	72.15	569.9387	0.91
2250	5	4.5485	1071.9087	1346.4348	79.61	1372.200	98.12
1500	5	4.2020	660.0950	864.5136	76.36	903.3708	95.70
750	3	5.3235	417.9971	586.1813	71.31	642.0582	91.30
1500	6	5.6984	895.2029	1153.2485	77.63	1212.664	95.10
750	2	5.2779	414.3357	582.2262	71.17	636.5326	91.47

图 3-27 显示了在负载扭矩变化的系统中不同转速下测得的风扇电机驱动效率。

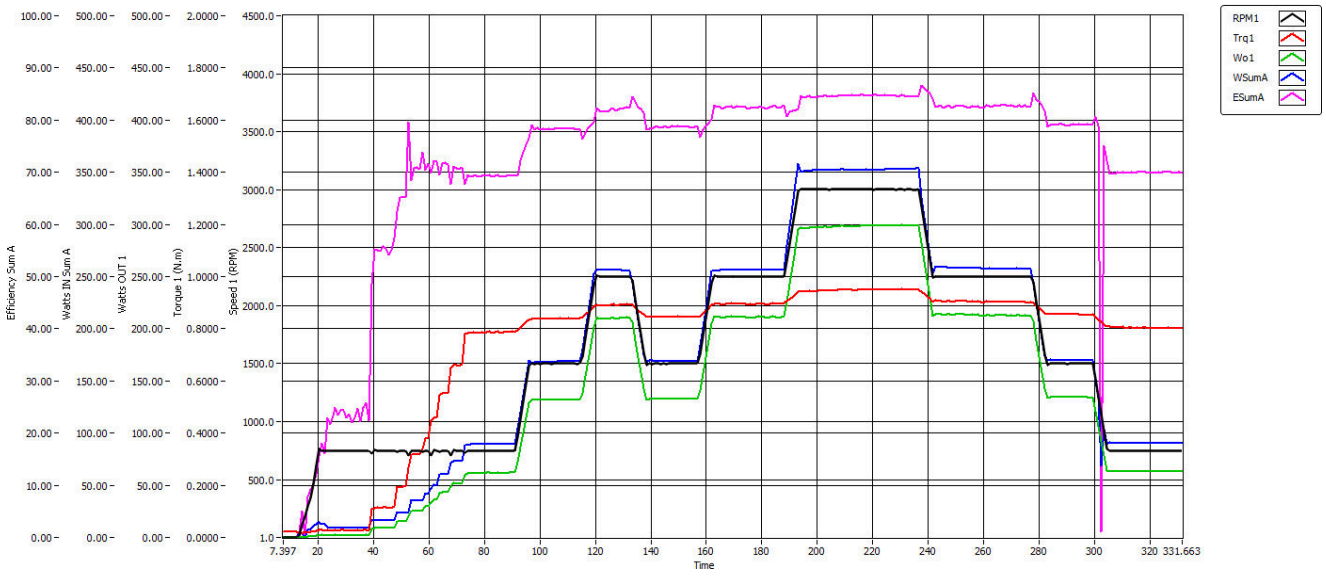


图 3-27. 各种转速和负载下的风扇电机输出功率和效率

图 3-28 显示了在负载扭矩变化的系统中不同转速下测得的风扇电机驱动效率。

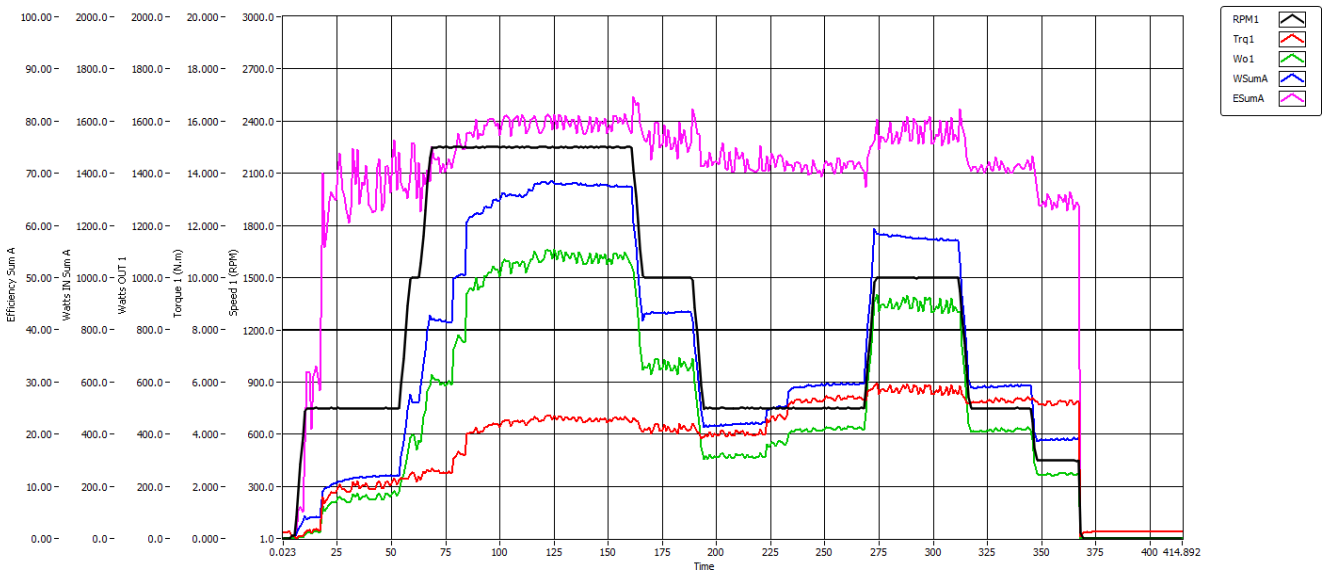


图 3-28. 各种转速和负载下的压缩机电机输出功率和效率

表 3-4、表 3-5 和表 3-6 显示了不同负载下输入为 165V VAC、220V VAC 和 265V VAC 时的 PFC 性能数据。

表 3-4. 165V VAC 输入下的 PFC 性能数据

V_{INAC} (V)	I_{INAC} (A)	P_{INAC} (W)	PF	THDi (%)	V_{OUT} (V)	I_{OUT} (A)	P_{OUT} (W)	效率 (%)
165.0	0.89	145.6	0.982	13.29	376.8	0.363	137	94.1
165.0	1.75	287.6	0.987	11.69	376.5	0.729	275	95.6
165.0	2.62	430.9	0.992	8.57	376.1	1.101	415	96.3
165.0	3.49	572.9	0.991	9.65	376.4	1.465	552	96.4
165.0	4.38	715.9	0.989	9.15	376.5	2.194	827	96.5
165.0	5.22	857.3	0.991	7.76	376.5	2.194	827	96.5
165.0	6.11	1002.5	0.992	6.39	376.8	2.559	966	96.4
165.0	6.99	1146.1	0.993	5.49	377.3	2.926	1106	96.5
165.0	7.87	1295.5	0.994	4.97	377.5	3.301	1248	96.3
165.0	8.96	1465.5	0.995	3.75	380.2	3.715	1457	99.4
165.0	9.98	1642.4	0.996	3.39	380.3	4.097	1629	99.18
165.0	10.67	1757.6	0.997	3.12	380.4	4.427	1747	99.39

表 3-5. 220V VAC 输入下的 PFC 性能数据

V_{INAC} (V)	I_{INAC} (A)	P_{INAC} (W)	PF	THDi (%)	V_{OUT} (V)	I_{OUT} (A)	P_{OUT} (W)	效率 (%)
220.1	0.686	149.2	0.982	14.9	376.9	0.368	139	93.2
220.1	1.329	287.5	0.982	14.29	377.2	0.731	276	96.0
220.1	1.94	427.0	0.982	9.25	377.1	1.096	414	96.9
220.1	2.626	572.8	0.992	8.01	377	1.47	555	97.1
220.1	3.24	709.5	0.989	7.285	376.9	1.827	689	97.4
220.1	3.86	848.3	0.991	5.958	376.8	2.191	827	97.3
220.1	4.52	990.5	0.993	6.79	376.7	2.556	965	97.4
220.1	5.21	1138.5	0.992	6.4	376.6	2.938	1108	97.4
220.1	5.8	1271.5	0.993	5.98	376.6	3.283	1238	97.4
220.1	6.45	1416.5	0.994	5.35	376.6	3.656	1379	97.4
220.1	7.07	1549.5	0.994	4.93	376.6	4.003	1510	97.5
220.1	7.77	1709.8	0.995	4.29	376.6	4.404	1661	97.22

表 3-6. 265V VAC 输入下的 PFC 性能数据

V_{INAC} (V)	I_{INAC} (A)	P_{INAC} (W)	PF	THDi (%)	V_{OUT} (V)	I_{OUT} (A)	P_{OUT} (W)	效率 (%)
250.0	0.61	148.7	0.965	12.6	385.9	0.368	142	95.5
250.0	1.24	298.9	0.959	8.65	385.9	0.751	290	97.0
250.0	1.81	447.5	0.977	9.97	385.8	1.124	434	97.0
250.0	2.42	595.8	0.979	8.82	385.7	1.499	579	97.2
250.0	3.02	744.3	0.981	11.65	385.7	1.876	724	97.3
250.0	3.62	889.6	0.979	12.03	385.7	2.246	868	97.6
250.0	4.21	1034.9	0.982	10.95	385.6	2.619	1012	97.8
250.0	4.79	1183.7	0.985	9.75	385.6	2.993	1156	97.7
250.0	5.39	1335.2	0.988	7.39	385.6	3.374	1303	97.6
250.0	5.99	1488.9	0.991	5.93	385.8	3.759	1452	97.5
250.0	6.59	1640.1	0.992	5.12	385.9	4.137	1600	97.6
250.0	7.09	1768.6	0.993	4.79	384.9	4.478	1727	97.6

