Design Guide: TIDM-02012

混合动力汽车和电动汽车 HVAC eCompressor 高压电机控制参考设计

说明

TIDM-02012 是一款高压 5kW 参考设计，专为由中等性能 C2000™ F28003x 实时 MCU 控制的混合动力汽车 (HEV) 和电动汽车 (EV) eCompressor 应用而构建。此参考设计旨在使用 400V 和 800V 直流总线进行评估，适应电池电压更高的市场趋势。基于 controlCARD 的设计使用户能够评估多个 MCU 选项，还可进行扩展以支持 C2000™ 产品系列中的其他器件（包括未来根据路线图开发的器件），从而满足不断增长的网络安全、功能安全和其他汽车市场需求。

资源

TIDM-02012	设计文件夹
TMS320F280039C	产品文件夹
UCC21530-Q1、OPA607-Q1	产品文件夹
LM25184-Q1、TCAN1044A-Q1	产品文件夹
C2000WARE-MOTORCONTROL-SDK	工具文件夹

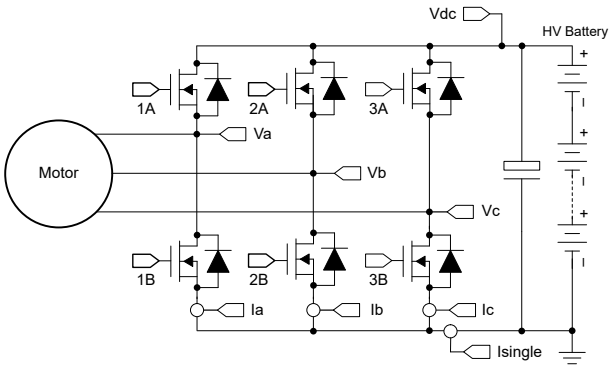
请咨询我司 TI E2E™ 支持专家

特性

- 无传感器 FOC 使用在成本优化型 C2000 实时微控制器上运行的 InstaSPIN FAST 观测器。
- 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。
- 增量软件构建可分步验证不同的软件模块，还可验证支持弱磁控制、MTPA、过调制和振动补偿的可选软件功能。
- 宽速度范围运行，从 5Hz 开始。
- 支持 CAN FD 和 LIN 接口，使用片上比较器实现电机过流保护。

应用

- 汽车 HVAC 压缩机



English Document: TIDUF17

1 系统说明

数十年来，内燃机（ICE）一直在为汽车以及加热和冷却系统提供动力。随着汽车行业电气化并过渡到具有小型内燃机的混合动力汽车 (HEV) 或完全没有发动机的纯电动汽车 (EV)，暖通空调 (HVAC) 系统将使用 PMSM 电机从高压电源驱动 eCompressor。eCompressor 可用于支持冷却（空调）、加热（热泵），并进行动力总成中其他系统（如电池包和牵引驱动器）的热管理。

如今，HEV/EV 中的 eCompressor 必须满足不断增加的需求，包括低成本、更小尺寸、更少振动和噪声、更高功率级别和更高能效。此参考设计演示了如何在不使用位置传感器的情况下使用磁场定向控制 (FOC) 来控制 eCompressor 电机。整个系统可帮助用户减少物料清单中关键组件的数量，提高效率，减少振动和噪声，并节省开发时间。此参考设计基于 TMS320F28003x 实时控制器系列，可扩展为基于 controlCARD 外形的产品系列中的未来 MCU。

支持包括 CAN FD 和 LIN 在内的汽车通用通信接口，以简化客户评估过程。支持多分流器和单分流器电流检测模式，以供具有不同设计目标的客户使用此设计进行评估。选择隔离式栅极驱动器以支持 400V 和 800V 直流总线电压，因为行业趋势是朝着更高的电压电平发展。

WARNING

TI 建议，该参考设计仅可在实验室环境中运行，不应将此参考设计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。

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 所示为该参考设计的方框图，并突出显示了主要 TI 元件。

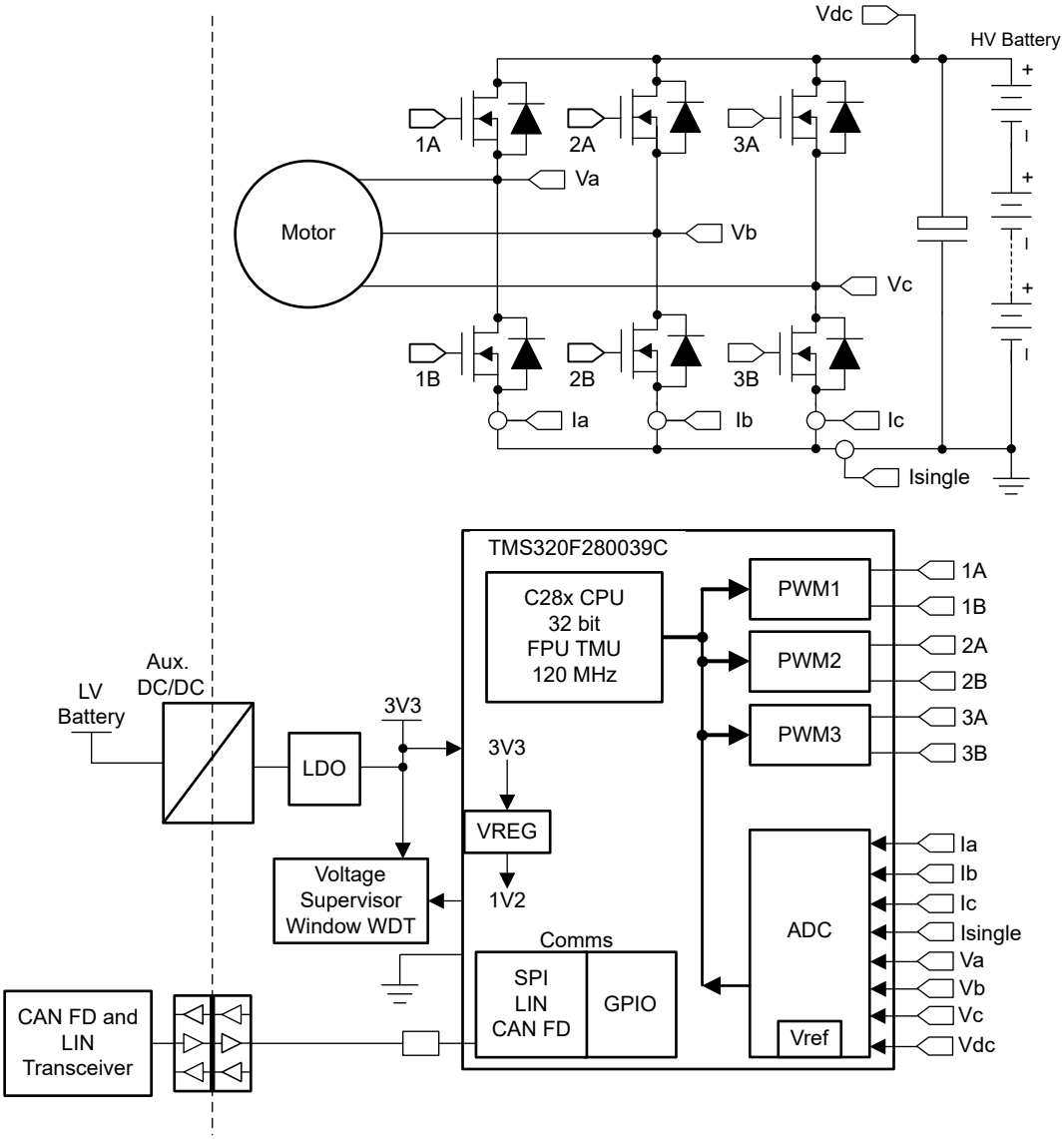


图 2-1. TIDM-02012 eCompressor 方框图

2.2 设计注意事项

此设计使用中等性能 C2000™ F28003x 实时 MCU 驱动 HEV 和 EV eCompressor。具有高抗噪性能的电流和电压检测流程对于精确的电机驱动而言是必不可少的。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 TIDM-02012。

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 TMS320F280039C

TMS320F28003x 是 C2000 实时微控制器系列中的一款器件，该系列可扩展、超低延迟器件旨在提高电力电子设备的效率，包括但不限于：高功率密度、高开关频率，并支持使用 GaN 和 SiC 技术。*使用 C2000™ 实时微控制器的基本开发指南* 应用手册基于 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 是一款与 C28x CPU 并行执行的独立 32 位浮点数学加速器，支持分担大量常见任务。

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。

2.3.2 UCC21530-Q1

UCC21530-Q1 是一款隔离式双通道栅极驱动器，具有 4A 峰值拉电流和 6A 峰值灌电流。该栅极驱动器的设计可驱动高达 5MHz 的 IGBT、Si MOSFET 和 SiC MOSFET，具有出色的传播延迟和脉宽失真。

输入侧通过 5.7kV_{RMS} 增强型隔离层与两个输出驱动器隔离，其共模瞬态抗扰度 (CMTI)至少为 100V/ns 。两个次级侧驱动器之间的内部功能隔离支持高达 1850V 的工作电压。

该驱动器可配置为两个低侧驱动器、两个高侧驱动器或一个死区时间 (DT) 可编程的半桥驱动器。EN 引脚拉至低电平时会同时关闭两个输出，悬空或拉高时可使器件恢复正常运行。作为一种失效防护机制，初级侧逻辑故障会强制两个输出为低电平。

2.3.3 OPA607-Q1

OPA607-Q1 器件是一款解补偿通用 CMOS 运算放大器，最小稳定增益为 6V/V，具有 3.8nV/√Hz 的低噪声和 50MHz 的 GBW。OPAx607-Q1 器件具有低电压温漂 (dVOS/dT) 和高带宽特性，因此非常适合低成本通用应用，如低侧电流检测和 TIA（跨阻放大器）。高阻抗 CMOS 输入使得 OPAx607-Q1 放大器非常适合连接具有高输出阻抗的传感器（例如压电式传感器）。

2.3.4 LM25184-Q1

LM25184-Q1 是一款初级侧调节 (PSR) 反激式转换器，在 4.5V 至 42V 的宽输入电压范围内具有高效率，可通过初级侧反激式电压对隔离输出电压采样。高集成度可实现简单可靠的高密度设计，其中只有一个元件穿过隔离层。通过采用边界导电模式 (BCM) 开关，可实现紧凑的磁解决方案以及优于 ±1.5% 的负载和线路调节性能。集成的 65V 功率 MOSFET 能够提供高达 15W 的输出功率并提高应对线路瞬变的余量。

LM25184-Q1 简化了隔离式直流/直流电源的实现，且可通过可选功能优化目标终端设备的性能。该器件通过一个电阻器来设置输出电压，同时使用可选的电阻器通过抵消反激式二极管的压降热系数来提高输出电压精度。其他功能包括内部固定或外部可编程软启动、用于可调节线路 UVLO 的精密使能输入（带迟滞功能）、间断模式过载保护和带自动恢复功能的热关断保护。

2.3.5 TCAN1044A-Q1

TCAN1044A-Q1 是一款高速控制器局域网 (CAN) 收发器，符合 ISO 11898-2:2016 高速 CAN 规范的物理层要求。此类收发器具有经过认证的电磁兼容性 (EMC)，是数据速率高达 5 兆位/秒 (Mbps) 的传统 CAN 和 CAN FD 网络的理想选择。这些器件可以在更简单的网络中实现高达 8Mbps 的运行速度。TCAN1044AV-Q1 包括通过 V_{IO} 引脚实现的内部逻辑电平转换功能，允许将收发器 I/O 直接连接到 1.8V、2.5V、3.3V 或 5V 逻辑电平。该收发器支持低功耗待机模式，并且可通过符合 ISO 11898-2:2016 所定义唤醒模式 (WUP) 的 CAN 来唤醒。

2.4 系统设计原理

2.4.1 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 2-2 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 90º 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。

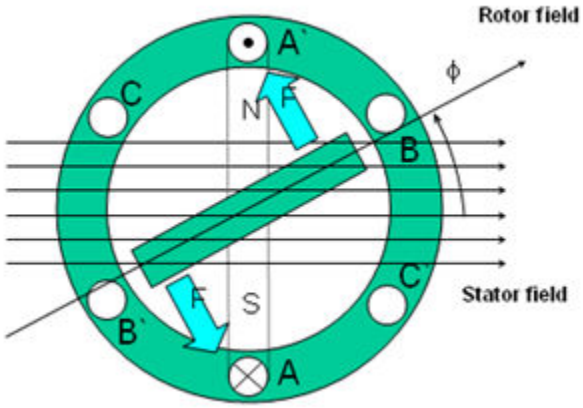


图 2-2. 旋转的定子磁通和转子磁通之间的相互作用产生扭矩

2.4.2 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 2-3 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。

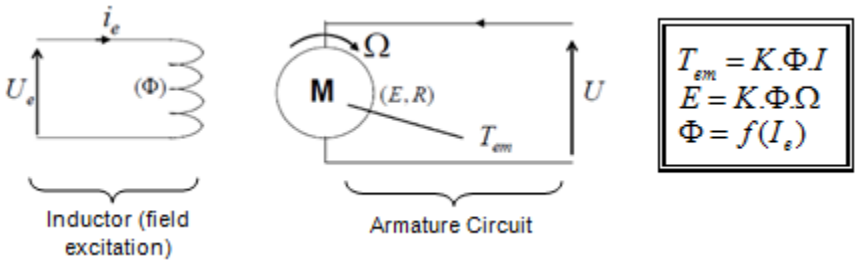


图 2-3. 在直流电机模型中磁通和扭矩是独立控制的

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 1 所示。

$$\tau_{em} = \vec{B}_{stator} \times \vec{B}_{rotor} \tag{1}$$

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 2 所示。

$$\tau_{em} \propto \psi_R \times i_{sq} \tag{2}$$

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 3 所示。

$$\bar{i}_s = i_a + \alpha i_b + \alpha^2 i_c \tag{3}$$

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 2-4 所示为定子电流的复数空间矢量。

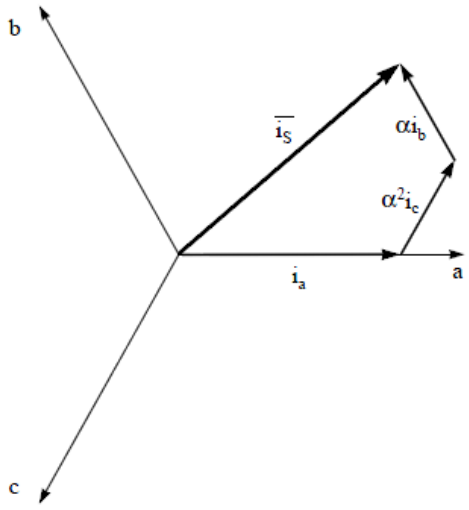


图 2-4. 定子电流空间矢量及其以 (a,b,c) 坐标系表示的分量

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 2-5 所示的矢量图。

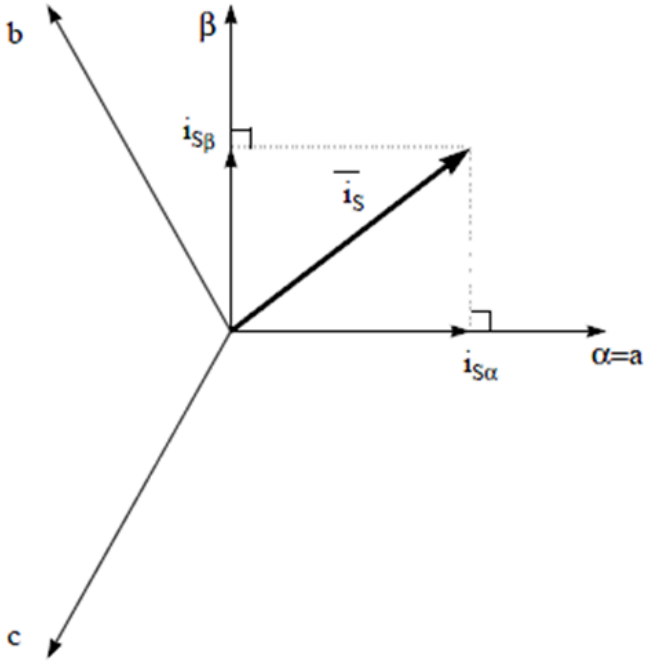


图 2-5. 静止坐标系中的定子电流空间矢量

将三相系统修改为 (α，β) 二维正交系统的投影如方程式 4 所示。

$$i_{s\alpha} = i_a$$
$$i_{s\beta} = \frac{1}{\sqrt{3}} i_a + \frac{2}{\sqrt{3}} i_b \quad (4)$$