3.4.2 函数波形

图 3-29 是风扇电机以 150Hz 的频率和 1N.m 的负载扭矩运行时的屏幕截图。会显示以下内容：

- 通道 1（黄色）：DAC 输出的转子角度
- 通道 2（绿色）：DAC 输出的检测相电流
- 通道 3（紫色）：DAC 输出的检测相电压
- 通道 4（粉色）：使用电流探头测量的相电流

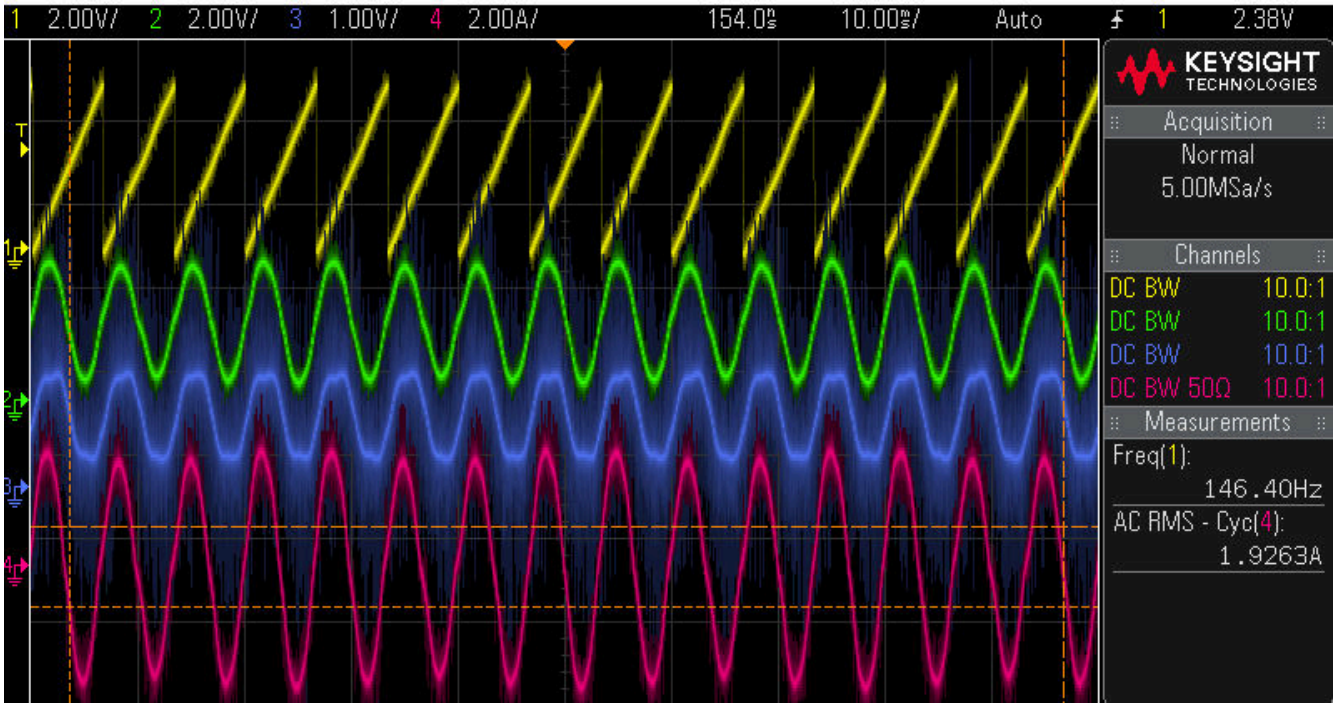


图 3-29. 150Hz 下风扇电机的相电流和电压波形

图 3-29 是压缩机电机以 100Hz 的频率和 5N.m 的负载扭矩运行时的屏幕截图。该屏幕截图显示

- 通道 1（黄色）：DAC 输出的转子角度
- 通道 2（绿色）：DAC 输出的检测相电流
- 通道 3（紫色）：DAC 输出的检测相电压
- 通道 4（粉色）：使用电流探头测量的相电流

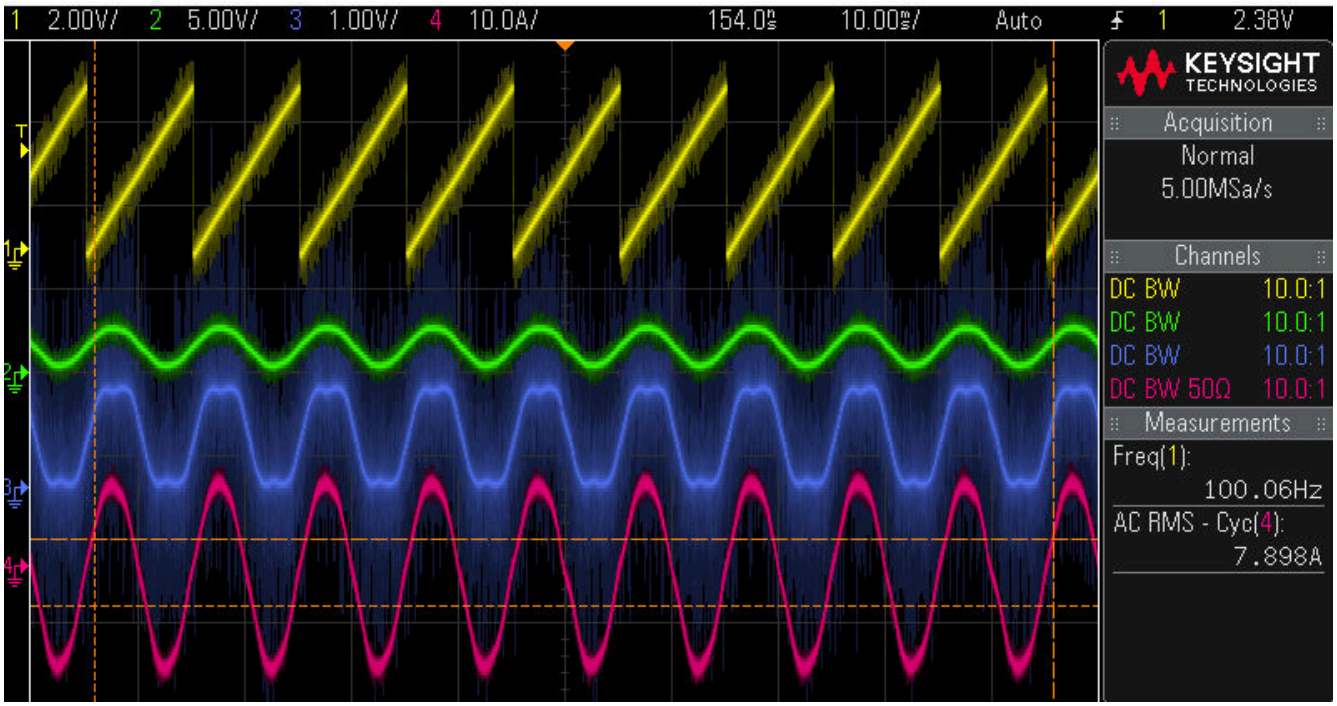


图 3-30. 100Hz 下压缩机电机的相电流和电压波形

快速启动是在风扇控制中实现的，它能够在启动前检测电机是否正在运行，并在不需要加速时跳过加速阶段。风扇电机从一开始就在无传感器 FOC 中运行，无需在启动前停止。

图 3-29 是风扇电机在没有快速启动功能的情况下运行时的屏幕截图。该屏幕截图显示

- 通道 1（黄色）：DAC 输出的转子角度
- 通道 2（绿色）：DAC 输出的检测相电流
- 通道 4（粉色）：使用电流探头测量的相电流

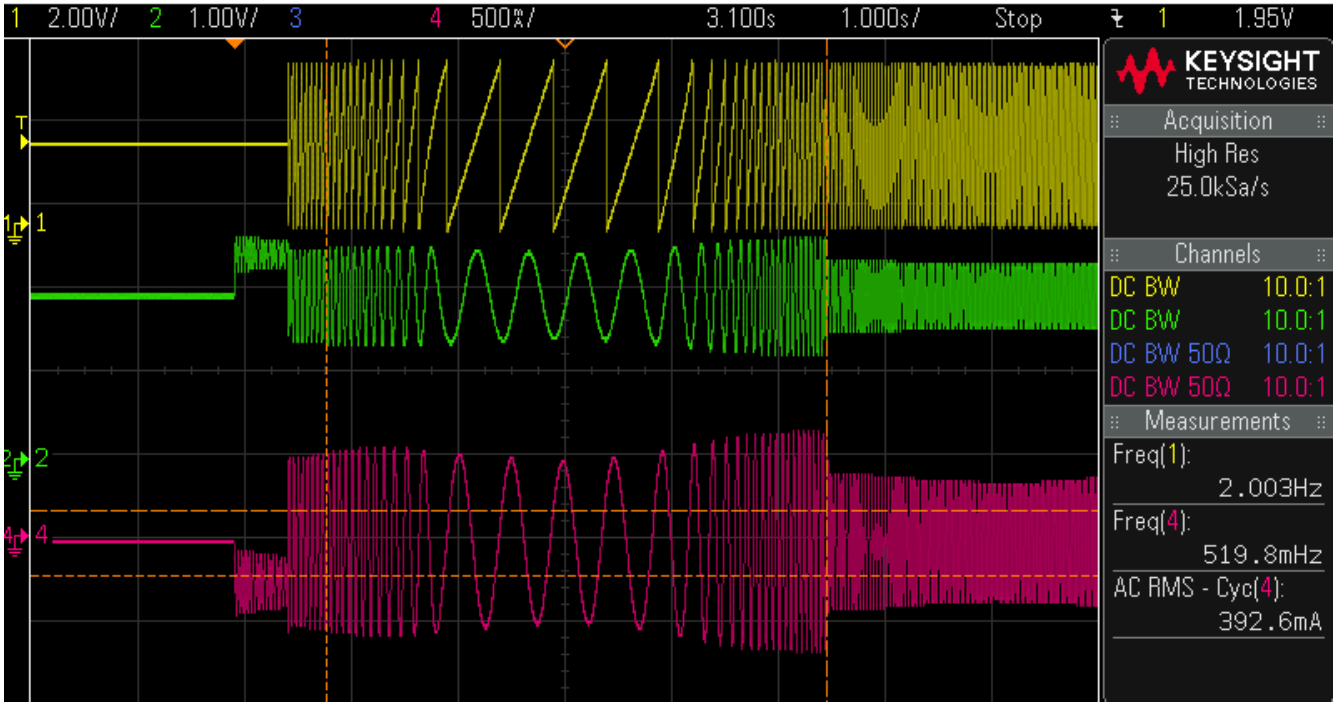


图 3-31. 无快速启动时风扇电机的相电流和转子角度

图 3-32 是风扇电机在有快速启动功能的情况下运行时的屏幕截图。该屏幕截图显示

- 通道 1（黄色）：DAC 输出的转子角度
- 通道 2（绿色）：DAC 输出的检测相电流
- 通道 4（粉色）：使用电流探头测量的相电流

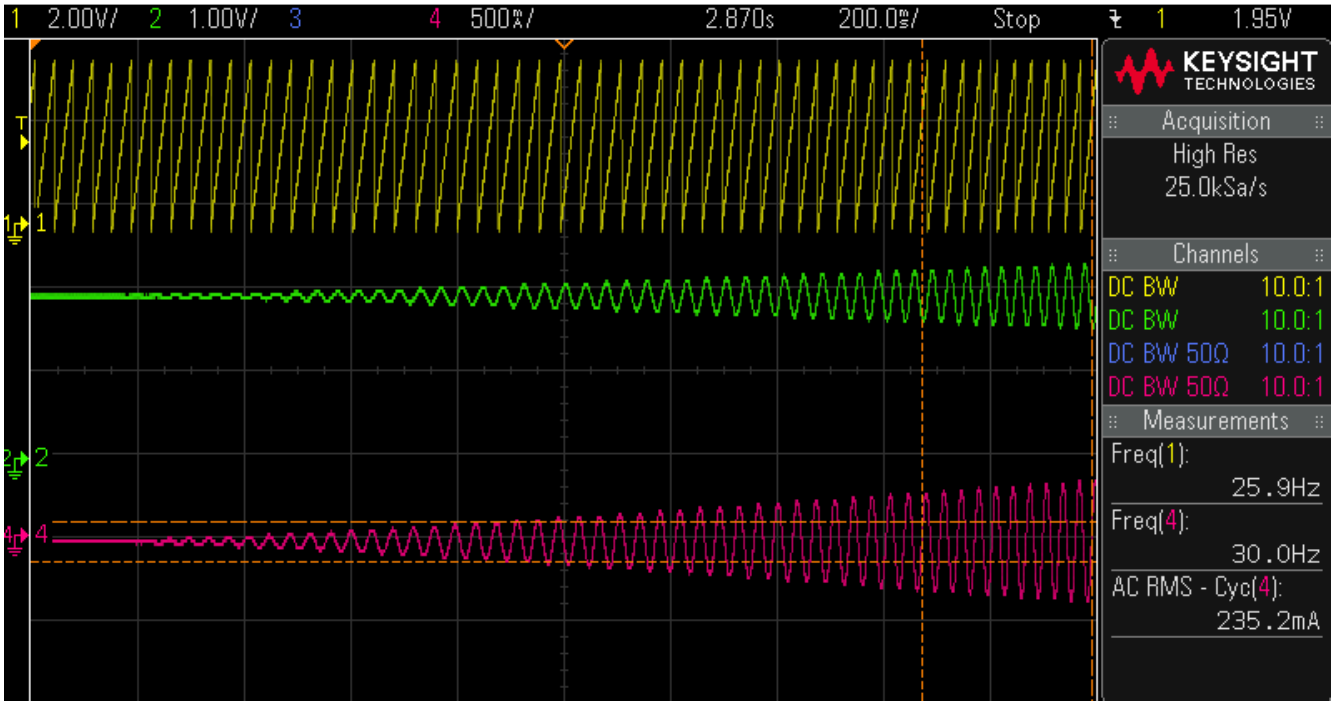


图 3-32. 有快速启动时风扇电机的相电流和转子角度

图 3-33 是风扇和压缩机电机以 100Hz 的频率和高负载运行时的系统屏幕截图。该屏幕截图显示

- 通道 1（黄色）：DAC 输出的风扇电机转子角度
- 通道 2（绿色）：DAC 输出的压缩机电机转子角度
- 通道 3（紫色）：使用电流探头测量的风扇电机相电流
- 通道 4（粉色）：使用电流探头测量的压缩机相电流