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 2-6 显示了来自该二维坐标系的电流矢量的关系。

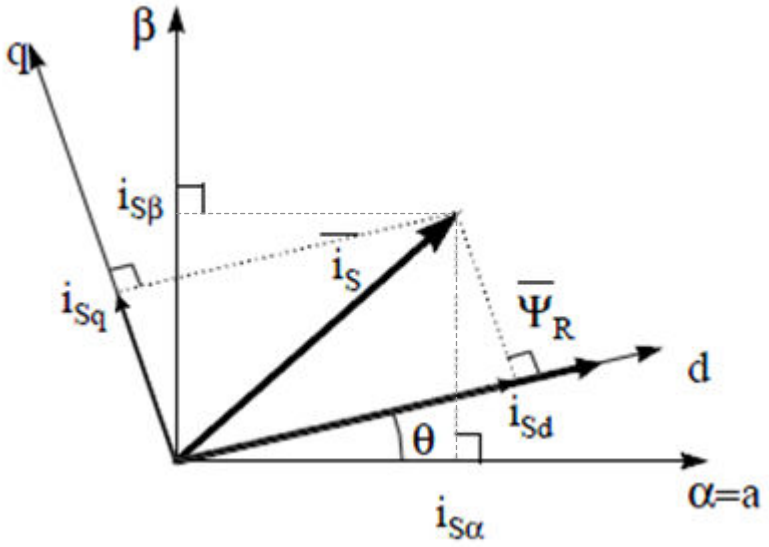


图 2-6. d,q 旋转坐标系中的定子电流空间矢量

电流矢量的磁通和扭矩分量由方程式 5 决定。

$$i_{sd} = i_{s\alpha}\cos(\theta) + i_{s\beta}\sin(\theta)$$
$$i_{sq} = -i_{s\alpha}\sin(\theta) + i_{s\beta}\cos(\theta) \quad (5)$$

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 2-7 总结了使用 FOC 进行扭矩控制的基本配置方案：

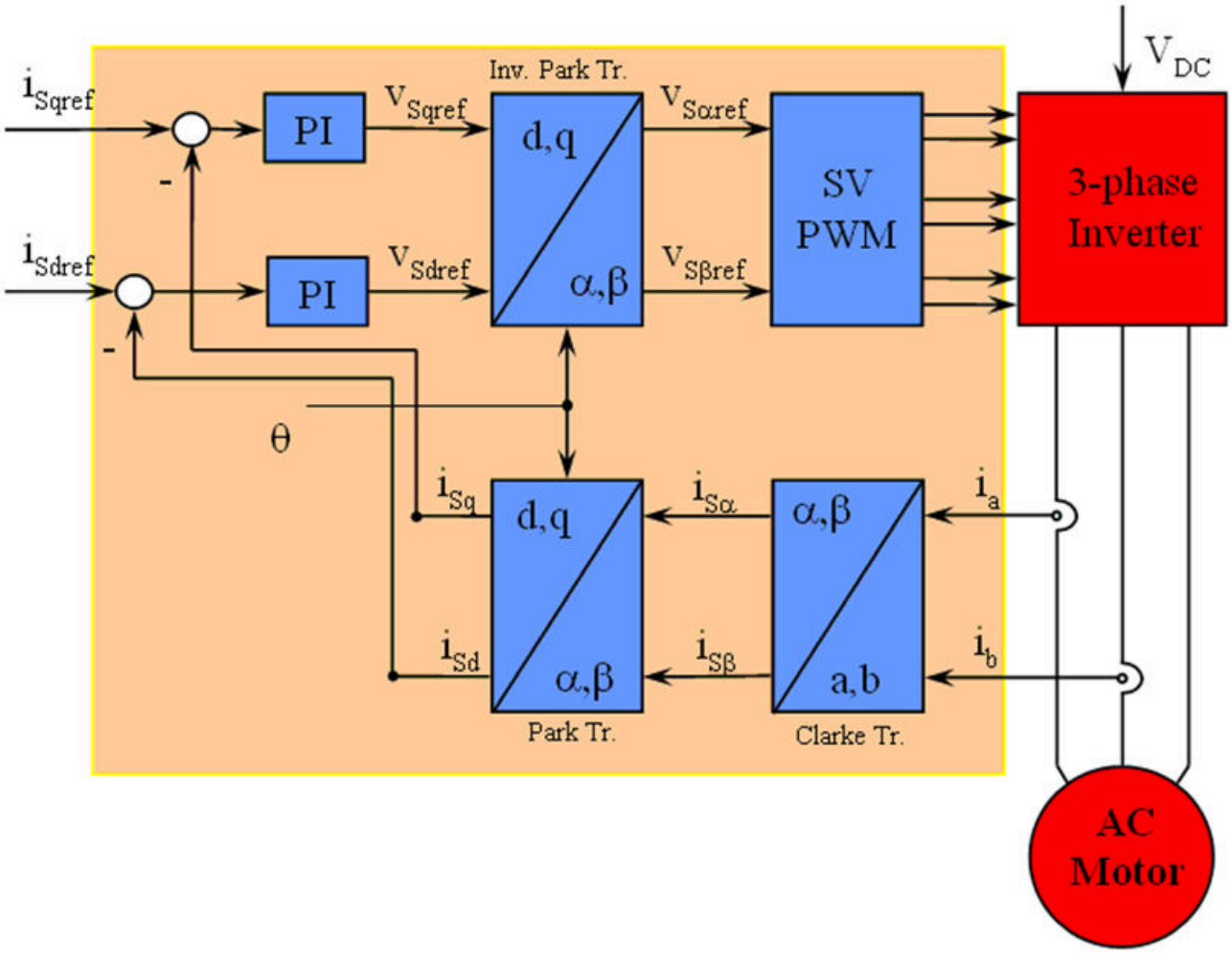


图 2-7. 交流电机 FOC 基本配置方案

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 2-8 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。

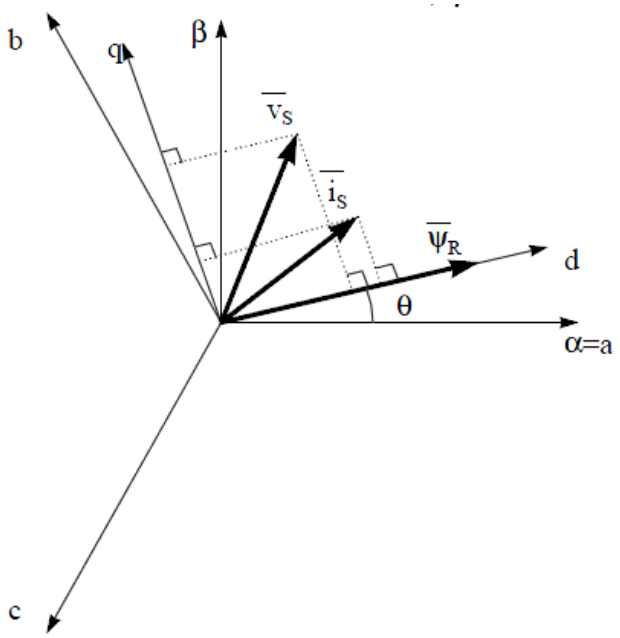


图 2-8. (d, q) 旋转坐标系中的电流、电压和转子磁通空间矢量

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，这个系统的关键模块是使用 InstaSPIN-FOC FAST™ 估算器来估算转子位置。

图 2-9 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。

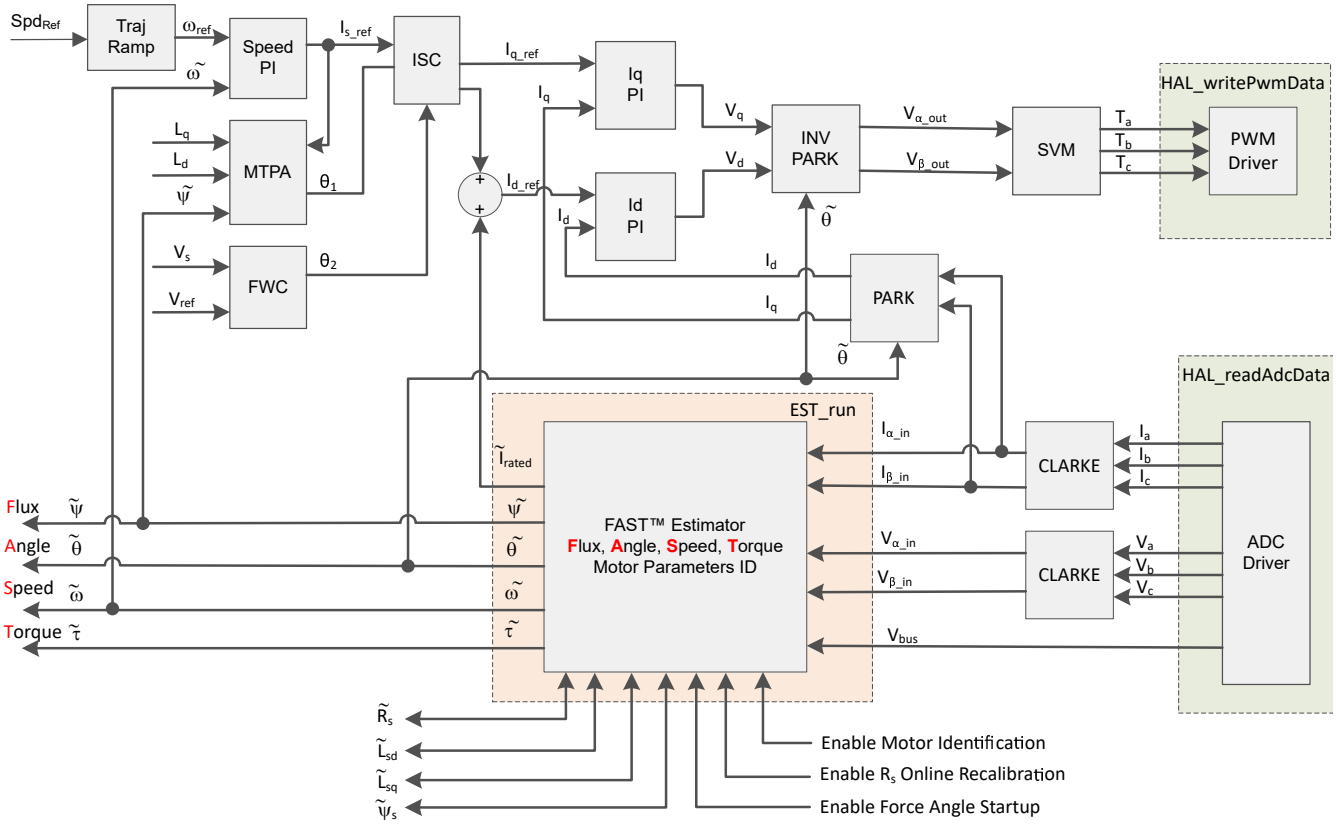


图 2-9. 使用 FAST 并具有 FWC 和 MTPA 功能的压缩机 PMSM 的无传感器 FOC

2.4.3 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 6 和方程式 7 所示。

$$v_d = L_d\frac{di_d}{dt} + R_s i_d - p\omega_m L_q i_q \tag{6}$$

$$v_q = L_q\frac{di_q}{dt} + R_s i_q + p\omega_m L_d i_d + p\omega_m \psi_m \tag{7}$$

图 2-10 显示了 IPM 同步电机的动态等效电路。

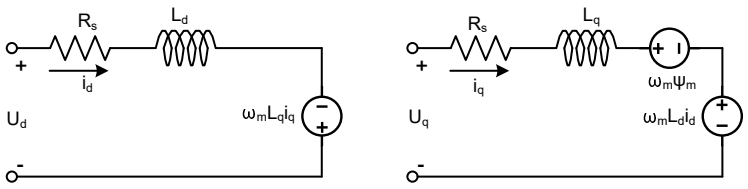


图 2-10. IPM 同步电机的等效电路

IPMSM 产生的总电磁扭矩可以由方程式 8 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。

$$T_e = \frac{3}{2}p\left[\psi_m i_q + \left(L_d - L_q\right) i_d i_q\right] \quad (8)$$

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 9 和方程式 10 表示。

$$I_a = \sqrt{i_d^2 + i_q^2} \le I_{max} \quad (9)$$

$$V_a = \sqrt{v_d^2 + v_q^2} \le V_{max} \quad (10)$$

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 11 中所示的值。

$$\sqrt{v_d^2 + v_q^2} \le v_{max} = \frac{v_{dc}}{\sqrt{3}} \quad (11)$$

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 12，如下所示。

$$\sqrt{L_d^2\left(i_d + \frac{\psi_{pm}}{L_d}\right)^2 + L_q^2 i_q^2} \le \frac{V_{max}}{\omega_m} \quad (12)$$

方程式 11 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 12 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 2-11 所示。

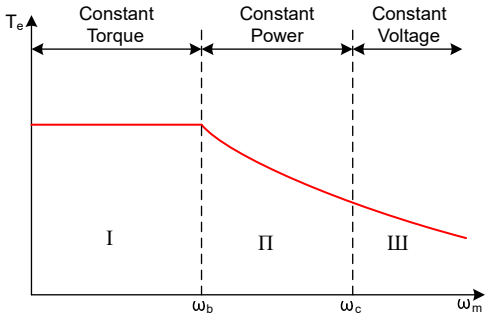


图 2-11. IPMSM 控制工作区域

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 8，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。

$$I_s = \sqrt{i_d^2 + i_q^2} \quad (13)$$

$$I_d = I_s \cos\beta \quad (14)$$

$$I_q = I_s \sin\beta \tag{15}$$

其中 β 是同步 (d-q) 坐标系中的定子电流角度。方程式 8 可以表示为方程式 16，其中 I_s 替换了 i_d和 i_q。

方程式 16 表明电机扭矩取决于定子电流矢量的角度，因此

$$T_e = \frac{3}{2} p I_s \sin\beta \left[\psi_m + (L_d - L_q) I_s \cos\beta \right] \tag{16}$$

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 17 所示）时，可以找到 MTPA 点。

$$\frac{dT_e}{d\beta} = \frac{3}{2} p \left[\psi_m I_s \cos\beta + (L_d - L_q) I_s^2 \cos 2\beta \right] = 0 \tag{17}$$

接下来，可以通过方程式 18 得出 MTPA 控制的电流角度。

$$\beta_{mtpa} = \cos^{-1} \frac{-\psi_m + \sqrt{\psi_m^2 + 8*(L_d - L_q)^2 * I_s^2}}{4*(L_d - L_q)*I_s} \tag{18}$$

因此，可以使用 MTPA 控制的电流角度通过方程式 19 和方程式 20 来表示有效的 d 轴和 q 轴基准电流。

$$I_d = I_s * \cos\beta_{mtpa} \tag{19}$$

$$I_q = I_s * \sin\beta_{mtpa} \tag{20}$$

不过，如方程式 18 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 21 和方程式 22 所示。

$$L_d = f_1(i_d, i_q) = f_1(i_d) \tag{21}$$

$$L_q = f_2(i_q, i_d) = f_2(i_q) \tag{22}$$

为了通过简化方程式 18 来减轻 ISR 计算负担，基于电机参数的常数 β_{mtpa} 改为用方程式 24 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。

$$K_{mtpa} = \frac{\psi_m}{4*(L_q - L_d)} = 0.25 * \frac{\psi_m}{(L_q - L_d)} \tag{23}$$

$$\beta_{mtpa} = cos^{-1}\left(K_{mtpa}/I_s - \sqrt{(K_{mtpa}/I_s)^2 + 0.5} \right) \tag{24}$$