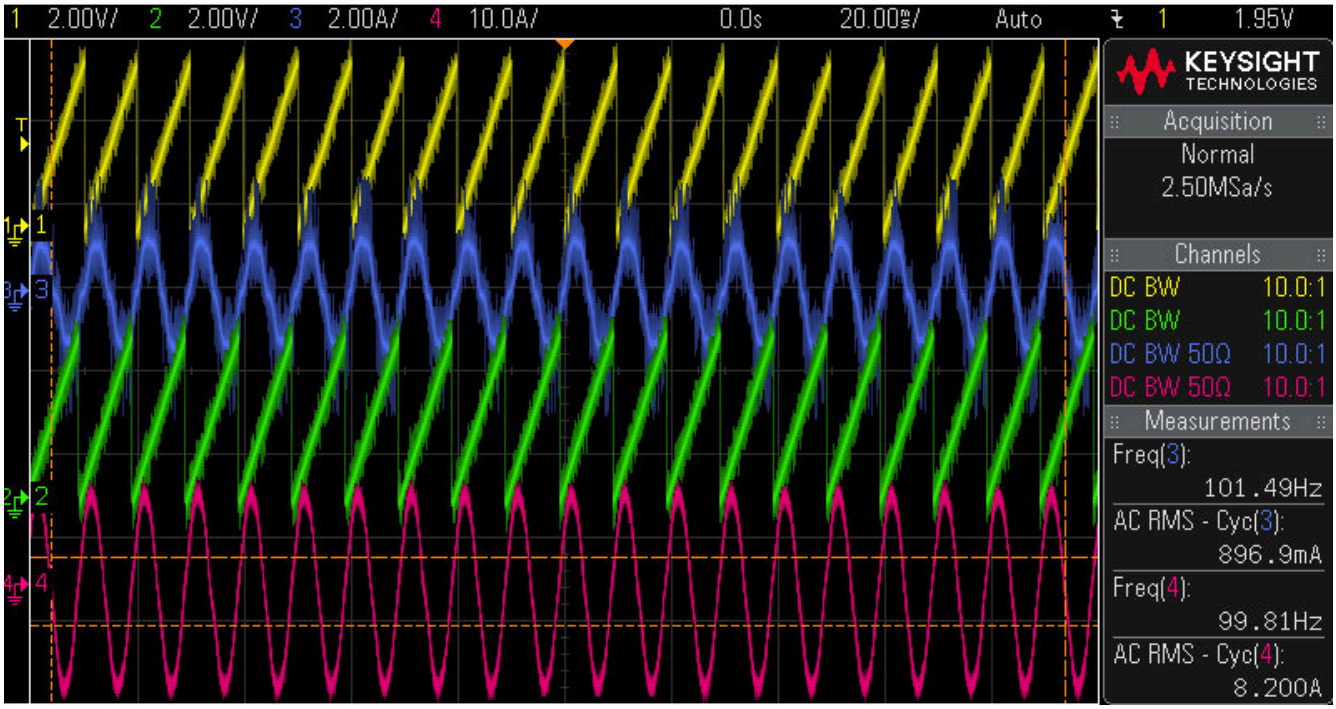


图 3-33. 风扇和压缩机电机的相电流和转子角度

3.4.3 瞬态波形

风扇电机采用 HD-705-6N 测力计，该测力计是使用 DSP6001 控制器和 M-TEST 软件的控制器。

图 3-34 显示了不同转速、恒定负载下的风扇电机输出功率、逆变器输入和输出功率。

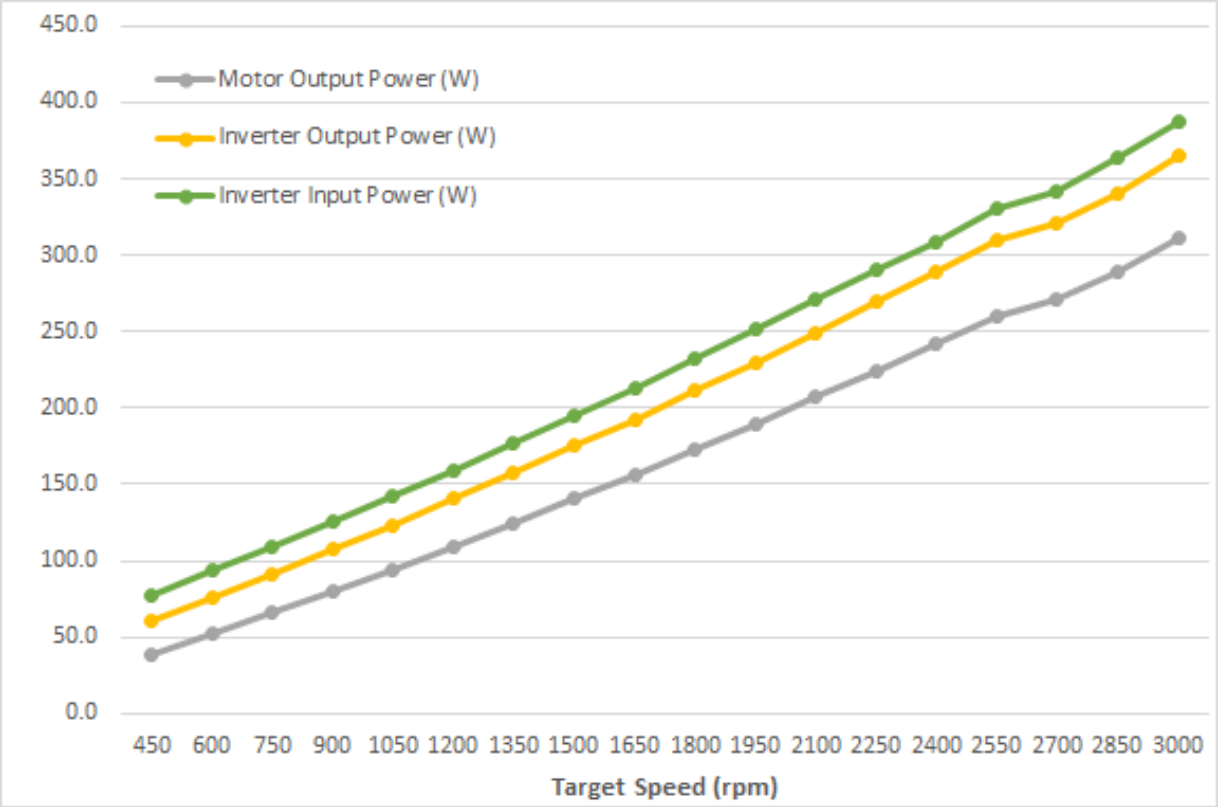


图 3-34. 不同转速、恒定负载下的风扇电机输出功率、逆变器输入和输出功率

图 3-35 显示了不同转速、恒定负载下的风扇电机输出和逆变器效率。

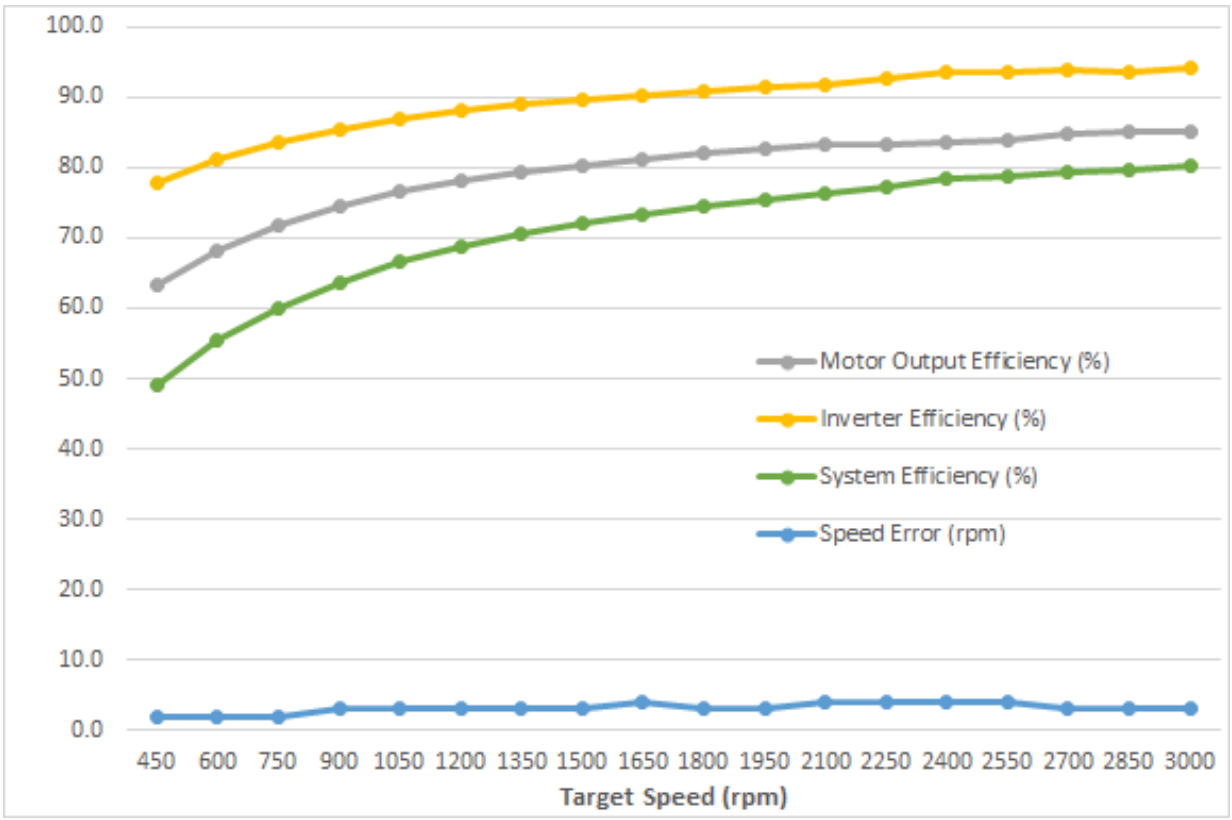


图 3-35. 不同转速、恒定负载下的风扇电机输出和逆变器效率

图 3-36 显示了不同转速、恒定负载下的压缩机电机输出功率、逆变器输入和输出功率。

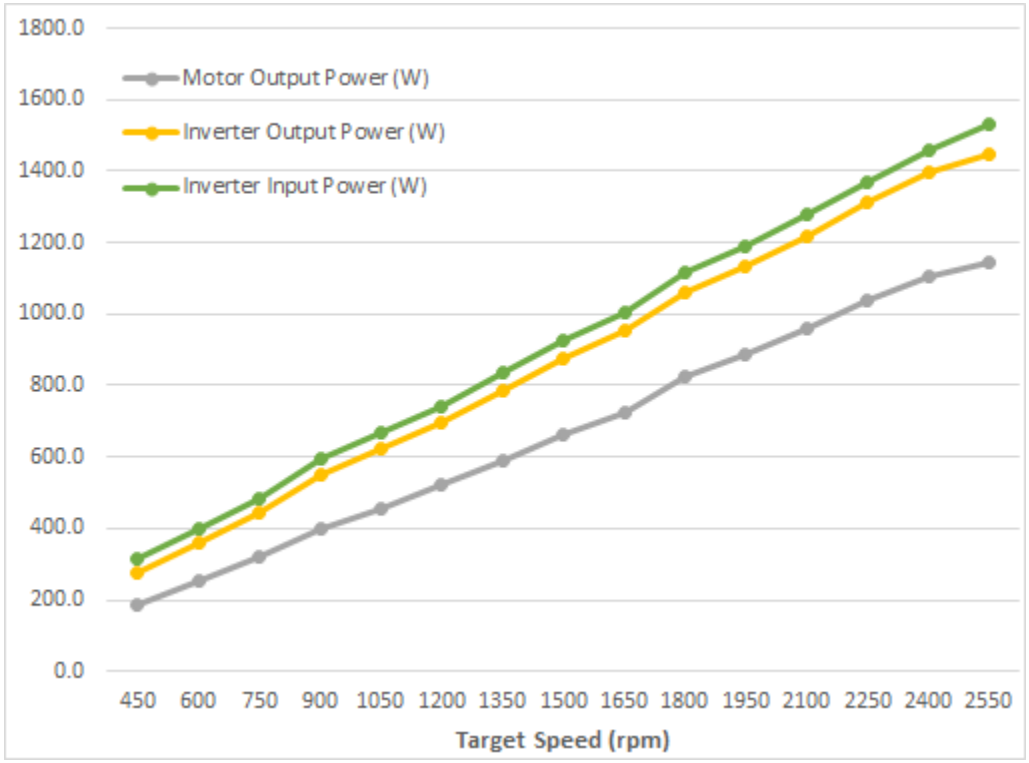


图 3-36. 不同转速、恒定负载下的压缩机电机输出功率、逆变器输入和输出功率

图 3-37 显示了不同转速、恒定负载下的压缩机电机输出和逆变器效率。

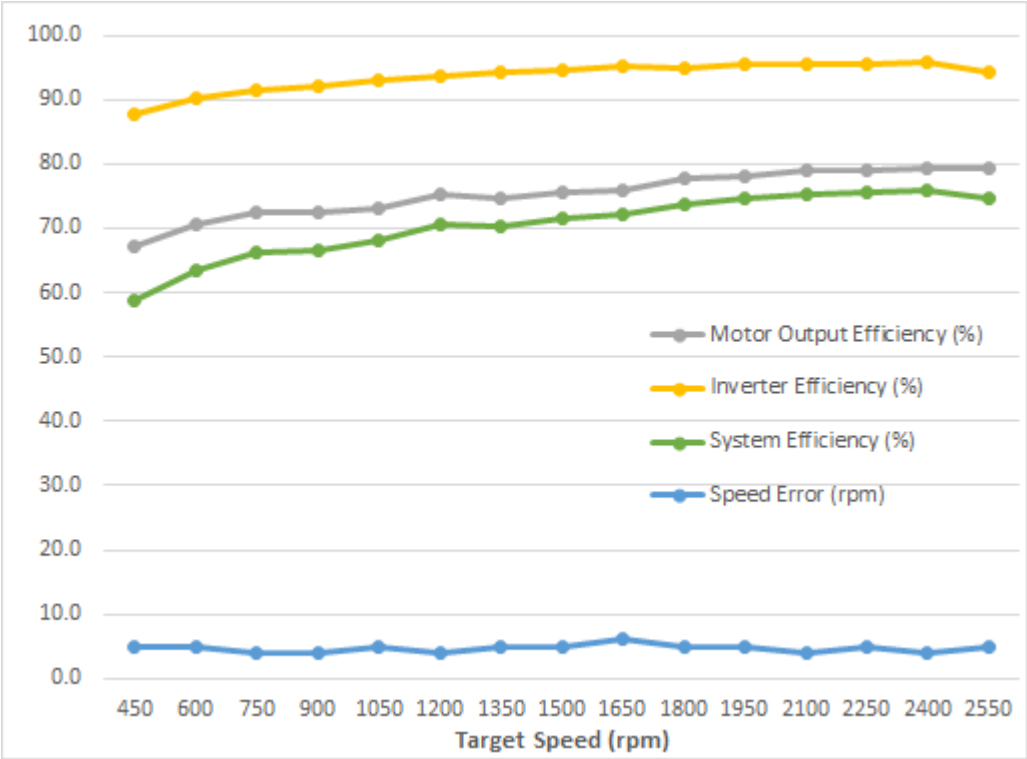


图 3-37. 不同转速、恒定负载下的压缩机电机输出和逆变器效率

图 3-38 显示了不同转速、恒定负载下采用 FAST 或 eSMO 控制技术时的风扇电机输出效率。

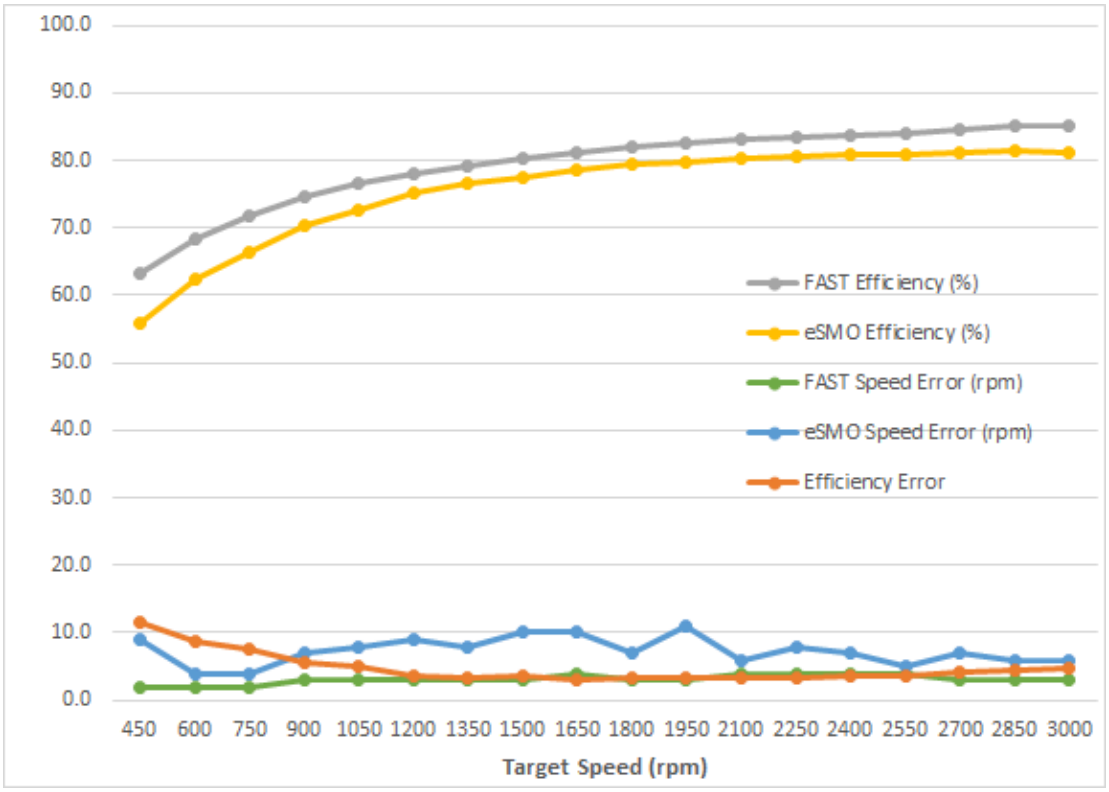


图 3-38. 不同转速、恒定负载下采用 FAST 或 eSMO 控制技术时的风扇电机输出效率

图 3-39 显示了不同转速和负载下的可变瞬态、功率和效率。

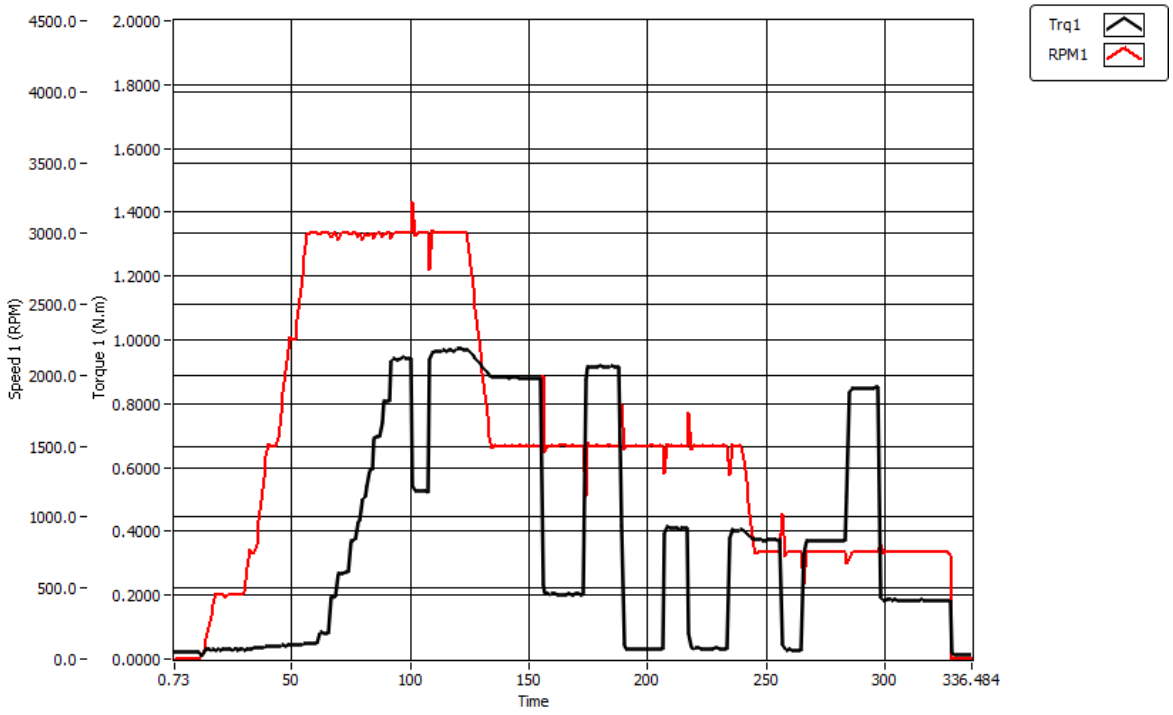


图 3-39. 不同转速和负载下的负载瞬态测量值

3.4.4 MCU CPU 负载、存储器和外设使用

3.4.4.1 完全实现的 CPU 负载

表 3-7 显示了在对双电机控制使用完全 InstaSPIN-FOC 实现（使用具有 100MHz CPU 的 F280025C 上的 PFC）、将用户代码加载到闪存中并将 ISR 代码复制到 RAM 中以进行执行时使用的 CPU 周期数、CPU 负载和可用的 MIPS。

表 3-7. 使用 FAST 进行双电机控制时的 F28002x CPU 负载

CPU = 100MHz	用于 ISR 的最大 CPU 周期数	最大 CPU 利用率 [%]	使用的最大 MIPS [MIPS]
PWM = 72kHz、ISR = 36kHz 时的 PFC	465	16.74	16.74
PWM = 6kHz、ISR = 6kHz 时的电机 1	2372	14.23	14.23
PWM = 18kHz、ISR = 6kHz 时的电机 2	2226	13.36	13.36
总 CPU 利用率		**44.33**	**44.33**
可用的 CPU MIPS			**55.67**

表 3-8 显示了在对双电机控制使用完全 eSMO 实现（使用具有 100MHz CPU 的 F280025C 上的 PFC）、将用户代码加载到闪存中并将 ISR 代码复制到 RAM 中以进行执行时使用的 CPU 周期数、CPU 负载和可用的 MIPS。

表 3-8. 使用 eSMO 进行双电机控制时的 F28002x CPU 负载

F280025C CPU = 100MHz 可用 MIPS = 100MIPS	用于 ISR 的最大 CPU 周期数	最大 CPU 利用率 [%]	使用的最大 MIPS [MIPS]
PWM = 72kHz、ISR = 36kHz 时的 PFC	465	16.74	16.74
PWM = 6kHz、ISR = 6kHz 时的电机 1	1745	10.47	10.47
PWM = 18kHz、ISR = 6kHz 时的电机 2	1568	9.41	9.41
总 CPU 利用率 [%]		**36.62**	

表 3-8. 使用 eSMO 进行双电机控制时的 F28002x CPU 负载 (continued)

F280025C CPU = 100MHz 可用 MIPS = 100MIPS	用于 ISR 的最大 CPU 周期数	最大 CPU 利用率 [%]	使用的最大 MIPS [MIPS]
可用 CPU MIPS [MIPS]			**63.38**

表 3-9 显示了在对双电机控制使用完全 InstaSPIN-FOC 实现（使用具有 120MHz CPU 的 F280039C/F2800137 上的 PFC）、将用户代码加载到闪存中并将 ISR 代码复制到 RAM 中以进行执行时使用的 CPU 周期数、CPU 负载和可用的 MIPS。

表 3-9. 使用 FAST 进行双电机控制时的 F28003x/F280013x CPU 负载

CPU = 120MHz	用于 ISR 的最大 CPU 周期数	最大 CPU 利用率 [%]	使用的最大 MIPS [MIPS]
PWM = 72kHz、ISR = 时的 PFC	465	13.95	16.74
PWM = 6kHz、ISR = 6kHz 时的电机 1	2372	11.86	14.23
PWM = 18kHz、ISR = 6kHz 时的电机 2	2226	11.13	13.36
总 CPU 利用率 [%]		**36.94**	
可用 CPU MIPS [MIPS]			**75.67**