第二个中间变量 G_{mtpa} 由方程式 25 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 26 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}。

$$G_{mtpa} = K_{mtpa}/I_s \tag{25}$$

$$\beta_{mtpa} = cos^{-1}\left(G_{mtpa} - \sqrt{G_{mtpa}^2 + 0.5} \right) \tag{26}$$

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来减弱磁通量以扩大可达到的转速范围。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 9 和方程式 10 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 12 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 2-12 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。

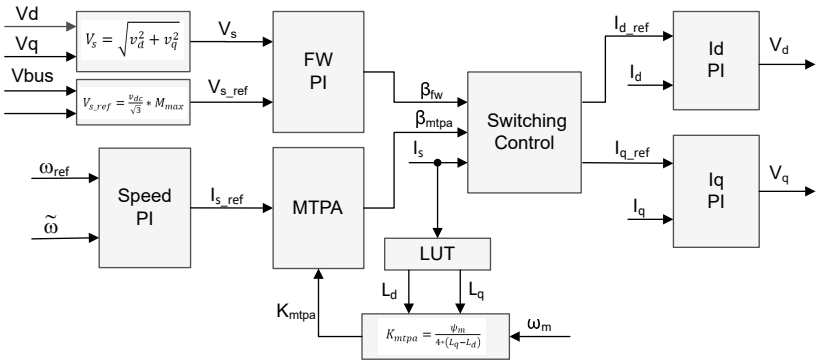


图 2-12. 弱磁和每安培最大扭矩控制的方框图

使用 FAST 并具有 FWC 和 MTPA 功能的压缩机 PMSM 的无传感器 FOC 显示了基于 FAST 的 FOC 的实现方框图。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 2-13 所示。

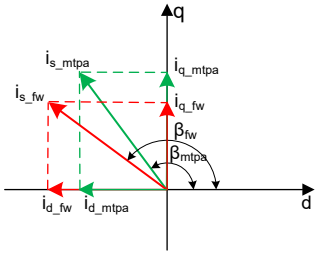


图 2-13. FW 和 MTPA 期间 IPMSM 的电流相量图

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 14 和方程式 15 所示。可以根据下面的方程式 27 和方程式 28 来选择电流角度。

$$\beta = \beta_{fW}\, if\, \beta_{fW} > \beta_{mtpa} \tag{27}$$

$$\beta = \beta_{mpta}\, if\, \beta_{fW} < \beta_{mtpa} \tag{28}$$

图 2-14 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。

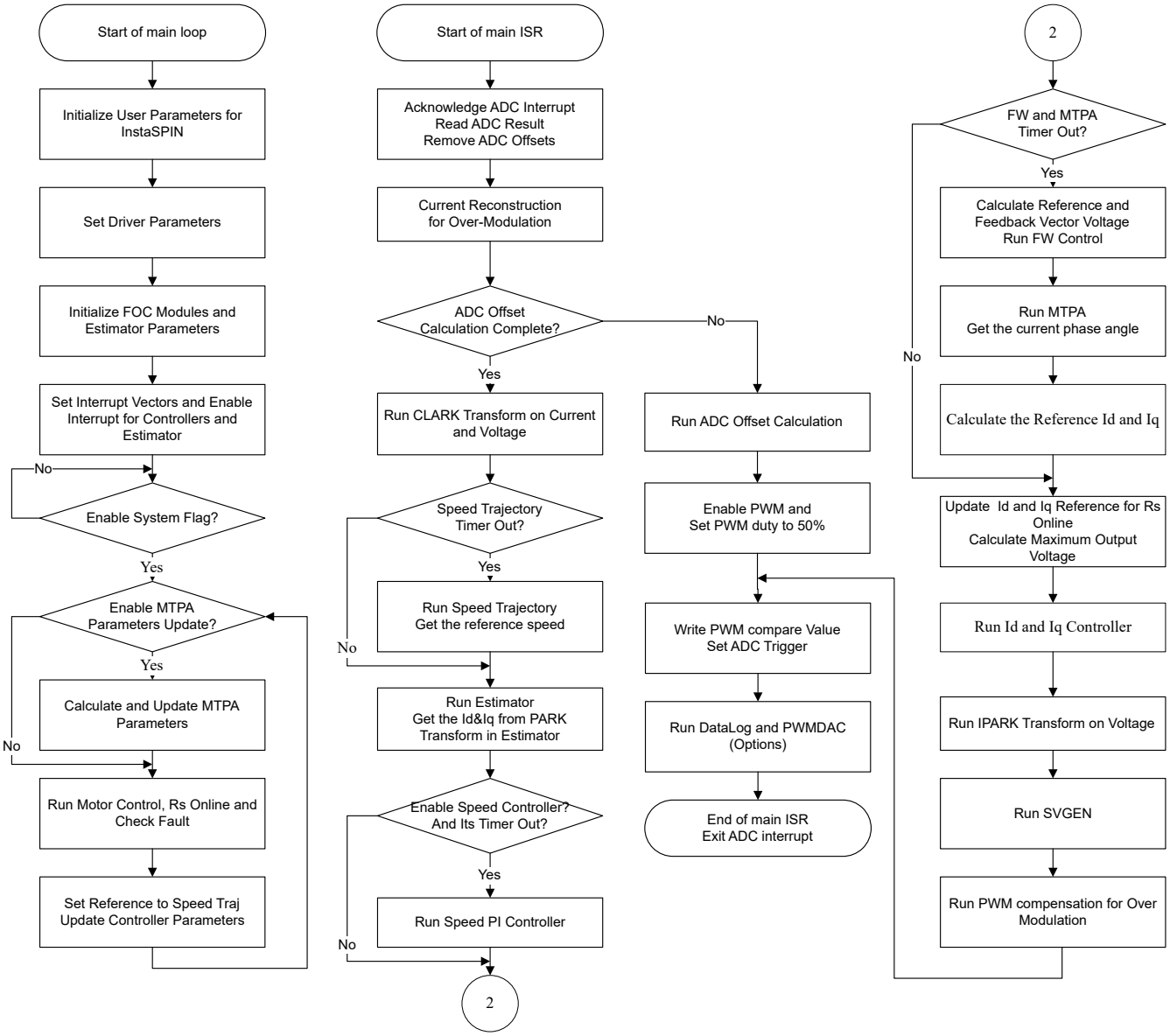


图 2-14. 采用 FW 和 MTPA 的 InstaSPIN-FOC 工程的流程图

2.4.4 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 2-15 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。

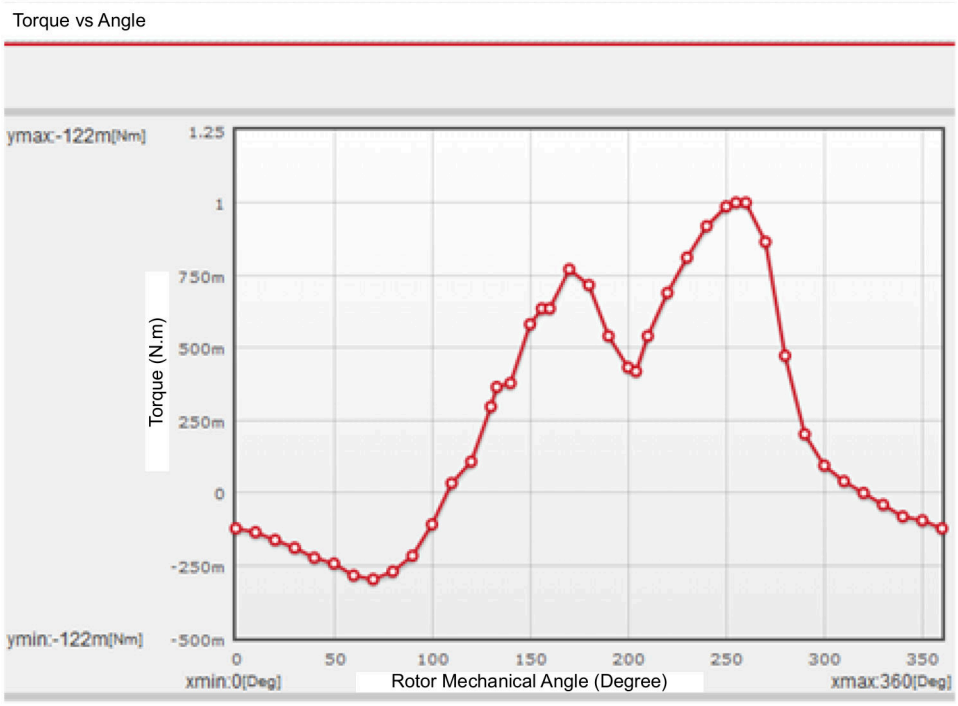


图 2-15. 负载扭矩波形

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 2-16 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）。

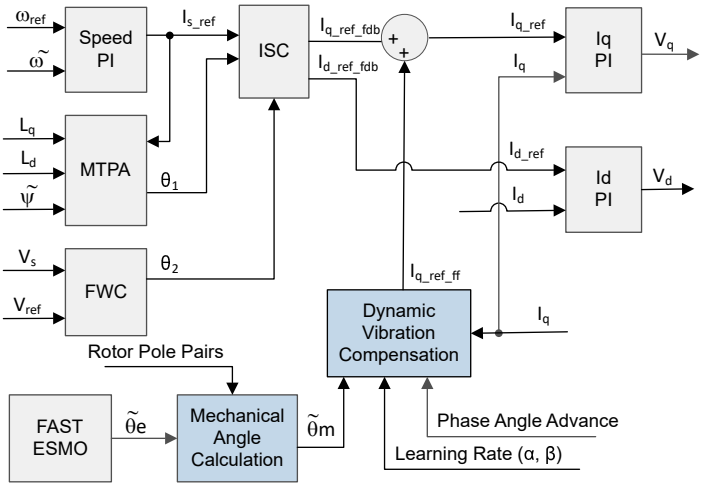


图 2-16. 具有振动补偿功能、基于 FOC 的电机驱动器方框图

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 2-16 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。

2.4.5 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电流标度值、电压标度值和电压滤波器极点，供 FAST 观测器进行 eCompressor 电机控制。

2.4.5.1 电机电流反馈

在硬件设计中支持用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

当前版本的软件仅支持三分流器电流检测方法。计划在未来的版本中支持单分流器电流检测方法。

2.4.5.1.1 采用三分流器的电流检测

在每个 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。该失调基准电压由电压跟随器生成，如图 2-17 所示。该 1.65V 基准电压用于三分流器和单分流器交流电压反馈检测电路。

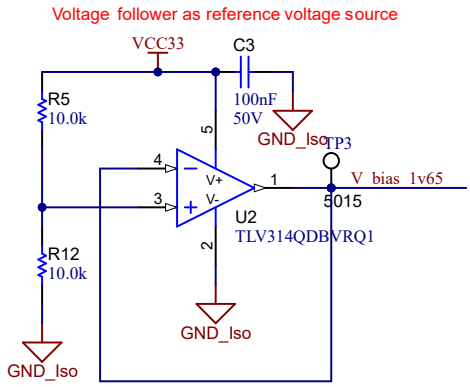


图 2-17. 3.3V 输入电路提供的 1.65V 基准

图 2-18 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。此电路的传输函数由 方程式 29 给定。

$$V_{OUT} = V_{OFFSET} + (I_{IN} \times R_{SHUNT} \times G_i) \tag{29}$$

其中 R_{shunt} = 0.005 (Ω) 和 V_{offset} = 1.65 (V)

利用计算出的电阻值，可得到图 2-18 所示的检测电路，G_i 由方程式 30 给出。

$$G_i = \frac{R_{fb}}{R_{in}} = \frac{R29}{R31} = \frac{10K}{1K} = 10 \tag{30}$$

微控制器可测量的最大峰峰值电流由方程式 31 给出。

$$I_{scale\_max} = \frac{V_{ADC\_max}}{R_{shunt} \times G_i} = \frac{3.3}{0.005 \times 10} = 66A \tag{31}$$

即峰峰值为 ±33A。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         (66.00f)
```

正确的电流反馈极性也很重要，因为这样才能确保微处理器精确测量电流。在该硬件电路板配置中，分流电阻器的负引脚接地，同时与运算放大器的同相引脚连接。

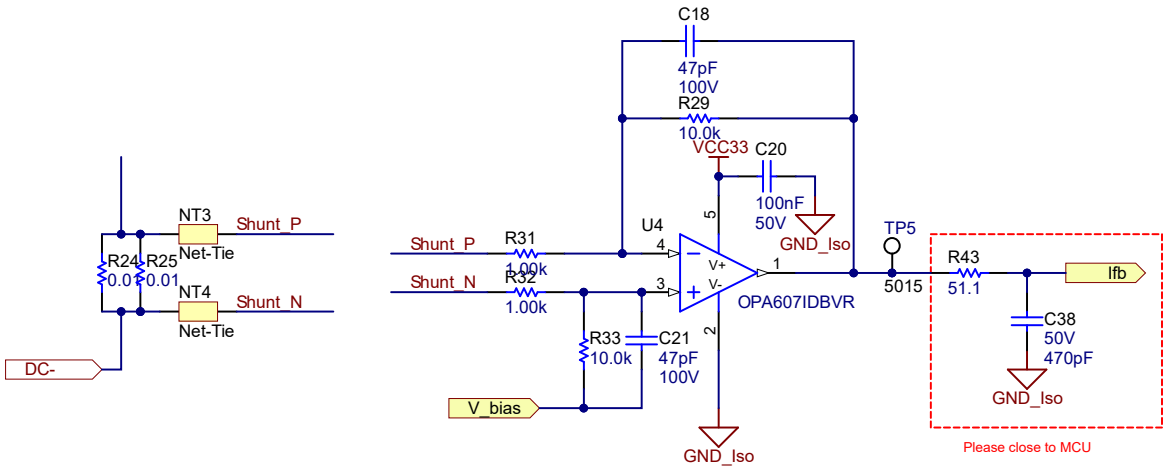


图 2-18. 采用三分流器的电机电流检测

2.4.5.1.2 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* 中详细介绍了单分流器技术。

在该参考电路板上，通过独立的分流电阻器和放大器电路来实现单分流电流检测技术，如图 2-19 所示。

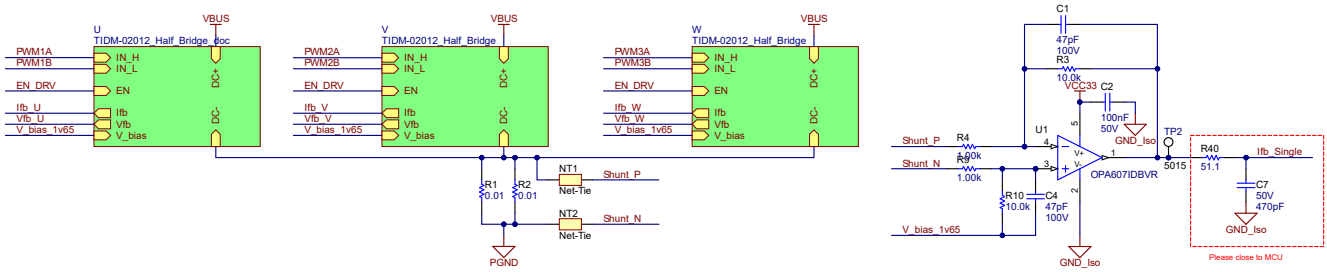


图 2-19. 采用单分流器的电机电流检测电路

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。

2.4.5.2 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。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 2-20 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压缩机电机和直流总线。

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 32 进行计算。

$$V_{FS} = V_{ADC\_FS} \times G_v = 3.3V \times 293\text{。}955 = 970.05V \tag{32}$$

其中 G_v 是衰减因子，可通过以下公式进行计算

$$G_V = \frac{(R26 + R27 + R28 + R30)}{R30} = \frac{(499K + 499K + 499K + 5.11K)}{5.11K} = 293\text{。}955 \tag{33}$$

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          (970.05f)
```

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 34 进行计算：

$$f_{filter\_pole} = \frac{1}{(2 \times \pi \times R_{Parallel} \times C)} = 664\text{。}94 \quad Hz \tag{34}$$

$Where,$
$C = 47nF$

$$R_{Parallel} = \left(\frac{(499K + 499K + 499K) \times 5.11K}{(499K + 499K + 499K) + 5.11K}\right) = 5.0926k\Omega$$

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            (664.94f)
```

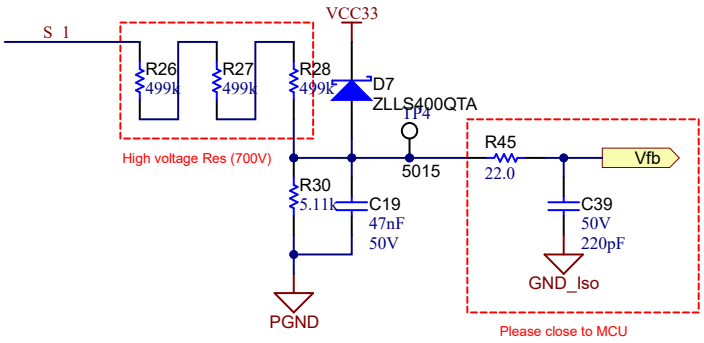


图 2-20. 电机电压检测电路

3 硬件、软件、测试要求和测试结果

3.1 硬件要求

本部分将详细介绍硬件，并说明电路板上的不同区域，以及如何为本设计指南所述的实验设置这些区域。

3.1.1 硬件板概述

图 3-1 显示了典型 eCompressor 驱动系统的概述。

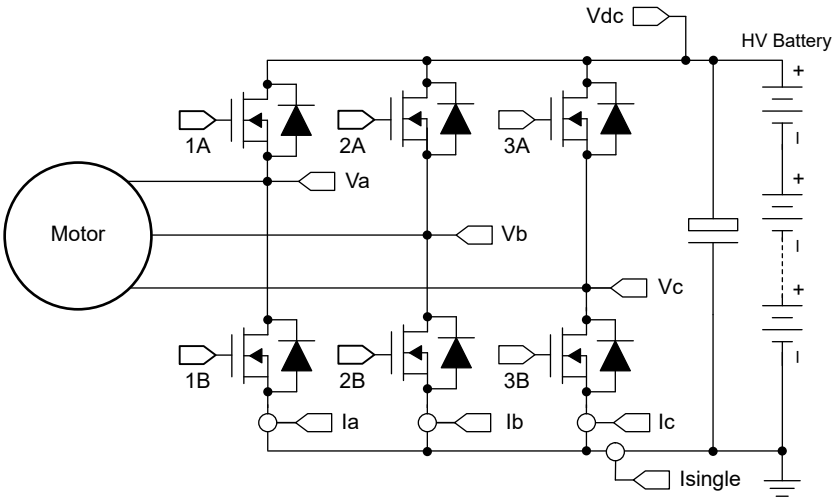


图 3-1. TIDM-02012 的硬件电路板方框图

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-02012 PCB 的不同块。

- 直流总线输入
  - 直流总线输入连接器
  - 额定电压为 1.3kV 的 10uF 薄膜电容器
- 用于 eCompressor 电机的三相逆变器
  - 功率高达 5kW 的三相逆变器支持 PMSM 或 IPM
  - 15 kHz 开关频率
  - 用于电流检测的 3 分流器/单分流器
  - 模拟信号的放大和输入滤波器
- 控制
  - TMS320F28039C MCU controlCARD
  - 具有 FPU 和 TMU 的 100MHz 32 位 CPU
- 辅助电源

- 板载电源可产生隔离的 +6V 和 +16V。然后，+6V 通过 LDO 驱动至 +5V 和+3.3V，为 controlCARD 和控制电路供电。

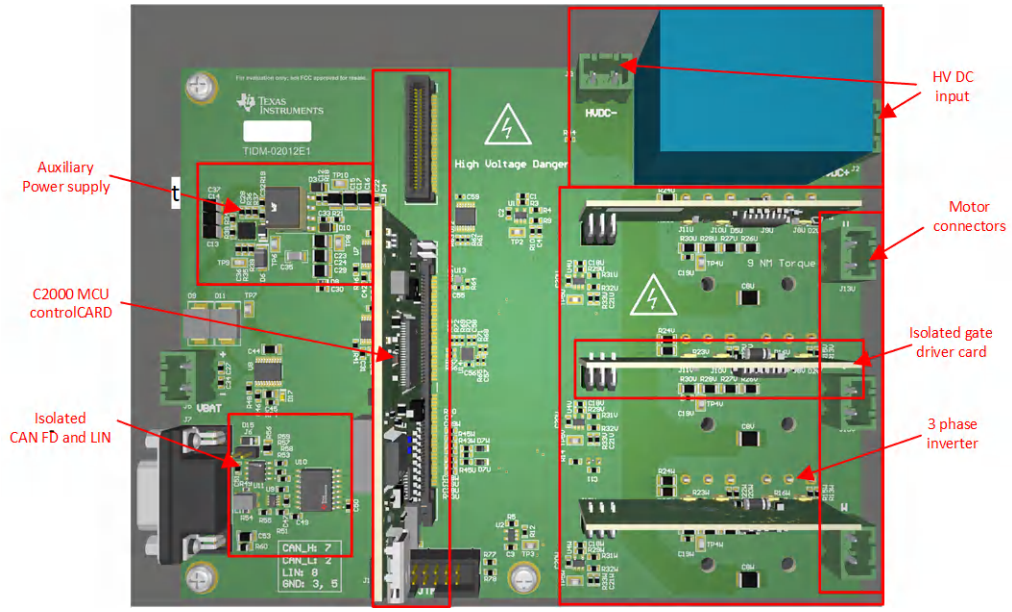


图 3-2. eCompressor 参考设计电路板布局布线

备注

虽然 TMS320F280039C controlCARD 仅采用一个 120 引脚 HSEC 连接器，但我们保留了 180 引脚 HSEC 连接器，用于与未来器件兼容。

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

- 对于输入，如果使用直流电源，电源支持高达 800V 的电压。将输入直流电源的输入电流限制设置为 25A，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

- 直流电源
- 数字示波器
- 万用表
- 三相 PM 同步电机
- 测力计
- 三相功率分析仪

3.2 测试设置

3.2.1 硬件设置

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 USB 电缆连接到 TMS320F280039C controlCARD，进行 JTAG 连接。
2. 将压缩机电动机电线连接到端子 J13U、J13V 和 J13W。
3. 向端子 J5 施加 12V 辅助直流电源。
4. 在端子 J2 和 J3 上施加直流总线电源。直流电源的最大输出为 800VDC。
5. 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。

3.2.2 软件设置

下载并安装构建和运行工程所需的 Code Composer Studio (CCS)。此工程需要 CCS 版本 12.0.0 或更高版本。有关 CCS 安装和使用的更多详细信息，请参阅 CCS 用户指南。

此参考设计的软件在 C2000Ware MotorControl SDK v4.01.00.00 或更高版本中提供。下载并安装后，您可以转至 `<SDK 安装位置>\solutions\tidm_02012_ecompressor\` 浏览此设计的文件夹。

3.2.2.1 Code Composer Studio 工程

要在 CCS 中导入参考工程，请点击 **Project → Import CCS Projects**，然后浏览到 `<SDK 安装位置>\solutions\tidm_02012_ecompressor\<device>\ccs`，然后点击 **Select Folder**。选择名为 *tidm_02012_ecompressor_<device>* 的工程，然后点击 **Finish**。现在，该工程应在 CCS 的 **Project Explorer** 窗格中可见。

src_foc 文件夹包含典型的 FOC 模块，包括变换、PID 函数和估算器。*src_lib* 文件夹包含 InstaSPIN-FOC 库和相关头文件。这些模块独立于特定器件和电路板，还可用于 SDK 中的多种其他解决方案。

src_control 文件夹包含电机驱动控制文件，这些文件在中断服务例程和后台任务中调用电机控制核心算法函数。*src_sys* 文件夹包含为其他系统功能保留的一些文件，例如 CAN 通信的驱动程序。您可以自行添加用于系统控制、通信等功能的代码。这些模块专用于此参考设计工程，但与器件和电路板无关。

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。这些文件包含特定于器件的驱动程序，用于运行解决方案。如果要将工程迁移到您自己的电路板或其他器件，只需根据器件或电路板的引脚分配和功能更改 *hal.c*、*hal.syscfg*、*hal.h* 和 *user_mtr1.h* 文件。

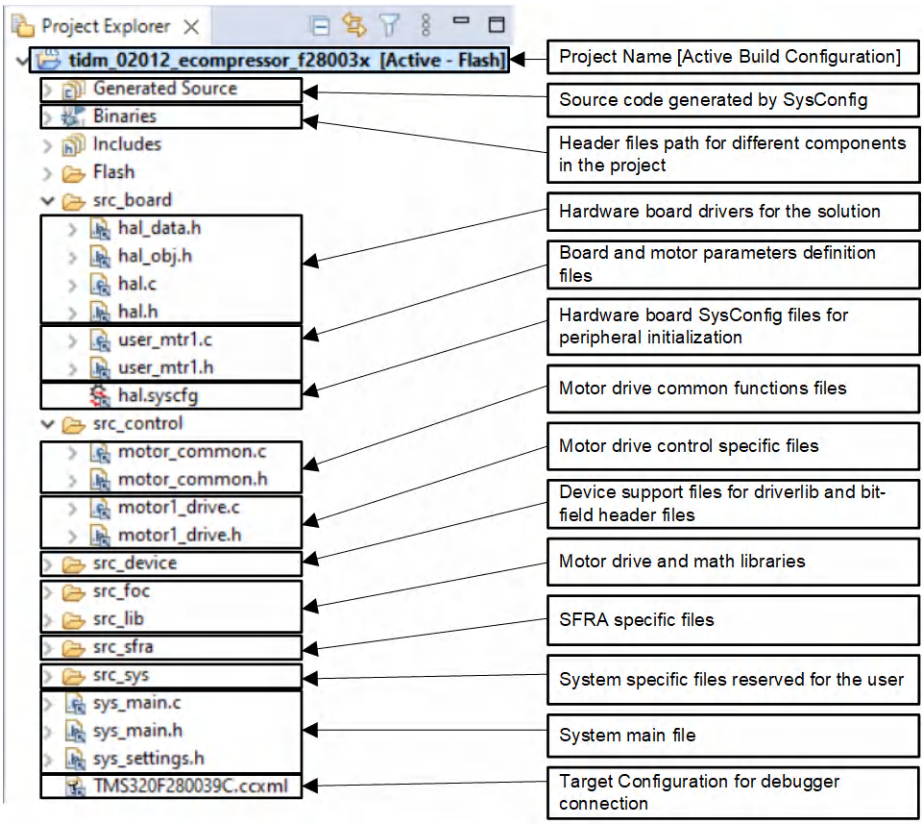


图 3-3. 参考工程的 Project Explorer 视图

软件中有几项可选功能，可以使用工程属性中的预定义符号来启用和禁用。这些预定义符号如下：

- **MOTOR1_OVM** 可启用过调制
- **MOTOR1_FWC** 可启用弱磁控制 (FWC)
- **MOTOR1_MTPA** 可启用每安培最大扭矩 (MTPA) 模式
- **MOTOR1_VIBCOMPA** 或 **MOTOR1_VIBCOMPT** 可启用振动补偿

要查看和编辑预定义符号，请右键点击工程并选择 **Properties**。然后转至 **C2000 Compiler** 选项的 **Predefined Symbols** 部分，如图 3-4 所示。默认情况下，符号名称后附加的“_N”可禁用上文列出的功能。编辑符号，删除“_N”可启用该功能。

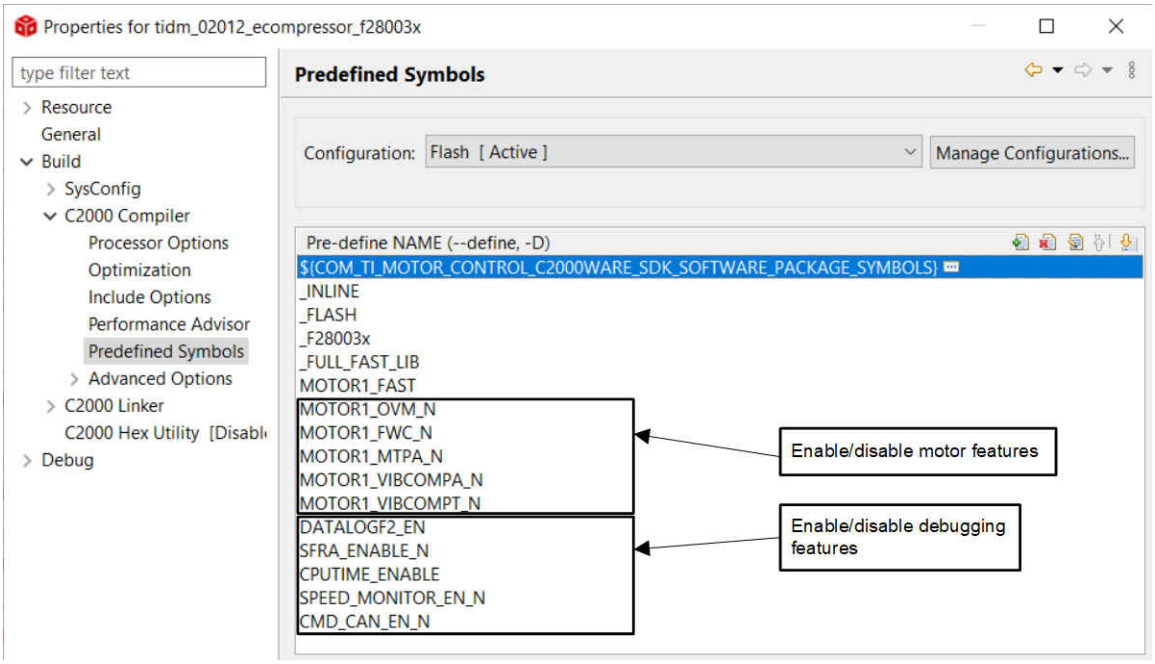


图 3-4. 在工程属性中选择预定义符号

3.2.2.2 软件结构

图 3-5 中显示了工程的总体结构。器件外设配置基于 C2000Ware Driverlib，使用 SysConfig 部分生成。如果要将参考设计软件迁移到您自己的电路板或另一器件，只需更改 *hal.c*、*hal.syscfg* 和 *hal.h* 文件以及 *user_mtr1.h* 中的参数。

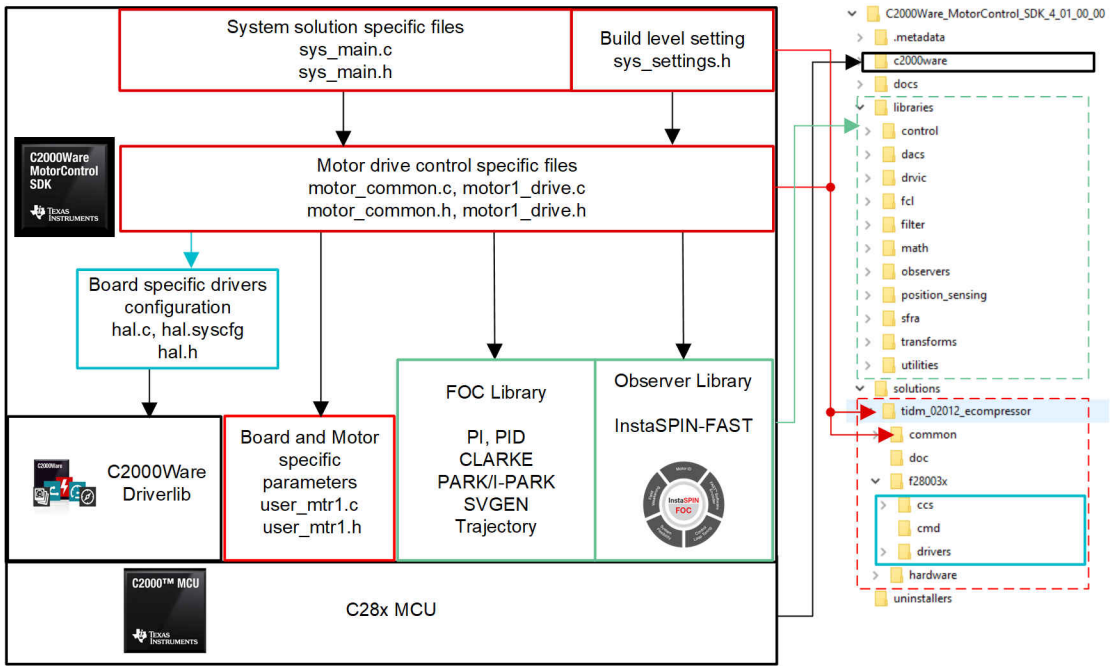


图 3-5. 工程结构概览

图 3-6 显示了固件的工程软件流程图，其中包括一个用于实时电机控制的 ISR、一个主循环用于在后台循环中更新电机控制参数。ISR 将由 ADC 转换结束 (EOC) 触发。

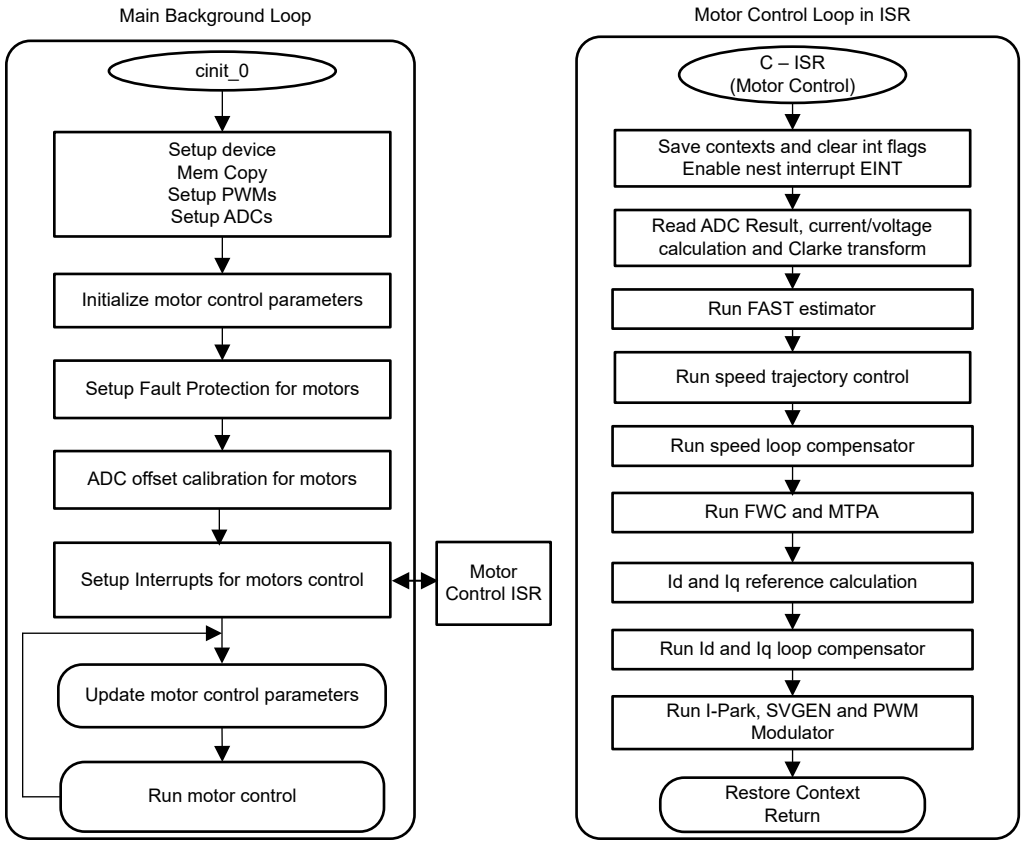


图 3-6. 后台软件和电机控制 ISR 流程图

为了简化系统开发和设计，将该软件组织为四个增量构建，如表 3-1 中所述，这使得学习和熟悉电路板和软件变得更加容易。这种方法对于调试和测试电路板也有帮助。要选择特定的构建选项，您将编辑 *sys_settings.h* 文件中的 *DMC_BUILDLEVEL* 选项并重新编译工程。下一部分将详细描述此过程。

表 3-1. 递增构建选项

DMC_BUILDLEVEL 值	说明
DMC_LEVEL_1	50% 占空比，偏移校准并验证相移
DMC_LEVEL_2	开环控制，用于检查检测信号
DMC_LEVEL_3	闭合电流环路，用于检查硬件设置
DMC_LEVEL_4	参数识别并使用 InstaSPIN-FOC 运行

3.3 测试步骤

此系统软件被逐步构建，直到最终系统可以令人放心地运转。要选择特定的构建选项，请在 *sys_settings.h* 中选择相应的 *DMC_BUILDLEVEL* 选项。选中构建选项后，右键点击工程名称并点击 **Rebuild Project** 编译工程。