表 3-10 显示了在对双电机控制使用完全 eSMO 实现（使用具有 120MHz CPU 的 F280039C/F2800137 上的 PFC）、将用户代码加载到闪存中并将 ISR 代码复制到 RAM 中以进行执行时使用的 CPU 周期数、CPU 负载和可用的 MIPS。

表 3-10. 使用 eSMO 进行双电机控制时的 F28003x/F280013x CPU 负载

F280039C CPU = 120MHz 可用 MIPS = 120MIPS	用于 ISR 的最大 CPU 周期数	最大 CPU 利用率 [%]	使用的最大 MIPS [MIPS]
PWM = 72kHz、ISR = 32kHz 时的 PFC	465	13.95	16.74
PWM = 6kHz、ISR = 6kHz 时的电机 1	1745	8.72	10.47
PWM = 18kHz、ISR = 6kHz 时的电机 2	1568	7.84	9,41
总 CPU 利用率 [%]		**30.52**	
可用 CPU MIPS [MIPS]			**83.38**

3.4.4.2 存储器使用

显示了在 TMS320F280025C、TMS320F280039C 或 TMS320F2800137 MCU 上运行应用所需的存储器大小。微控制器存储器的很大一部分仍可用于其他任务。

表 3-11. 采用 FAST 的双电机控制的存储器使用

存储器类型	F280025C 上 使用的存储器	F280025C 上的可用存储器	F280025C 存储器利用率	F280039C 上 使用的存储器	F280039C 上的可用存储器	F280039C 存储器利用率	F2800137 上 使用的存储器	F2800137 上的可用存储器	F2800137 存储器利用率
闪存	**45.4 KB**	128KB	35.5%	52.8 KB	384 KB	13.8%	50.3 KB	256KB	19.6%
RAM	**17.7 KB**	24 KB	73.7%	21.3 KB	69 KB	30.8%	20.7 KB	36 KB	57.6%

表 3-12. 采用 eSMO 的双电机控制的存储器使用

存储器类型	F280025C 上使用的存储器	F280025C 上的可用存储器	F280025C 存储器利用率	F280039C 上使用的存储器	F280025C 上的可用存储器	F280039C 存储器利用率	F2800137 上使用的存储器	F2800137 上的可用存储器	F2800137 存储器利用率
闪存	**38.0 KB**	128KB	13.8%	39.4 kB	384 KB	12.3%	37.0 KB	256KB	14.4%
RAM	**15.3 KB**	24 KB	30.9%	15.7 KB	69 KB	22.8%	15.6 KB	36 KB	43.3%

3.4.4.3 外设使用

表 3-13 列出了该参考设计使用的 F28002x 或 F28003x 的外设和引脚。不允许将这些外设用于任何其他目的，但用户可以根据其硬件设计修改引脚分配。

表 3-13. F28002x/003x/0013x MCU 外设使用

模块	用途	注释
EPWM1、EPWM3、EPWM5	电机 1 三相 PWM	总共 6 个 PWM 通道
EPWM2、EPWM4、EPWM6	电机 2 三相 PWM	总共 6 个 PWM 通道
EPWM7	PFC 两相 PWM	总共 2 个 PWM 通道
ADCA、ADCC	电机 1 和 2 电流	3 个 ADC 通道，用于对每个电机进行电流检测
ADCA、ADCC	电机 1 和 2 电压	仅 InstaSPIN-FOC 需要，3 个 ADC 通道用于对每个电机进行电压检测
ADCA（F28002x、F280013x）、ADCB（F28003x）	PFC 电流、交流输入电压和直流输出电压	
CMPSS1、CMPSS3、CMPSS4	电机 1 和 2 三相过流故障	为每个电机提供三相电流过流故障保护功能
CMPSS2	PFC 过流和直流总线过压故障	
X-Bar	CPIO 和连接到 PWM 跳闸的 CMPSS 输出	
SCIA	与室内控制进行通信	也可以为此模块分配其他 UART 或 LIN 接口。
CPU 计时器 0	后台循环中用于电机和系统控制的虚拟计时器	也可以为此模块分配 CPU 计时器 1 或 2。

3.5 将固件迁移至新的硬件板

如果用户希望将参考设计迁移到自己的硬件板上，那么用户需要在文件 hal.c、hal.h、user_mtr1.h、user_mtr2.h 和 user_pfc.h 中相应地更改电机和 PFC 控制相关 PWM、CMPSS、ADC 外设配置、硬件参数和电机参数，如以下各节中所述。

3.5.1 配置 PWM、CMPSS 和 ADC 模块

用于控制电机和 PFC 的应用参数写为 *#define*，可根据硬件，在 *hal.h* 中配置 PWM、CMPSS 和 ADC 模块基地址。压缩机电机定义的 PWM、CMPSS、ADC 如以下代码所示。

为压缩机电机驱动器配置 PWM 和 CMPSS 基地址

```
// EPWM
#define MTR1_PWM_U_BASE          EPWM3_BASE
#define MTR1_PWM_V_BASE          EPWM5_BASE
#define MTR1_PWM_W_BASE          EPWM6_BASE

// CMPSS
#define MTR1_CMPSS_U_BASE        CMPSS1_BASE
#define MTR1_CMPSS_V_BASE        CMPSS3_BASE
#define MTR1_CMPSS_W_BASE        CMPSS3_BASE
```

为压缩机电机驱动器配置 ADC 基地址和通道

```
// ADC
#define MTR1_ADC_TRIGGER_SOC      ADC_TRIGGER_EPWM1_SOCA  // EPWM1_SOCA
#define MTR1_ADC_SAMPLE_WINDOW   14

#define MTR1_IU_ADC_BASE         ADCA_BASE               // ADCA-A11*/C0
#define MTR1_IV_ADC_BASE         ADCA_BASE               // ADCA-A5*/C2
#define MTR1_IW_ADC_BASE         ADCA_BASE               // ADCA-A0*/C15
#define MTR1_VU_ADC_BASE         ADCA_BASE               // ADCA-A7*/C3
#define MTR1_VV_ADC_BASE         ADCA_BASE               // ADCA-A12*/C1
#define MTR1_VW_ADC_BASE         ADCA_BASE               // ADCA-A1*
#define MTR1_VDC_ADC_BASE        PFC_VDC_ADC_BASE        // ADCC-A4/C14*

#define MTR1_IU_ADCRES_BASE      ADCARESULT_BASE         // ADCA-A11*/C0
#define MTR1_IV_ADCRES_BASE      ADCARESULT_BASE         // ADCA-A5*/C2
#define MTR1_IW_ADCRES_BASE      ADCARESULT_BASE         // ADCA-A0*/C15
```

```
#define MTR1_VU_ADCRES_BASE     ADCARESULT_BASE         // ADCA-A7*/C3
#define MTR1_VV_ADCRES_BASE     ADCARESULT_BASE         // ADCA-A12*/C1
#define MTR1_VW_ADCRES_BASE     ADCARESULT_BASE         // ADCA-A1*
#define MTR1_VDC_ADCRES_BASE    PFC_VDC_ADCRES_BASE     // ADCC-A4/C14*

#define MTR1_IU_ADC_CH_NUM      ADC_CH_ADCIN11          // ADCA-A11*/C0
#define MTR1_IV_ADC_CH_NUM      ADC_CH_ADCIN5           // ADCA-A5*/C2
#define MTR1_IW_ADC_CH_NUM      ADC_CH_ADCIN0           // ADCA-A0*/C15
#define MTR1_VU_ADC_CH_NUM      ADC_CH_ADCIN7           // ADCA-A7*/C3
#define MTR1_VV_ADC_CH_NUM      ADC_CH_ADCIN12          // ADCA-A12*/C1
#define MTR1_VW_ADC_CH_NUM      ADC_CH_ADCIN1           // ADCA-A1*
#define MTR1_VDC_ADC_CH_NUM     PFC_VDC_ADC_CH_NUM      // ADCC-A4/C14*

#define MTR1_IU_ADC_SOC_NUM     ADC_SOC_NUMBER0         // ADCA-A11*/C0-SOC0
#define MTR1_IV_ADC_SOC_NUM     ADC_SOC_NUMBER1         // ADCA-A5*/C2 -SOC1
#define MTR1_IW_ADC_SOC_NUM     ADC_SOC_NUMBER2         // ADCA-A0*/C15-SOC2
#define MTR1_VU_ADC_SOC_NUM     ADC_SOC_NUMBER3         // ADCA-A7*/C3 -SOC3
#define MTR1_VV_ADC_SOC_NUM     ADC_SOC_NUMBER4         // ADCA-A12*/C1-SOC4
#define MTR1_VW_ADC_SOC_NUM     ADC_SOC_NUMBER5         // ADCA-A1*    -SOC5
#define MTR1_VDC_ADC_SOC_NUM    PFC_VDC_ADC_SOC_NUM     // ADCC-A4/C14*
```

为压缩机电机驱动器控制配置外设中断

```
// Interrupt
#define MTR1_ADC_INT_BASE       ADCA_BASE               // ADCA-A1  -SOC5
#define MTR1_ADC_INT_NUM        ADC_INT_NUMBER2         // ADCA_INT2-SOC5
#define MTR1_ADC_INT_SOC        ADC_SOC_NUMBER6         // ADCA_INT2-SOC5
#define MTR1_PIE_INT_NUM        INT_ADCA2               // ADCA_INT2-SOC5
#define MTR1_CPU_INT_NUM        INTERRUPT_CPU_INT10     // ADCA_INT2-CPU_INT10
#define MTR1_INT_ACK_GROUP      INTERRUPT_ACK_GROUP10   // ADCA_INT2-CPU_INT10
```

根据硬件，在 *hal.h* 中配置 ADC 引脚与 CMPSS 模块之间的连接，有关详细信息，请参阅 TMS320F28002x 实时微控制器技术参考手册、TMS320F28003x 实时微控制器技术参考手册或 TMS320F280013x 实时微控制器技术参考手册中的表“模拟引脚和内部连接”。

```
// ADC pins connection to CMPSS
#define MTR1_IU_CMPHP_SEL       ASYSCTL_CMPHPMUX_SELECT_1    //A11*/C0, CMPSS1-HP
#define MTR1_IU_CMPLP_SEL       ASYSCTL_CMPLPMUX_SELECT_1    //A11*/C0, CMPSS1-LP, N/A

#define MTR1_IV_CMPLP_SEL       ASYSCTL_CMPLPMUX_SELECT_3    //A5*/C2,  CMPSS3-LP

#define MTR1_IW_CMPLP_SEL       ASYSCTL_CMPLPMUX_SELECT_3    //A0*/C15, CMPSS3-LP, N/A

#define MTR1_IU_CMPHP_MUX       1                            //A11*/C0, CMPSS1-HP
#define MTR1_IU_CMPLP_MUX       1                            //A11*/C0, CMPSS1-LP, N/A

#define MTR1_IV_CMPLP_MUX       1                            //A5*/C2,  CMPSS3-LP, N/A

#define MTR1_IW_CMPLP_MUX       2                            //A0*/C15, CMPSS3-LP, N/A
```

根据硬件，在 *hal.h* 中配置从 CMPSS 传递到 EPWM 和 GPIO 输出的跳闸信号，有关详细信息，请参阅 TMS320F28002x 实时微控制器技术参考手册、TMS320F28003x 实时微控制器技术参考手册或 TMS320F280013x 实时微控制器技术参考手册中的“ePWM X-BAR 多路复用器配置表”和“输出 X-BAR 多路复用器配置表”。

```
// XBARINTPUT to EPWM
#define MTR1_XBAR_TRIP_ADDRL    XBAR_O_TRIP7MUX0TO15CFG
#define MTR1_XBAR_TRIP_ADDRH    XBAR_O_TRIP7MUX16TO31CFG

#define MTR1_IU_XBAR_EPWM_MUX   XBAR_EPWM_MUX00_CMPSS1_CTRIPH       // CMPSS1-HP
#define MTR1_IV_XBAR_EPWM_MUX   XBAR_EPWM_MUX05_CMPSS3_CTRIPL       // CMPSS3-LP
#define MTR1_IW_XBAR_EPWM_MUX   XBAR_EPWM_MUX05_CMPSS3_CTRIPL       // CMPSS3-LP, N/A

#define MTR1_IU_XBAR_MUX        XBAR_MUX00          // CMPSS1-HP
#define MTR1_IV_XBAR_MUX        XBAR_MUX05          // CMPSS3-LP
#define MTR1_IW_XBAR_MUX        XBAR_MUX05          // CMPSS3-LP

#define MTR1_XBAR_INPUT         XBAR_INPUT1
#define MTR1_TZ_OSHT            EPWM_TZ_SIGNAL_OSHT1
#define MTR1_XBAR_TRIP          XBAR_TRIP7
#define MTR1_DCTRIPIN           EPWM_DC_COMBINATIONAL_TRIPIN7
```

相关的 ADC 通道用于电机电流检测哪些引脚在内部连接到比较器子系统 (CMPSS)，在 *hal.c* 文件中的 HAL_setupCMPSSs() 函数中配置 CMPSS 寄存器，代码如下所示。压缩机电机控制采用两个 CMPSS 模块，每个 CMPSS 的两个模拟比较器用于实现电机 U 相和 V 相的正负过流保护。

```
// HAL_setupCMPSSsMTR
void HAL_setupCMPSSsMTR(HAL_MTR_Handle handle)
{
    HAL_MTR_Obj *obj = (HAL_MTR_Obj *)handle;
    uint16_t cmpsaDACH;
    uint16_t cmpsaDACL;

    // Refer to the Table 9-2 in Chapter 9 of TMS320F28004x
    // Technical Reference Manual (SPRUI33B), to configure the ePWM X-Bar
    if(obj->motorNum == MTR_1)
    {
        cmpsaDACH = MTR1_CMPSS_DACH_VALUE;
        cmpsaDACL = MTR1_CMPSS_DACL_VALUE;

        ASysCtl_selectCMPHPMux(MTR1_IU_CMPHP_SEL, MTR1_IU_CMPHP_MUX);

        ASysCtl_selectCMPLPMux(MTR1_IV_CMPLP_SEL, MTR1_IV_CMPLP_MUX);

        // --------------- U-Phase ---------------------------------------------
        // Enable CMPSS and configure the negative input signal to come from the DAC
        CMPSS_enableModule(obj->cmpssHandle[0]);
}
```

CMPSS 生成的信号进入 X-Bar，在此处这些信号能够以不同且独特的方式组合，以标记来自多个来源的独特跳闸事件，从而实现故障保护。故障包括来自 CMPSS 的过流信号和电源模块的故障指示灯输出。在 *hal.c* 文件中的 HAL_setupMtrFaults() 函数中配置 XBAR 寄存器，代码如下所示。

```
void HAL_setupMtrFaults(HAL_MTR_Handle handle)
{
    HAL_MTR_Obj *obj = (HAL_MTR_Obj *)handle;
    uint16_t cnt;

    uint16_t tzSignal;
    uint16_t dcTripIn;
    if(obj->motorNum == MTR_1)
    {
    }
    else if(obj->motorNum == MTR_2)
    {
    }
}
```

在 hal.c 文件中的 HAL_setupGPIOs() 中根据硬件配置 GPIO，代码如下所示。

```
void HAL_setupGPIOs(HAL_Handle handle)
{
    // GPIO0->EPWM1A->M2/FAN_UH
    GPIO_setPinConfig(GPIO_0_EPWM1_A);
    GPIO_setDirectionMode(0, GPIO_DIR_MODE_OUT);
    GPIO_setPadConfig(0, GPIO_PIN_TYPE_STD);
    return;
}  // end of HAL_setupGPIOs() function
```

如上所述，根据硬件为风扇电机和 PFC 控制配置 PWM、ADC、CMPSS 和故障保护。PFC 使用 HAL_setupCMPSSsPFC() 和 HAL_setupPFCFaults()。

还需要根据电机和 PFC 控制使用的 CMPSS 在 hal.h 文件中的 HAL_enableMtrPWM()、HAL_enablePFCPWM()、HAL_clearMtrFaultStatus() 和 HAL_clearPFCFaultStatus() 中更改配置代码，如下面标记为**粗体**的代码所示。

```
static inline void HAL_enableMtrPWM(HAL_MTR_Handle handle)
{
    HAL_MTR_Obj *obj = (HAL_MTR_Obj *)handle;
    if(obj->motorNum == MTR_1)
    {
        // Clear any comparator digital filter output latch
```

```
        CMPSS_clearFilterLatchHigh(obj->cmpssHandle[0]);
        CMPSS_clearFilterLatchHigh(obj->cmpssHandle[1]);
        CMPSS_clearFilterLatchLow(obj->cmpssHandle[2]);
    }
    else if(obj->motorNum == MTR_2)
    {
        // Clear any comparator digital filter output latch
        CMPSS_clearFilterLatchHigh(obj->cmpssHandle[0]);
        CMPSS_clearFilterLatchLow(obj->cmpssHandle[1]);
        CMPSS_clearFilterLatchLow(obj->cmpssHandle[2]);
    }
......
    return;
} // end of HAL_enableMtrPWM() function
```

```
static inline void HAL_clearMtrFaultStatus(HAL_MTR_Handle handle)
{
    if(obj->motorNum == MTR_1)
    {
        // Clear any comparator digital filter output latch
        CMPSS_clearFilterLatchHigh(obj->cmpssHandle[0]);
        CMPSS_clearFilterLatchHigh(obj->cmpssHandle[1]);
        CMPSS_clearFilterLatchLow(obj->cmpssHandle[2]);
    }
    else if(obj->motorNum == MTR_2)
    {
        // Clear any comparator digital filter output latch
        CMPSS_clearFilterLatchHigh(obj->cmpssHandle[0]);

        CMPSS_clearFilterLatchLow(obj->cmpssHandle[1]);
        CMPSS_clearFilterLatchLow(obj->cmpssHandle[2]);
    }
......
    return;
} // end of HAL_clearMtrFaultStatus() function
```

3.5.2 设置硬件板参数

user_mtr<1/2>.h 用于存储所有用户参数以进行电机控制。模数转换器输入端的最大相电流和相电压值取决于硬件，应基于电流和电压检测以及对 ADC 输入的调节。使用的电流传感器和电压（相）传感器的数量在 user_mtr<1/2>.h 中定义，具体取决于硬件。

“*user_pfc.h*”用于存储所有用户参数以进行 PFC 控制。模数转换器输入端的最大电流、交流电压和直流电压值取决于硬件，应基于电流和电压检测以及对 ADC 输入的调节。

所有可配置参数都在 *user_mtr<1/2>.h* 和 *user_pfc.h* 文件中进行定义。可以使用 *TIDM_02010_DMPFC_Hardware Parameters Calculation.xlsx* Excel® 电子表格计算这些参数。该文件包含在 TIDM-02010 存档文件中，位于文件夹 *..\solutions\tidm_02010\docs* 下，要计算这些值，请将这些标记为***粗体***的参数复制到 *user_mtr<1/2>.h* 和 *user_pfc.h* 中，如以下代码所示。

```
//! \brief Defines the maximum voltage at the AD converter
//  Full scale voltage of AD converter, not the current voltage
#define USER_M1_ADC_FULL_SCALE_VOLTAGE_V         (441.54f)

//! \brief Defines the analog voltage filter pole location, Hz
#define USER_M1_VOLTAGE_FILTER_POLE_Hz           (448.6576819f)

//! \brief Defines the maximum current at the AD converter
// High Voltage motor control kit
#define USER_M1_ADC_FULL_SCALE_CURRENT_A         (39.6f)
```

```
//! \brief Defines the maximum voltage at the AD converter
#define USER_PFC_ADC_FULL_SCALE_DC_VOLTAGE_V    (441.54f)

//! \brief Defines the maximum voltage at the AD converter
#define USER_PFC_ADC_FULL_SCALE_AC_VOLTAGE_V    (441.54f)

//! \brief Defines the maximum current at the AD converter
#define USER_PFC_ADC_FULL_SCALE_CURRENT_A       (49.5f)
```