```
//=============================================================================
// Incremental Build options for System check-out
//=============================================================================
#define DMC_LEVEL_1   1 // 50% duty, offset calibration and verify phase shift
#define DMC_LEVEL_2   2 // Open loop control to check sensing signals
#define DMC_LEVEL_3   3 // Closed current loop to check the hardware settings
#define DMC_LEVEL_4   4 // Parameters identification and run with InstaSPIN-FOC

#define DMC_BUILDLEVEL  DMC_LEVEL_4
```

3.3.1 1 级递增构建

此构建级别的目标：

- 使用 HAL 初始化 MCU 的外设。
- 验证 PWM 和 ADC 驱动器模块。
- 确认 ADC 偏移验证。
- 熟悉 CCS 的操作。

在该构建级别中，电路板以开环模式执行（采用固定占空比）。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。此外，可以在该构建级别中执行输入和输出电压检测校准。在此过程中，电机必须保持断开。图 3-7 显示了该构建级别的软件方框图。

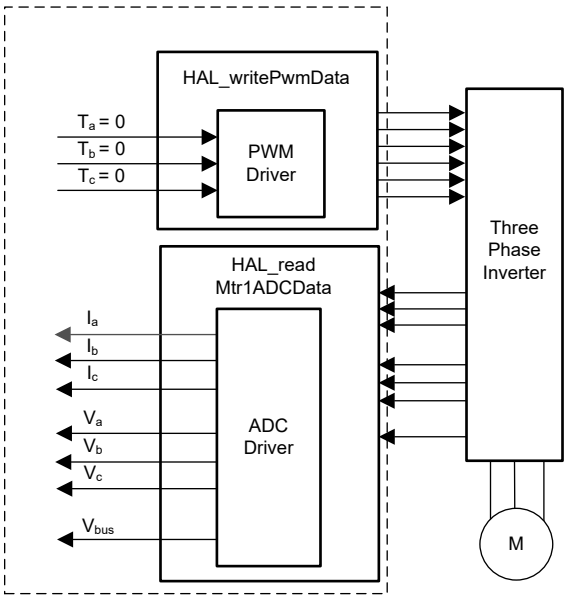


图 3-7. 构建级别 1 软件方框图

3.3.1.1 工程设置

将工程导入 CCS，并选择适当的构建配置。打开 *sys_settings.h* 文件并将 *DMC_BUILDLEVEL* 的 #define 设置为 *DMC_LEVEL_1*。这会将工程配置为运行第一个增量构建。右键点击 **Project Explorer** 内的工程并选择 **Rebuild Project**。确认 **Console** 窗格显示工程构建时没有任何错误。

成功完成构建后，选中 *tidm_02012_ecompressor* 工程，转至 **Run → Debug**，或点击工具栏上的 **Debug** 按钮。工程默认情况下将使用工程中的 *TMS320F280039C.ccxml* 文件启动调试会话。*TMS320F280039C.ccxml* 配置为使用 TMDSCNCD280039C controlCARD 电路板上的德州仪器 (TI) XDS110 USB 调试探针。

点击 **Debug** 后，CCS 将自动连接至目标，将输出文件载入器件内并更改为 **CCS Debug** 视图。程序应该在 main() 的开始处暂停。

如果 **Expressions** 窗格尚未打开，请点击 CCS 菜单栏中的 **View → Expressions**。您可以手动添加变量、也可以导入与此构建级别关联的推荐变量列表，方法是在 **Expressions** 窗口中右键点击并选择 **Import...**，并找到文件 `<SDK 安装位置>\solutions\tidm_02012_ecompressor\common\debug\tidm_02012_level1.txt`。点击 **OK**，窗口将填充图 3-8 中所示的变量。

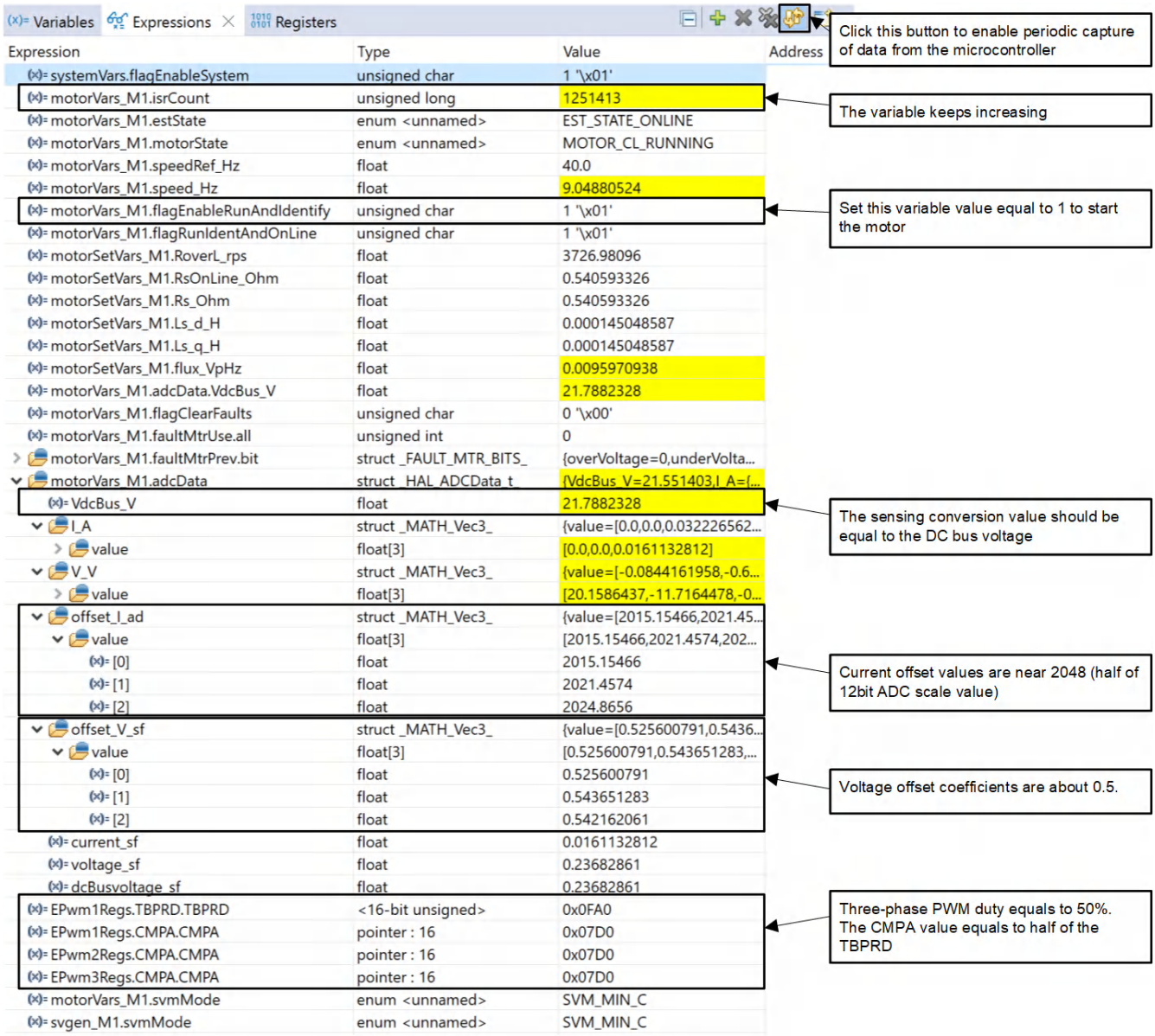
点击“Expressions”窗口工具栏中的 **Continuous Refresh** 按钮，告知 CCS 以 CCS 调试首选项中定义的速率持续更新数据。

3.3.1.2 运行应用程序

转至 **Run → Resume** 或点击工具栏中的 **Resume** 按钮来运行代码。该工程现在应该开始运行，您应该会在“Expressions”窗口中看到变量更新，并且 controlCARD 上的 D2 LED 会闪烁。检查以下各项，确认应用和硬件设置能够正常工作：

- 看到 *systemVars.flagEnableSystem* 自动设为 1 后，将变量 *motorVars_M1.flagEnableRunAndIdentify* 设为 1。您的变量应与图 3-8 中所示的类似。

- 如果未检测到故障，变量 *motorVars_M1.flagRunIdentAndOnLine* 应自动设为 1。*motorVars_M1.isrCount* 应不断增加。
- 检查电机逆变器板的校准偏移。电机相电流检测值的偏移值应等于 ADC 满量程电流的大约一半。
- 使用示波器探测电机驱动控制的 PWM 输出。在此构建级别中，三个 PWM 的占空比设置为 50%。PWM 开关频率应与 *user_mtr1.h* 中为 *USER_M1_PWM_FREQ_kHz* 定义的频率相同。



Expression	Type	Value	Address
systemVars.flagEnableSystem	unsigned char	1 '\x01'	
motorVars_M1.isrCount	unsigned long	1251413	
motorVars_M1.estState	enum <unnamed>	EST_STATE_ONLINE	
motorVars_M1.motorState	enum <unnamed>	MOTOR_CL_RUNNING	
motorVars_M1.speedRef_Hz	float	40.0	
motorVars_M1.speed_Hz	float	9.04880524	
motorVars_M1.flagEnableRunAndIdentify	unsigned char	1 '\x01'	
motorVars_M1.flagRunIdentAndOnLine	unsigned char	1 '\x01'	
motorSetVars_M1.RoverL_rps	float	3726.98096	
motorSetVars_M1.RsOnLine_Ohm	float	0.540593326	
motorSetVars_M1.Rs_Ohm	float	0.540593326	
motorSetVars_M1.Ls_d_H	float	0.000145048587	
motorSetVars_M1.Ls_q_H	float	0.000145048587	
motorSetVars_M1.flux_VpHz	float	0.0095970938	
motorVars_M1.adcData.VdcBus_V	float	21.7882328	
motorVars_M1.flagClearFaults	unsigned char	0 '\x00'	
motorVars_M1.faultMtrUse.all	unsigned int	0	
motorVars_M1.faultMtrPrev.bit	struct _FAULT_MTR_BITS_	{overVoltage=0,underVolta...	
motorVars_M1.adcData	struct _HAL_ADCData_t_	{VdcBus_V=21.551403,I_A={...	
VdcBus_V	float	21.7882328	
I_A	struct _MATH_Vec3_	{value=[0.0,0.0,0.032226562...	
value	float[3]	[0.0,0.0,0.0161132812]	
V_V	struct _MATH_Vec3_	{value=[-0.0844161958,-0.6...	
value	float[3]	[20.1586437,-11.7164478,-0...	
offset_I_ad	struct _MATH_Vec3_	{value=[2015.15466,2021.45...	
value	float[3]	[2015.15466,2021.4574,202...	
[0]	float	2015.15466	
[1]	float	2021.4574	
[2]	float	2024.8656	
offset_V_sf	struct _MATH_Vec3_	{value=[0.525600791,0.5436...	
value	float[3]	[0.525600791,0.543651283,...	
[0]	float	0.525600791	
[1]	float	0.543651283	
[2]	float	0.542162061	
current_sf	float	0.0161132812	
voltage_sf	float	0.23682861	
dcBusvoltage_sf	float	0.23682861	
EPwm1Regs.TBPRD.TBPRD	<16-bit unsigned>	0x0FA0	
EPwm1Regs.CMPA.CMPA	pointer : 16	0x07D0	
EPwm2Regs.CMPA.CMPA	pointer : 16	0x07D0	
EPwm3Regs.CMPA.CMPA	pointer : 16	0x07D0	
motorVars_M1.svmMode	enum <unnamed>	SVM_MIN_C	
svgen_M1.svmMode	enum <unnamed>	SVM_MIN_C	

图 3-8. 表达式视图中的构建级别 1 变量

您可以首先点击工具栏上的 **Suspend** 按钮或选择 **Target → Suspend** 来暂停 CPU。要从头开始重新运行应用，请点击 **CPU Reset** 工具栏按钮或点击 **Run → Reset → CPU Reset**，然后点击 **Restart** 按钮或 **Run → Restart** 来重置控制器。您可以点击 **Terminate** 按钮，或点击 **Run → Terminate** 来关闭 CCS 调试会话。程序将暂停，并断开 CCS 与控制器的连接。

请注意，每次更改代码时无需终止调试会话。您可以转到 **Run → Load → Load Program...**（如果是同一个文件，请选择 **Reload Program...**）。如果 CCS 检测到您已重新编译可执行文件，还会自动提示您，询问是否要重新加载该可执行文件。

3.3.2 2 级递增构建

此构建级别的目标：

- 实现简单的电机标量 v/f 控制以驱动电机，从而验证电流和电压检测电路以及栅极驱动器电路。
- 测试用于电机控制的 InstaSPIN-FOC FAST 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。图 3-9 显示了该构建级别的软件流程。

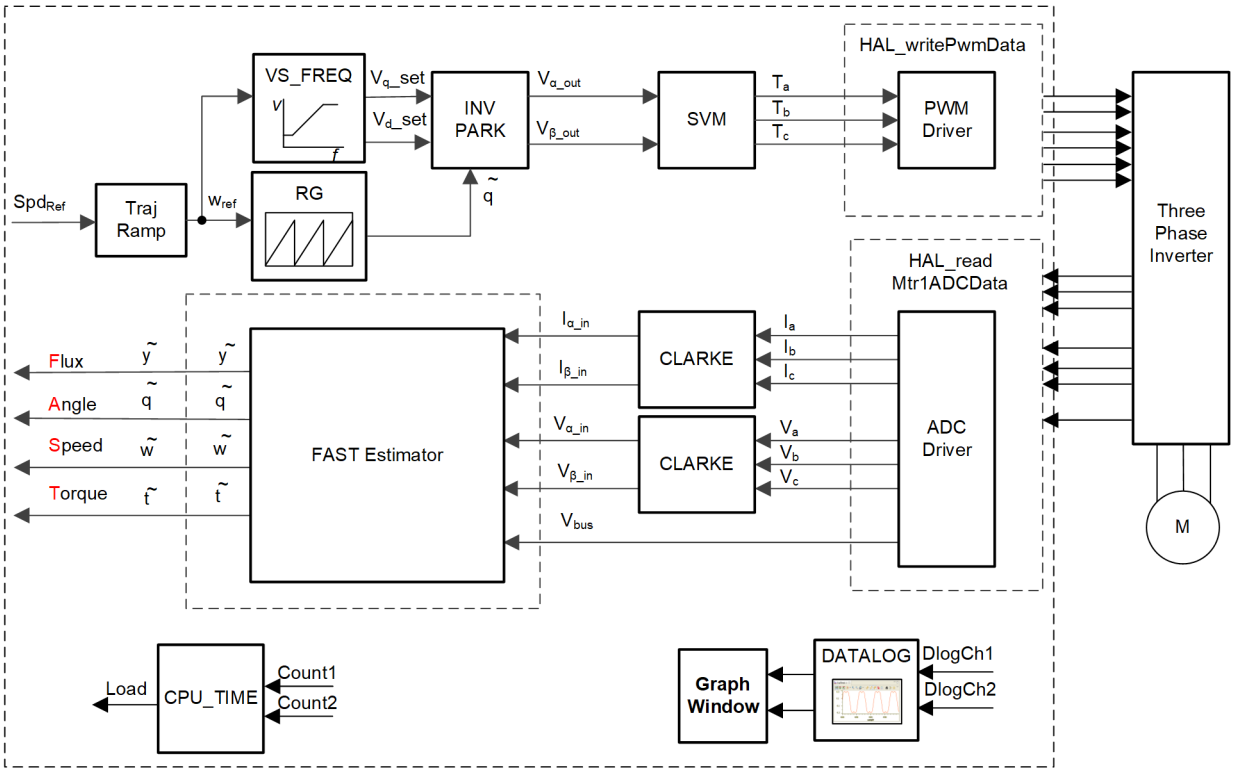


图 3-9. 构建级别 2 软件方框图

3.3.2.1 工程设置

按照节 3.3.1.1 中描述的步骤，将 *DMC_BUILDLEVEL* 更改为 *DMC_LEVEL_2*，然后重新编译和加载工程。您还应该将 **Expressions** 视图的内容更改为 *tidm_02012_level2.txt* 中列出的变量。

3.3.2.2 运行应用程序

转至 **Run** → **Resume** 或点击工具栏中的 **Resume** 按钮来运行代码。请检查以下各项：

- *systemVars.flagEnableSystem* 需要在固定时间后设为 1，这意味着已完成偏移校准。故障标志 *motorVars_M1.faultMtrUse.all* 应等于 0。将变量 *motorVars_M1.flagEnableRunAndIdentify* 设为 1。
- 电机将以 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
```