3.5.3 配置故障保护参数

该系统实现了故障管理，包括过流、过压、欠压、失速、过载、启动失败。故障保护参数在 *user_mtr<1/2>.h* 和 *user_pfc.h* 中定义，如以下代码所示，具体取决于硬件板、电机、PFC 和系统。

```
//! \brief motor over current threshold
#define USER_MOTOR1_OVER_CURRENT_A          (8.0)           //

//! \brief motor lost phase current threshold
#define USER_M1_LOST_PHASE_CURRENT_A        (0.2f)

//! \brief motor unbalance ratio percent threshold
#define USER_M1_UNBALANCE_RATIO             (0.2f)
```

```
//! \brief AC bus over voltage threshold
#define USER_OVER_VOLTAGE_FAULT_ACV             (280.0f)

//! \brief AC bus over voltage threshold
#define USER_OVER_VOLTAGE_NORM_ACV              (270.0f)

//! \brief AC bus under voltage threshold
#define USER_UNDER_VOLTAGE_FAULT_ACV            (90.0f)

//! \brief AC bus under voltage threshold
#define USER_UNDER_VOLTAGE_NORM_ACV             (100.0f)

//! \brief DC bus over voltage threshold
#define USER_OVER_VOLTAGE_FAULT_DCV             (410.0f)

//! \brief DC bus over voltage threshold
#define USER_OVER_VOLTAGE_NORM_DCV              (400.0f)

//! \brief DC bus under voltage threshold
#define USER_UNDER_VOLTAGE_FAULT_DCV            (15.0f)

//! \brief DC bus under voltage threshold
#define USER_UNDER_VOLTAGE_NORM_DCV             (20.0f)

//! \brief DC bus over voltage threshold
#define USER_OVER_VOLTAGE_SHUTDOWN_DCV          (420.0f)

//! \brief DC bus shut down voltage threshold
#define USER_SHUTDOWN_VOLTAGE_DCV               (420.0f)
```

3.5.4 设置电机电气参数

user_mtr<1/2>.h 中提供的用于 PMSM/BLDC 电机的参数如以下代码所示。如果在该电机上实施 FAST 技术或从电机数据表中获取，则可以识别电机参数。

```
#define USER_MOTOR1_TYPE                    MOTOR_TYPE_PM
#define USER_MOTOR1_NUM_POLE_PAIRS          (4)
#define USER_MOTOR1_Rr_Ohm                  (0.0)
#define USER_MOTOR1_Rs_Ohm                  (2.62655902f)
#define USER_MOTOR1_Ls_d_H                  (0.00860825367f)
#define USER_MOTOR1_Ls_q_H                  (0.00860825367f)
#define USER_MOTOR1_RATED_FLUX_VpHz         (0.377903223f)
```

3.5.5 设置 PFC 控制参数

pfc_ctrl.h 中提供的用于 PFC 控制的参数如以下代码所示。

```
#define CNTL_2p2z_B0_1  0.4104341549f
#define CNTL_2p2z_B1_1 -0.3822445616f
#define CNTL_2p2z_B2_1  0.0712511235f
#define CNTL_2p2z_A1_1 -1.1159959670f
#define CNTL_2p2z_A2_1  0.1159959670f
```

4 设计和文档支持

4.1 设计文件

4.1.1 原理图

要下载原理图，请参阅 TIDM-02010 中的设计文件。

4.1.2 物料清单

要下载物料清单 (BOM)，请参阅 TIDM-02010 中的设计文件。

4.1.3 Altium 工程

要下载 Altium 工程文件，请参阅 TIDM-02010 中的设计文件。

4.1.4 Gerber 文件

要下载光绘文件，请参阅 TIDM-02010 中的设计文件。

4.1.5 PCB 布局指南

- 考虑到应用的成本敏感性，该参考设计使用具有两层 1oz 铜并采用单侧 SMD 元件放置方式的 PCB 进行实施。设计 PCB 时需要牢记几个重要方面。下面对每个块的系统级放置方式和布局进行了说明。
- 大功率路径中的元件处于 PCB 的外边缘，并尽可能互相靠近。微控制器放置在中心，以便与所有需要控制的电源块保持最佳距离。引脚进行了合理分配，以尽可能减小控制/反馈信号引线距离并尽可能减少模拟信号和数字信号之间的交叉。
- 交流线路保护和 EMI 滤波器
  - 所有交流线路保护元件紧密地放置在一起，尽可能缩短连接路径。在保护和 EMI 滤波器电路周围提供了接地连接保护。
  - 有源 EMI 滤波器放置在最佳距离处，以便更接近开关并以最小距离连接到 EARTH 端子。
- IPFC 驱动器

  在 IPFC 驱动器中，三个电流路径对于 PCB 布局非常关键 - 大功率交流环路、直流环路和栅极驱动环路。这些路径需要尽可能短且具有最大宽度，以降低寄生环路电感。

  - 交流环路 - 由二极管桥（源）、电感器和 MOSFET 漏极和 MOSFET 源极（回路）组成。在该环路上，主要是通过电感器、MOSFET 漏极和二极管阳极之间的连接来处理高频和大功率。连接该节点时特别小心，通过减小距离和增加铜面积来更大限度地降低寄生电感。
  - 直流回路 - 由二极管桥（源）、电感器、二极管、电容器、负载（回路）组成。为了均匀分布 rms 电流应力，电解电容器组应放置在最佳位置，使每个电解电容器与二极管阴极的电气距离大致保持相同。该设计使用铜平面进行 V_{DC} 和 PGND 连接。为了抑制高频分量，金属膜电容器放置在二极管的阴极旁边。它可以显著降低环路电感。
  - 栅极驱动环路 – 由驱动器电源（源）、栅极驱动器 IC、MOSFET 栅极和 MOSFET 源极引脚（回路）组成。该设计针对 IPFC 的两相使用并联布置，以尽可能减小其他两个交流/直流环路。由于该并联布置，栅极驱动器无法访问外部相位 MOSFET 栅极。已使用 SMD 绝缘粗跳线将栅极驱动器信号连接至 MOSFET 栅极。
- 压缩机和风扇驱动器
  - 由于对纹波的要求非常高，压缩机驱动器放置在最靠近 IPFC 驱动器的直流总线电容器组的位置，而风扇放置在压缩机旁边。
  - 实施了采用 4 线检测的低侧分流电阻器方法来进行电流检测。已使用具有阻抗匹配电阻器的差分对将来自分流电阻器的检测信号连接至 OPAMP 电路。分流电阻器已放置在模块附近，并直接连接接地铜平面。
- 辅助电源
  - 由于对电源和纹波的要求非常低，辅助电源放置在风扇驱动器之后。使用专用铜平面将 APS 接地端连接至直流总线电容器组。该布置可更大限度地减少高频/大功率电机电流与控制电路之间的干扰。

4.2 软件文件

要下载 Code Composer Studio 集成开发环境，请访问 CCSTUDIO。

要下载特定于 TIDM-02010 硬件的软件设计文件，请参阅 C2000WARE-MOTORCONTROL-SDK 中的设计文件。

4.3 文档支持

1. 德州仪器 (TI)，*TMS320F28002x 实时微控制器* 数据表。
2. 德州仪器 (TI)，*TMS320F28002x 实时微控制器* 技术参考手册。
3. 德州仪器 (TI)，*TMS320F28003x 微控制器* 数据表。
4. 德州仪器 (TI)，*TMS320F28003x 实时微控制器* 技术参考手册。
5. 德州仪器 (TI)，*TMS320F280013x 微控制器* 数据表。
6. 德州仪器 (TI)，*TMS320F280013x 实时微控制器* 技术参考手册。
7. 德州仪器 (TI)，*InstaSPIN-FOC 和 InstaSPIN-MOTION* 用户指南。
8. 德州仪器 (TI)，*MotorControl SDK 通用工程和实验* 用户指南。
9. 德州仪器 (TI)，*C2000™ 软件频率响应分析器 (SFRA) 库和补偿设计器* 用户指南。
10. 德州仪器 (TI)，*具有功率计量测试结果的两相交错式 PFC 转换器* 参考设计。
11. 德州仪器 (TI)，*使用单一直流链路分流器的 PMSM 无传感器 FOC* 应用手册。
12. 德州仪器 (TI)，*TIDM-1022 谷底开关升压功率因数校正 (PFC)* 参考设计。
13. 德州仪器 (TI)，*C2000 SysConfig* 应用手册。

4.4 支持资源

TI E2E™ 支持论坛是工程师的重要参考资料，可直接从专家获得快速、经过验证的解答和设计帮助。搜索现有解答或提出自己的问题可获得所需的快速设计帮助。

链接的内容由各个贡献者“按原样”提供。这些内容并不构成 TI 技术规范，并且不一定反映 TI 的观点；请参阅 TI 的《使用条款》。

4.5 商标

5 术语

SLYZ022	TI 术语表：本术语表列出并解释了术语、首字母缩略词和定义
PMSM	永久磁性同步电机
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
PFC	功率因数校正
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

6 修订历史记录

Changes from Revision * (January 2022) to Revision A (October 2022)	Page
• 更新了整个文档中的表格、图和交叉参考的编号格式	1
• 更新了方框图	5
• 添加了 TMS320F2800137 主题	5
• 从 *v4.00.0.00* 更新至 *v4.01.00.00*	40
• 将版本 11 更新至版本 12	40
• 添加了注释	40
• 在整个出版物中添加了 *F280013x*	40
• 更新了方框图	55
• 更新了方框图	59
• 更新了文档支持	85

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出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