- 此后，电机以 motorVars_M1.speedRef_Hz 变量中设置的速度旋转，在“Expressions”窗口中检查 motorVars_M1.speed_Hz 的值，这两个变量的值应非常接近，如图 3-10 所示。
- 将 DATALOG 模块与 CCS **Graph** 实用程序配合使用，检查电流检测信号。转至 **Tools** → **Graph** → **Dual Time** 启动此工具，然后点击 **Import** 按钮导航至文件 `<SDK 安装位置>\solutions\tidm_02012_ecompressor\common\debug\motor_datalog_fp2.graphProp`。点击

OK，应以图形方式显示当前读数，如图 3-11 所示。下面的代码显示了如何在 *sys_main.h* 中配置要记录的变量。有关 DATALOG 模块的更多详细信息，请参阅电机控制 SDK 通用工程和实验用户指南。

```
// set datalog parameters
datalogObj->iptr[0] = &motorVars_M1.adcData.I_A.value[0];
datalogObj->iptr[1] = &motorVars_M1.adcData.I_A.value[1];
```

- 通过减小变量 *motorVars_M1.overCurrent_A* 的值来验证过流故障保护，过流保护由 CMPSS 模块实现。如果将 *motorVars_M1.overCurrent_A* 设为小于电机相电流实际值的值，则会触发过流故障。发生故障时，PWM 输出将被禁用， *motorVars_M1.flagEnableRunAndIdentify* 将清零，*motorVars_M1.faultMtrUse.all* 将被设置为 0x10 (16)，如图 3-10 所示。
- 将变量 *motorVars_M1.flagEnableRunAndIdentify* 设为 0 停止电机。终止调试会话并关闭逆变器板的电源。

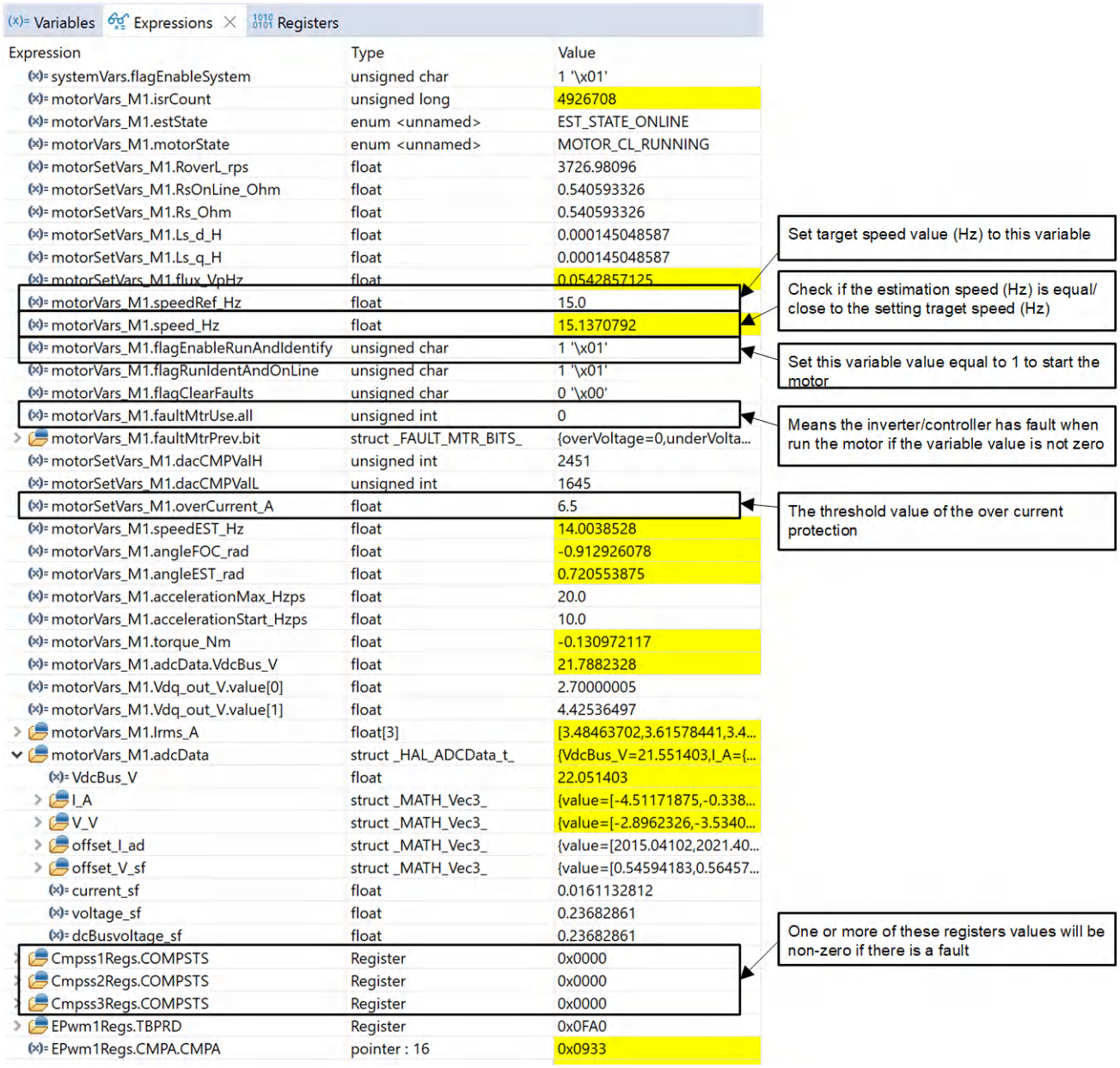


图 3-10. 表达式视图中的构建级别 2 变量

Graph 工具可用于显示 DATALOG 缓冲区。

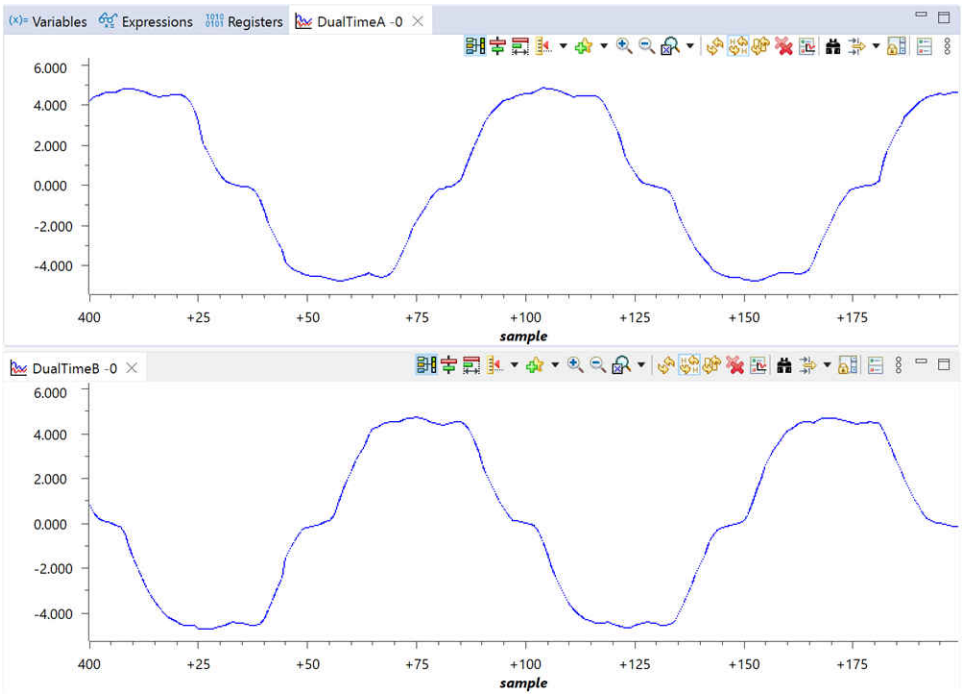


图 3-11. 在 CCS Graph 工具中构建级别 2 数据记录的电流读数

过流故障将触发 PWM 输出，并反映在故障标志变量中。

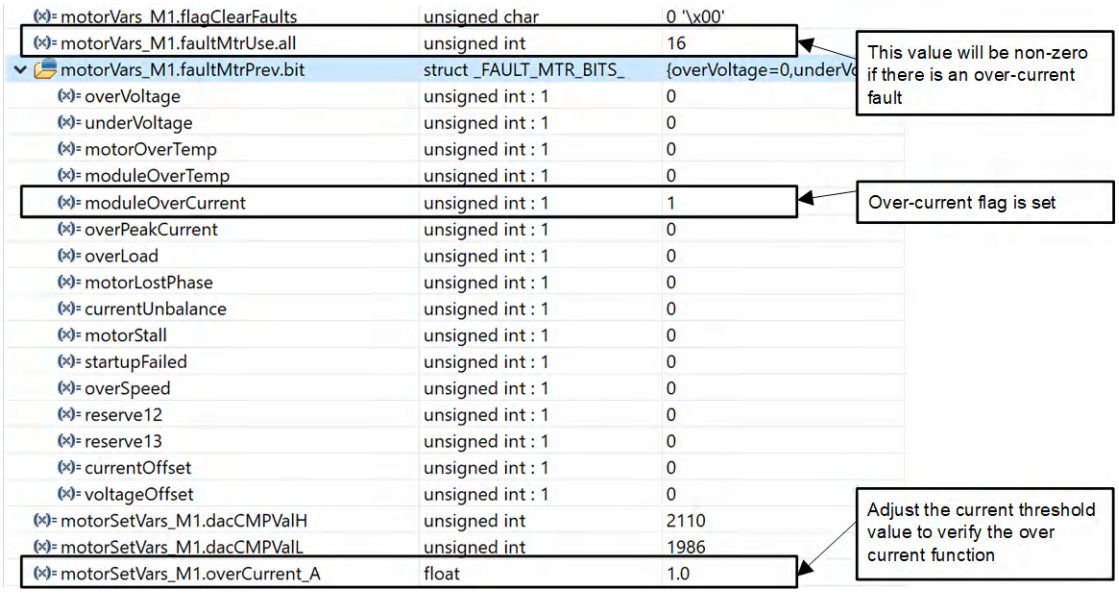


图 3-12. 表达式视图中的构建级别 2 过流保护检查

3.3.3 3 级递增构建

此构建级别的目标：

- 评估电机的闭合电流环路运行。
- 验证电流检测参数设置。

在这个构建级别，电机由 i/f 控制进行控制，转子角度由斜坡发生器模块生成。图 3-13 显示了该构建级别的软件流程。

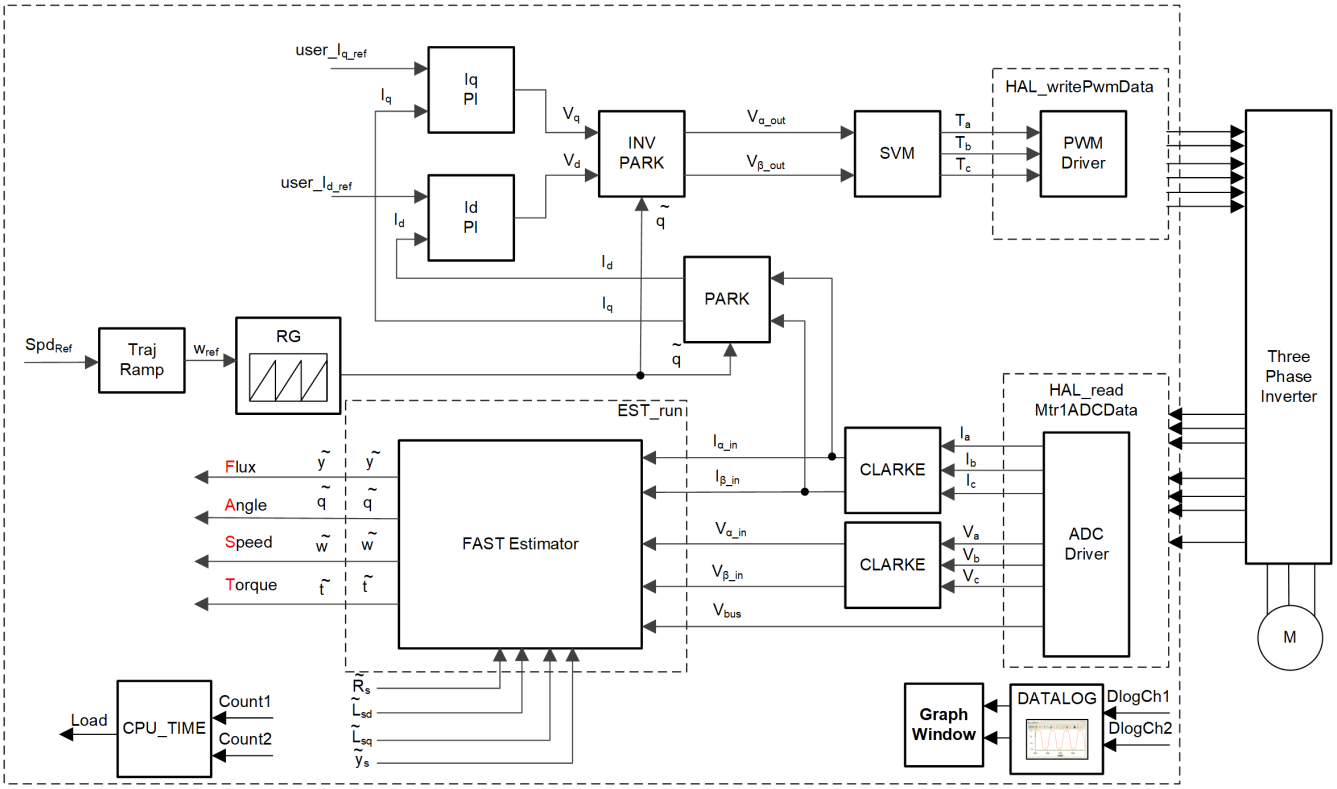


图 3-13. 构建级别 3 软件方框图

3.3.3.1 工程设置

按照节 3.3.1.1 中描述的步骤，将 *DMC_BUILDLEVEL* 更改为 *DMC_LEVEL_3*，然后重新编译和加载工程。您还应该将 **Expressions** 视图的内容更改为 *tidm_02012_level3.txt* 中列出的变量。

3.3.3.2 运行应用程序

转至 **Run → Resume** 或点击工具栏中的 **Resume** 按钮来运行代码。请检查以下各项：

- *systemVars.flagEnableSystem* 应在固定时间后设为 1，这意味着已完成偏移校准。故障标志 *motorVars_M1.faultMtrUse.all* 应等于 0。
- 运行电机时要进行电流闭环控制，请在 **Expressions** 窗口中将变量 *motorVars_M1.flagEnableRunAndIdentify* 设置为 1。电机在运行时将使用来自角度发生器的角度进行闭环控制，运行速度在 *motorVars_M1.speedRef_Hz* 变量中设置。检查 *motorVars_M1.speed_Hz* 的值，确认这两个值非常接近。
- 在“Expressions”窗口中更改 *Idq_set_A.value[1]* 的值，以设置基准扭矩电流。相应地，电机相电流将以相同的百分比增加。
- 将变量 *motorVars_M1.flagEnableRunAndIdentify* 设为 0 停止电机。终止调试会话并关闭逆变器板的电源。

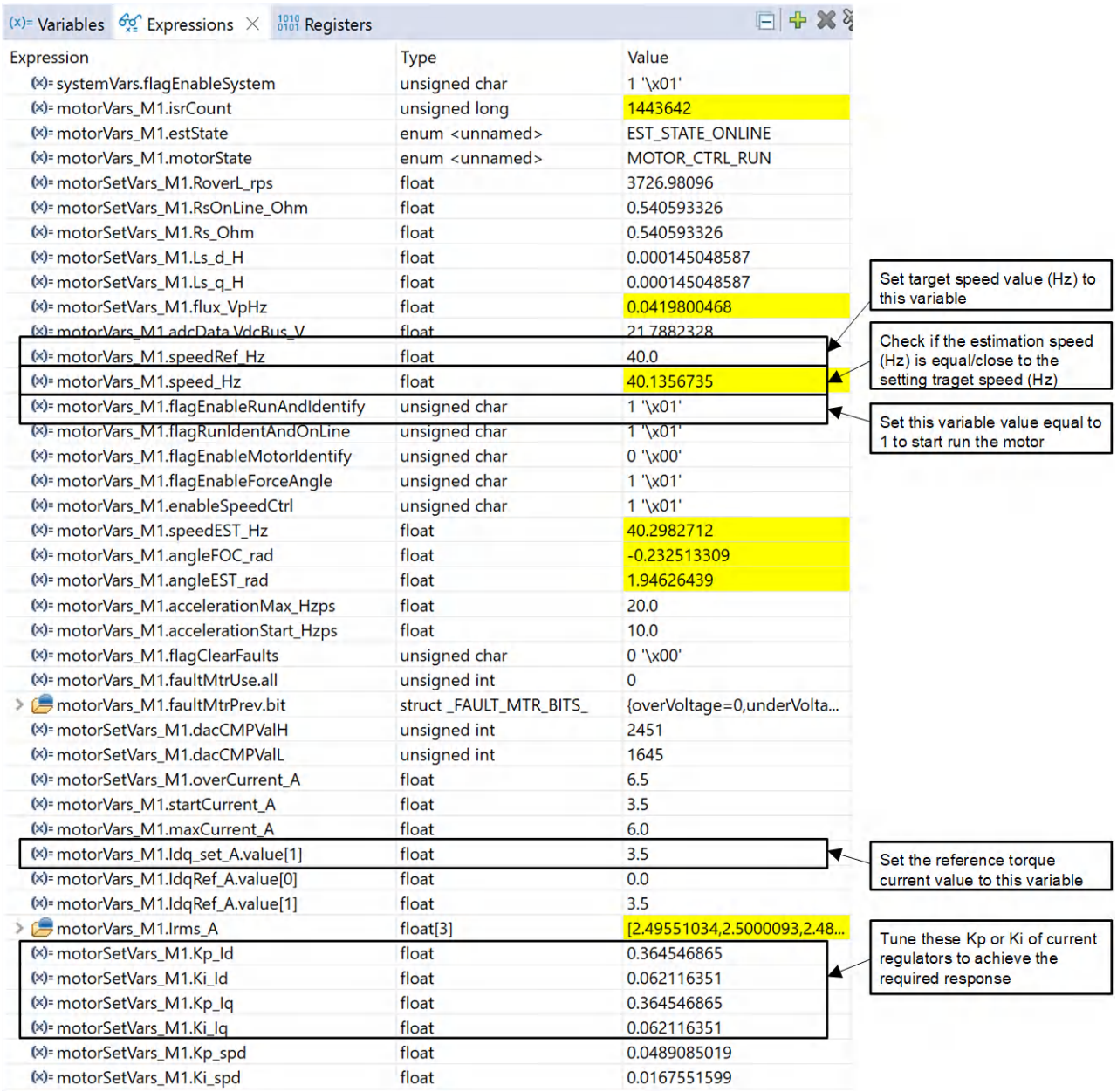


图 3-14. 表达式视图中的构建级别 3 变量

3.3.4 4 级递增构建

此构建级别的目标：

- 使用 FAST 估算器评估电机识别。
- 使用基于 FAST 的无传感器 FOC 评估整个电机驱动器。
- 评估其他特性，例如弱磁控制、MTPA 和扭矩补偿。

在此构建级别，外部速度环路是闭合的，内部电流环路适用于转子角度来自 FAST 的电机。图 3-15 显示了该构建级别的软件流程。

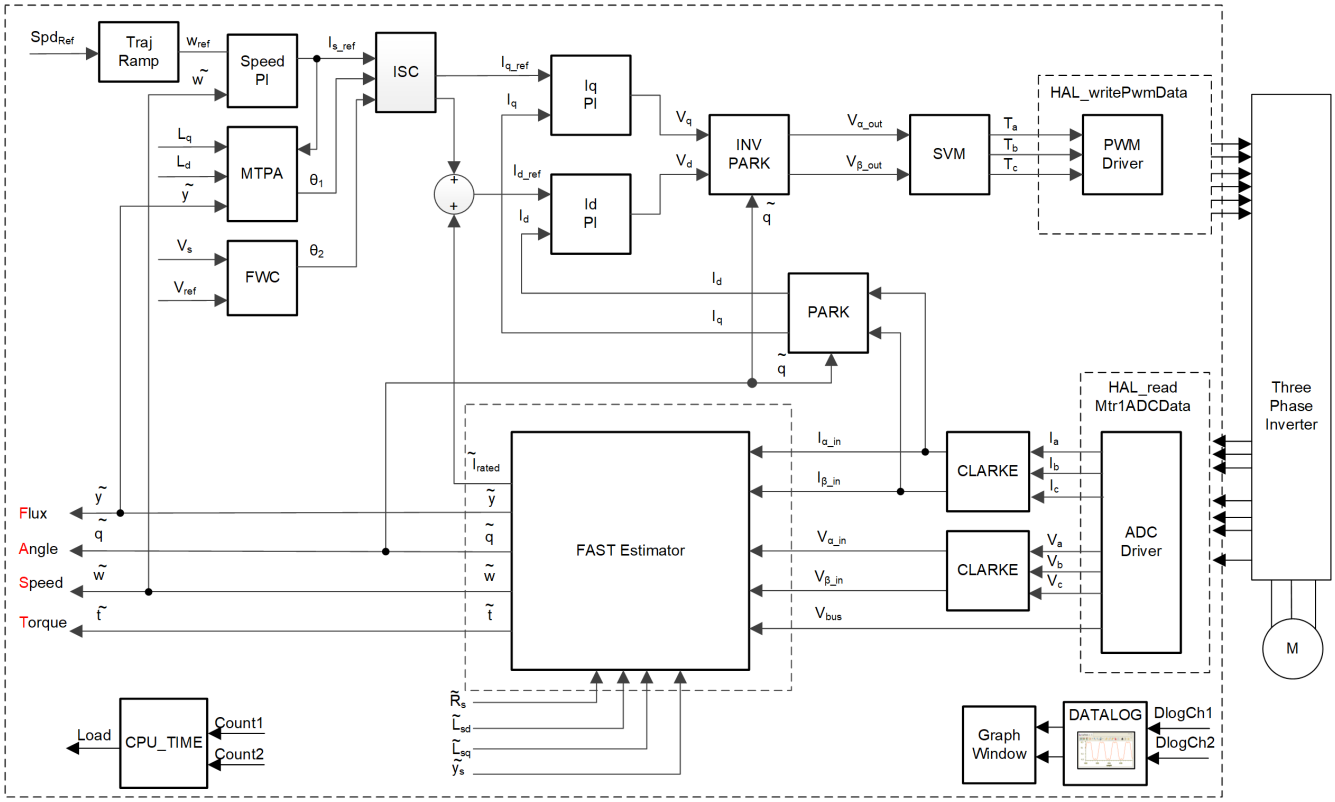


图 3-15. 构建级别 4 软件方框图

3.3.4.1 工程设置

按照节 3.3.1.1 中描述的步骤，将 *DMC_BUILDLEVEL* 更改为 *DMC_LEVEL_4*。应该对此构建级别进行若干额外的编辑：

- 要运行电机识别，您还应将 *sys_settings.h* 中的 *MOTOR_IDENT* 定义更改为 1，然后重新编译和加载工程。
- 必须在头文件 *user_mtr1.h* 中定义所需的电机参数，如以下示例代码所示。如果用户不太了解电机参数，那么在示例实验中使用 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                  (0.540593326f)
#define USER_MOTOR1_Ls_d_H                  (0.000145048587f)
#define USER_MOTOR1_Ls_q_H                  (0.000145048587f)
#define USER_MOTOR1_RATED_FLUX_VpHz         (0.038f)
```

- 根据电机规格在 user_mtr1.h 中设置正确的识别值。

```
#define USER_MOTOR1_RES_EST_CURRENT_A (1.5f)  // A - 10~30% of rated current of the motor
#define USER_MOTOR1_IND_EST_CURRENT_A (-1.0f) // A - 10~30% of rated current of the motor,
                                              // just enough to enable rotation
#define USER_MOTOR1_MAX_CURRENT_A (6.0f)      // A - 30~150% of rated current of the motor
#define USER_MOTOR1_FLUX_EXC_FREQ_Hz (40.0f)  // Hz - 10~30% of rated frequency of the motor
```

重新编译和加载工程。您还应该将 **Expressions** 视图的内容更改为 *tidm_02012_level4.txt* 中列出的变量。

3.3.4.2 运行应用程序

转至 **Run → Resume** 或点击工具栏中的 **Resume** 按钮来运行代码。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。故障标志 *motorVars_M1.faultMtrUse.all* 应等于 0。

将变量 *motorVars_M1.flagEnableRunAndIdentify* 设为 1。将执行电机识别；整个过程大约需要 150 秒。*motorVars_M1.flagEnableRunAndIdentify* 等于 0 并且电机停止后，电机参数即被识别。复制“Expressions”窗口中的变量值，替换 *user_mtr1.h* 中定义的电机参数，如下所示：

- USER_MOTOR1_Rs_Ohm = *motorSetVars_M1.Rs_Ohm* 的值
- USER_MOTOR1_Ls_d_H = *motorSetVars_M1.Ls_d_H* 的值
- USER_MOTOR1_Ls_q_H = *motorSetVars_M1.Ls_q_H* 的值
- USER_MOTOR1_RATED_FLUX_VpHz = *motorSetVars_M1.flux_VpHz* 的值

成功识别电机参数后，将 *MOTOR_IDENT* 改回 0 以禁用识别。重新编译并重新加载应用。

电机参数识别完成，或在 *user_mtr1.h* 文件中正确设置后，再次将 *motorVars_M1.flagEnableRunAndIdentify* 设置为 1，启动电机。请检查以下各项：

- 使用变量 *motorVars_M1.speedRef_Hz* 设置目标速度值，并观察电机轴速度如何随设定速度改变。要改变加速度，请在变量 *motorVars_M1.accelerationMax_Hzps* 中输入不同的值。
- 如节 3.3.2 中所述，您可以使用 CCS Graph 工具来监控变量。默认情况下，此构建级别会记录电机角度和电流波形。
- FOC 系统电流控制器的默认比例增益 (Kp) 和积分增益 (Ki) 在函数 *setupControllers()* 中计算。调用 *setupControllers()* 后，全局变量 motorSetVars_M1.Kp_Id、*motorSetVars_M1.Ki_Id*、*motorSetVars_M1.Kp_Iq* 和 *motorSetVars_M1.Ki_Iq* 将使用新计算得出的 Kp 和 Ki 增益进行初始化。如图 3-16 中所示，调整 **Expressions** 窗口中这四个变量的 Kp 和 Ki 值，使电流控制器实现预期的电流控制带宽和响应。Kp 增益会产生一个零点，以抵消电机定子的极点，可以轻松地计算得出。Ki 增益可调整电流控制器-电机系统的带宽。如果需要一个速度控制系统来实现特定阻尼，电流控制器的 Kp 增益将与速度控制系统的时间常数相关。
- FOC 系统速度控制器的默认比例增益 (Kp) 和积分增益 (Ki) 也在函数 *setupControllers()* 中计算。调用 *setupControllers()* 后，全局变量 *motorSetVars_M1.Kp_spd* 和 *motorSetVars_M1.Ki_spd* 将使用新计算得出的 Kp 和 Ki 增益进行初始化。如图 3-16 中所示，调整 **Expressions** 窗口中这两个变量的 Kp 和 Ki 值，使速度控制器实现预期的电流控制带宽和响应。与调整电流控制器相比，调整速度控制器的未知情况更多。默认计算得出的 Kp 和 Ki 只作为起点。

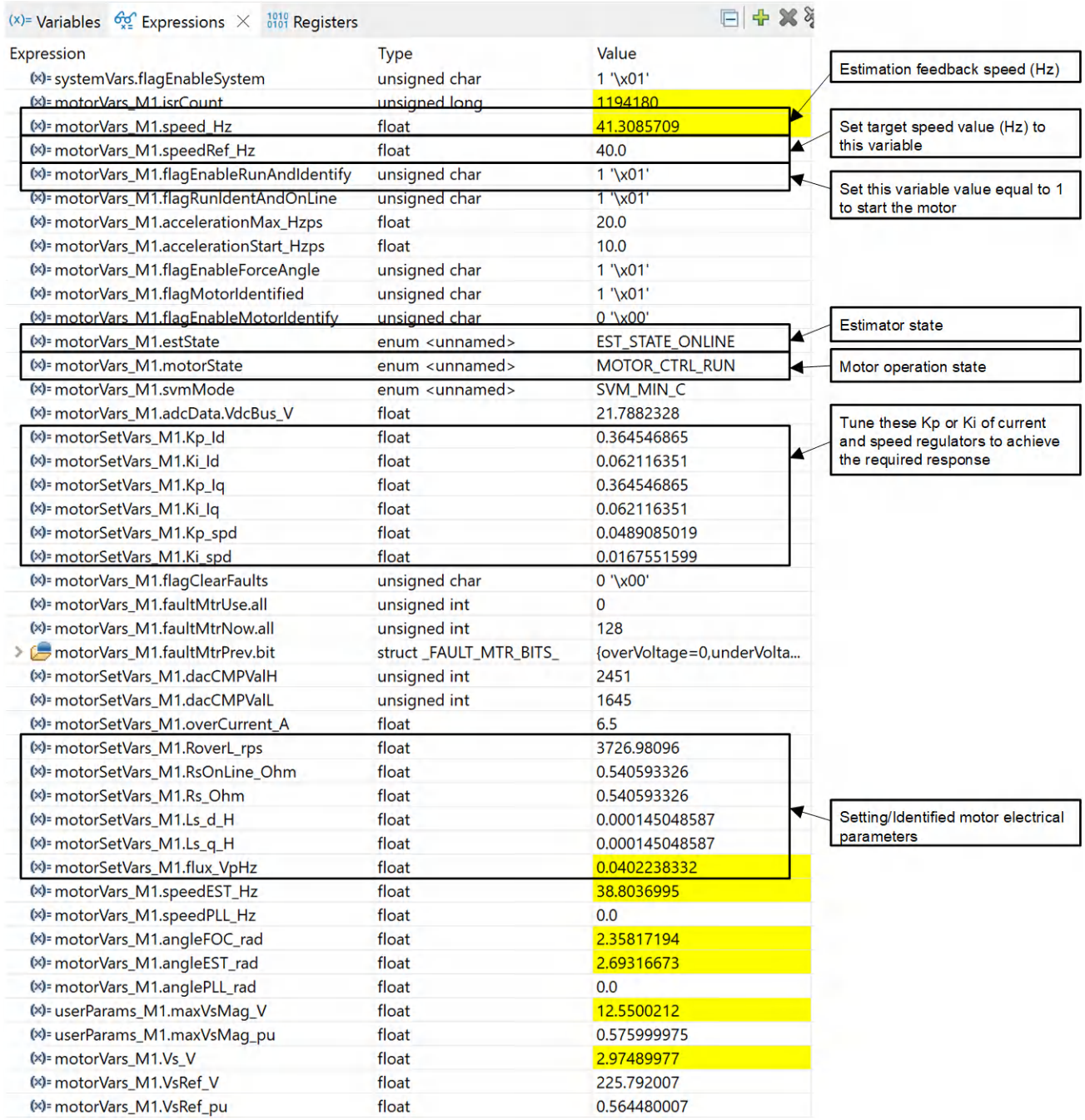


图 3-16. 表达式视图中的构建级别 4 变量

3.3.4.3 调整弱磁和 MTPA 控制

可以选择在电机驱动器 ISR 中调用 FWC 和 MTPA 函数来计算电流角，然后计算 d 轴和 q 轴的基准电流。在工程的编译器设置中添加预定义符号 *MOTOR1_FWC* 和 *MOTOR1_MTPA*（如节 3.2.2.1 所述），分别启用 FWC 和 MTPA。

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。

在 CCS Debug 透视图中向 **Expressions** 窗口添加变量 *motorVars_M1.VsRef_pu*、*motorSetVars_M1.Kp_fwc* 和 *motorSetVars_M1.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。

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“Expressions”窗口值：

- USER_M1_FWC_VREF = *motorVars_M1.VsRef_pu* 的值，FWC 的基准电压

- USER_M1_FWC_KP = *motorSetVars_M1.Kp_fwc* 的值，FWC PI 稳压器的 Kp 增益
- USER_M1_FWC_KI = *motorSetVars_M1.Ki_fwc* 的值，FWC PI 稳压器的 Ki 增益

MTPA 控制参数根据电机参数 L_d、L_q 和 ψ_m 计算，无需在线调整其他参数。

3.3.4.4 调整振动补偿

可以选择在电机驱动 ISR 中调用自动振动补偿功能，以计算前馈扭矩电流。如节 3.2.2.1 所述，在工程的编译器设置中添加预定义符号 *MOTOR1_VIBCOMPA*，以启用振动补偿。

在 CCS Debug 透视图中向 **Expressions** 窗口添加变量 *motorVars_M1.vibCompAlpha*、*motorVars_M1.vibCompGain* 和 *motorVars_M1.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。

- *motorVars_M1.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。
- *motorVars_M1.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。
- *motorVars_M1.vibCompGain* 是前馈扭矩基准扭矩电流值的增益系数（最大值为 1.0）。

更改速度参考 (*motorVars_M1.speedRef_Hz*) 和速度控制器增益（*motorSetVars_M1.Kp_spd* 和 *motorSetVars_M1.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。

在工程的构建配置中添加预定义符号 *SPEED_MONITOR_EN*，以启用电机运行转速振动。现在，通过设置标志 *motorVars_M1.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_M1.flagClearRecord = 1* 来实现新转速变化。记录转速变化约为 2Hz。

如果振动没有减少，请尝试增加转速控制器增益。此外，还可以通过将 *motorVars_M1.vibCompAlpha* 的值递减 0.02，尝试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 *motorVars_M1.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_M1.flagClearRecord = 1*。

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 = *motorVars_M1.vibCompAlpha* 的值，振动补偿模块的学习率从 0.0 到 1.0
- USER_MOTOR1_VIBCOMPA_GAIN = *motorVars_M1.vibCompGain* 的值，振动补偿模块的增益从 0.0 到 1.0
- USER_MOTOR1_VIBCOMPA_INDEX_DELTA = *motorVars_M1.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。在工程的编译器设置中添加预定义符号 ***MOTOR1_VIBCOMPT***，以启用此振动补偿方法。

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 *motorVars_M1.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。

3.3.4.5 CAN FD 命令接口

可选择将用于发送命令和接收调试信息的 CAN FD 接口添加到工程中。该接口不仅演示了 C2000 器件上 MCAN 模块的使用，而且在不方便连接基于 JTAG 的调试探针的情况下，可用于进行工程调试。

您需要将参考设计电路板上的 CAN FD 接口连接到外部器件，使参考设计应用与该器件通信。例如，您可以使用 CAN FD 转 USB 适配器从 PC 发送和接收帧。您还可以将另一个 C2000 开发板与 CAN FD 收发器配合使用。TMS320F280039C LaunchPad™ 开发套件 (LAUNCHXL-F280039C) 具有板载 CAN 收发器，可用于通信目的。LAUNCHXL-F280039C 的 CAN FD 通信实用程序应用已在参考设计软件文件夹中提供。本节中将演示此方法。

备注

要在 CCS 中同时连接多个器件，可能需要执行额外的步骤，特别是在两个电路板使用相同类型调试探针的情况下。如需调试环境设置方面的帮助，请参阅文章利用多种 XDS 调试探针进行调试。使用 CCS 的两个实例可能是最简单的方法，一个用于主应用，另一个用于 CAN 通信实用程序。

按照节 3.2.2.1 所述的步骤，导入名为 *tidm_02012_cancom_util_<device>* 的工程。构建工程，连接到 LaunchPad 并加载应用。在 **Expressions** 视图中，添加变量 *speedSet_Hz*、*flagEnableCmd*、*flagCmdRun* 和 *canComVars*，或导入 *tidm_02012_cancom_util.txt* 文件。

如节 3.2.2.1 中所述，在工程的编译器设置中添加预定义符号 *CMD_CAN_EN*，可在主 *tidm_02012_ecompressor* 工程中添加对 CAN 命令接口的支持。构建并加载工程，并将 *canComVars* 和 *motorVars_M1.cmdCAN* 变量添加到 **Expressions** 视图中，或导入 *tidm_02012_can.txt* 中的一组变量。

运行这两个应用。如果 CAN FD 通信正在工作，您应该能够在两个器件的调试会话中看到 *canComVars.txMsgCount* 和 *canComVars.rxMsgCount* 递增。在 LaunchPad CAN 实用程序调试会话中，尝试更新 *speedSet_Hz*。您应该会在主应用中看到 *motorVars_M1.speedRef_Hz* 中反映出的更新。如果它未更新，请检查是否未设置任何电机故障标志。

要发送运行电机的命令，请在 LaunchPad 调试会话中设置 *flagCmdRun*。这将在主应用中设置 *motorVars_M1.flagEnableRunAndIdentify*。在 LaunchPad 调试会话中，您应该会看到 *canComVars* 更新为主应用发送的状态信息，如图 3-17 所示。

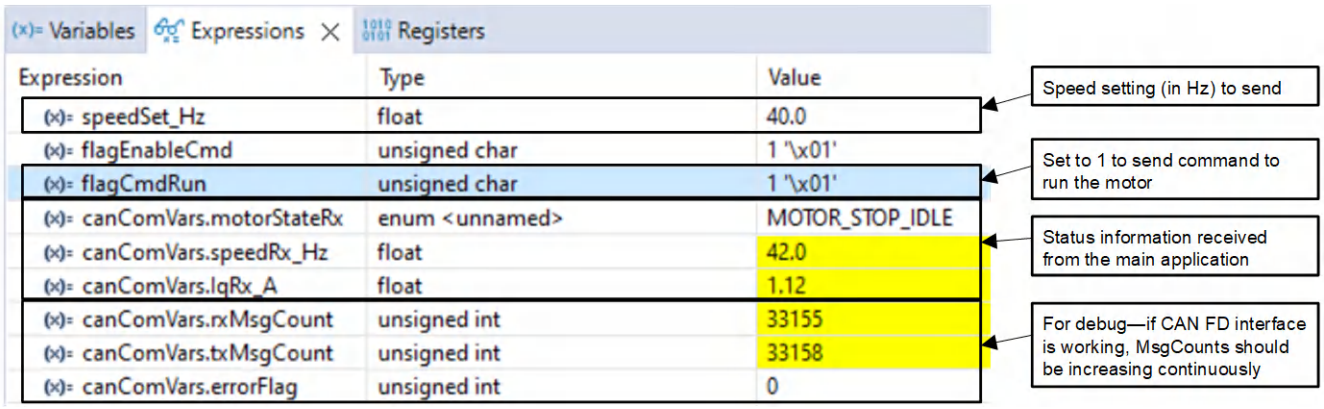


图 3-17. 表达式视图中的 CAN FD 命令实用程序变量

3.4 测试结果

使用具有 400V 直流总线输入的三相 PMSM 电机对 eCompressor 电机驱动器进行测试。实验 1-4 已经过测试，电机可以凭借 FAST 无传感器 FOC 控制平稳运行。

测试了 CAN FD 与另一个 TMS320F280039C (LaunchPAD) 的通信，包括远程控制和数据监控功能。

3.4.1 MCU CPU 负载、存储器和外设使用

表 3-2 显示了在 F280039C 上以 120MHz CPU 时钟运行参考工程时使用的 CPU 周期和 CPU 负载。这些数字基于构建级别 4，使用工程的默认设置来确定哪些函数从 RAM 运行（例如主 ISR）以及哪些函数从闪存运行。

表 3-2. 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]
CPU 利用率 (15kHz ISR)	2079	25.99	31.185

表 3-3 显示了在微控制器上运行应用所需的存储器大小。该存储器占用量基于默认工程设置。添加附加功能（如 MTPA 或振动补偿）或删除功能（例如从 *fast_full_lib.lib* 切换到 *fast_simple_lib.lib* 以删除电机识别）会导致存储器占用量发生一些变化。如表所示，存储器的很大一部分仍可用于执行其他任务。

表 3-3. 存储器使用

类型	F280039C 上已使用的存储器	F280039C 上的可用存储器	F280039C 存储器利用率
闪存	41.7 KB	384 KB	10.9%
RAM	15.3 KB	69 KB	22.2%

表 3-4 列出了此参考设计使用的外设。

表 3-4. F28003x 外设使用情况

模块	用途
ADCA、ADCB、ADCC	三相 PWM（共 6 个 PWM 通道）
EPWM1、EPWM2、EPWM3	电机电流和电压检测（共 7 个 ADC 通道）
CMPSS1、CMPSS2、CMPSS3	三相过流故障保护
EPWMXBAR TRIP7	CMPSS 输出至 EPWM 跳闸，用于过流保护
MCANA	通信
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
GPIO	一个用于 controlCARD LED D2，一个用于 DISABLE_FET_SUPPLY 信号

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-02012 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-02012 中的设计文件。

4.2 工具与软件

工具

TMDSCNCD280039C	TMS320F280039C 评估模块 controlCARD 是一款适用于 F28003x 器件系列的低成本评估和开发板。它用作此设计的子卡，通过标准 180 引脚 controlCARD HSEC 接口连接到参考电路板。
LAUNCHXL-F280039C	LAUNCHXL-F280039C 是适用于 F28003x 器件的低成本开发板。作为庞大的 TI MCU LaunchPad™ 生态系统的一部分，该器件还与各种插件模块交叉兼容。在此参考设计中，它是有用的 CAN FD 调试工具，因为它有一个板载 CAN 收发器。

软件

C2000WARE-MOTORCONTROL-SDK	MotorControl SDK 是一套软件、工具和文档，专为基于 C2000 实时控制器的电机控制系统开发而设计。此参考设计的软件可在 v4.01.00.00 或更高版本中找到。
CCSSTUDIO	构建和运行参考设计软件工程需要 Code Composer Studio 集成开发环境。建议为此工程使用 CCS v12.00.00 或更高版本。

4.3 文档支持

1. 德州仪器 (TI)，*TMS320F28003x 实时微控制器* 技术参考手册
2. 德州仪器 (TI)，*TMS320F28003x 实时微控制器* 数据表
3. 德州仪器 (TI)，*TMS320F280039C controlCARD 信息指南*
4. 德州仪器 (TI)，*InstaSPIN-FOC™ 和 InstaSPIN-MOTION™ 用户指南*
5. 德州仪器 (TI)，*使用单一直流链路分流器的 PMSM 无传感器 FOC*
6. 德州仪器 (TI)，*Motor Control SDK 通用工程和实验用户指南*

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

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