

基于MIMXRT10xx EVK的PMSM 磁场定向控制

原文链接: <https://www.nxp.com/docs/en/application-note/AN12214.pdf>

1 概述

本应用笔记描述了用于三相永磁同步电动机（PMSM）的有/无速度与位置传感器的电机控制软件的实现，包括基于恩智浦i.MX RT10xx处理器的MIMXRT10xx评估板（EVK）所使用的电动机参数识别算法。

恩智浦Freedom开发板FRDM-MC-LVPMSM被用作PMSM控制参考解决方案的硬件平台，还解决了电机控制软件的硬件相关部分，包括详细的外围设备设置和驱动程序说明。电机参数识别理论和算法也在本文中进行了描述。

本文最后一部分介绍并说明了基于FreeMASTER运行时调试工具的“电机控制应用优化（MCAT）”页面所表示的用户界面。这些工具代表了电动机参数识别、算法调整、软件控制、调试和诊断的一种简单易用的方法。

目录

1	概述.....	1
2	开发平台.....	2
3	MCU功能与外设设置.....	2
3.1	i.MX RT1050 EVK/i.MX RT1060 EVK.....	2
3.2	i.MX RT1020 EVK.....	7
3.3	CPU负荷和内存使用.....	12
4	电机控制外设的初始化.....	14
5	调整和控制应用.....	16
5.1	PMSM参数识别.....	18
5.2	使用MCAT 23的PMSM传感器/无传感器应用程序控制和调整.....	23
6	结论.....	40
7	缩略语和缩写.....	40
8	参考文献.....	41

2 开发平台

FRDM-MC-LVPMSM开发平台的电源输入电压为24-48 VDC，带有反极性保护电路。创建5.5 VDC的辅助电源来为FRDM MCU板供电。输出电流高达5 A RMS。逆变器本身由三相桥式逆变器（6-MOSFET）和三相MOSFET栅极驱动器实现。板上可以检测到模拟量（例如三相电动机电流，直流母线电压和直流母线电流）。还有一个用于速度和位置传感器的接口（霍尔编码器）。完整的NXP Freedom电机控制开发套件的框图如图1所示。

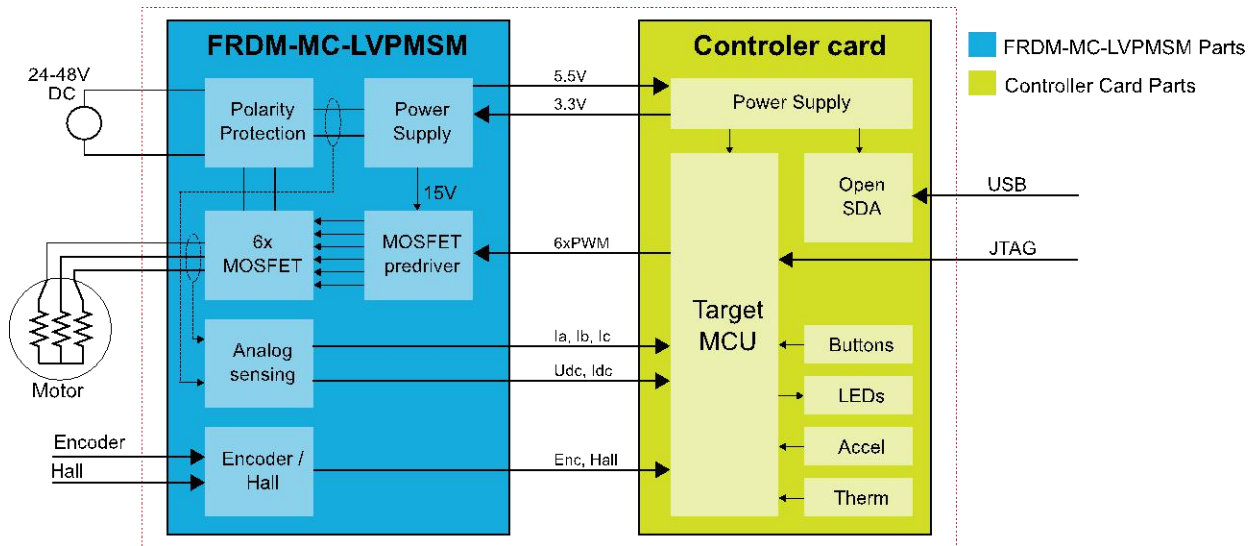


图1.电机控制开发平台框图

FRDM-MC-LVPMSM不需要复杂的设置，仅需要通过电线将所有信号从i.MX RT10xx EVK连接到FRDM-MC-LVPMSM。有关更多详细信息，请参见用户指南。有关恩智浦Freedom平台的更多信息，请访问www.nxp.com/freedom。

3 MCU功能与外设设置

本章介绍外围设备设置和应用时序。i.MX RT10xx是一个新处理器系列，采用了恩智浦ARM®Cortex®-7内核的高级实现，在i.MXRT1050 / 1060中的最高运行速度为600 MHz，在i.MXRT1020中的最高运行速度为500 MHz。

3.1 i.MX RT1050 EVK/i.MX RT1060 EVK

MIMXRT1050 EVK和MIMXRT1060 EVK开发板是旨在以小巧、低成本的封装展示i.MX RT10XX处理器最常用功能的平台。MIMXRT1050和MIMXRT1060 EVK是入门级开发板，可以帮助您熟悉处理器，然后再在更具体的设计上投入大量资源。EVK板提供各种类型的内存，尤其是64 Mbit Quad SPI flash和512 Mbit Hyper flash。

3.1.1 硬件定时和同步

正确和精确的时序对于电机控制应用至关重要。因此，专用于电机控制的外围设备要注意硬件层上的时序和同步。此外，可以将PWM频率设置为计算FOC算法的ADC中断（ADC ISR）频率的倍数。在本应用中，PWM频率等于FOC频率。时序图如图2所示。

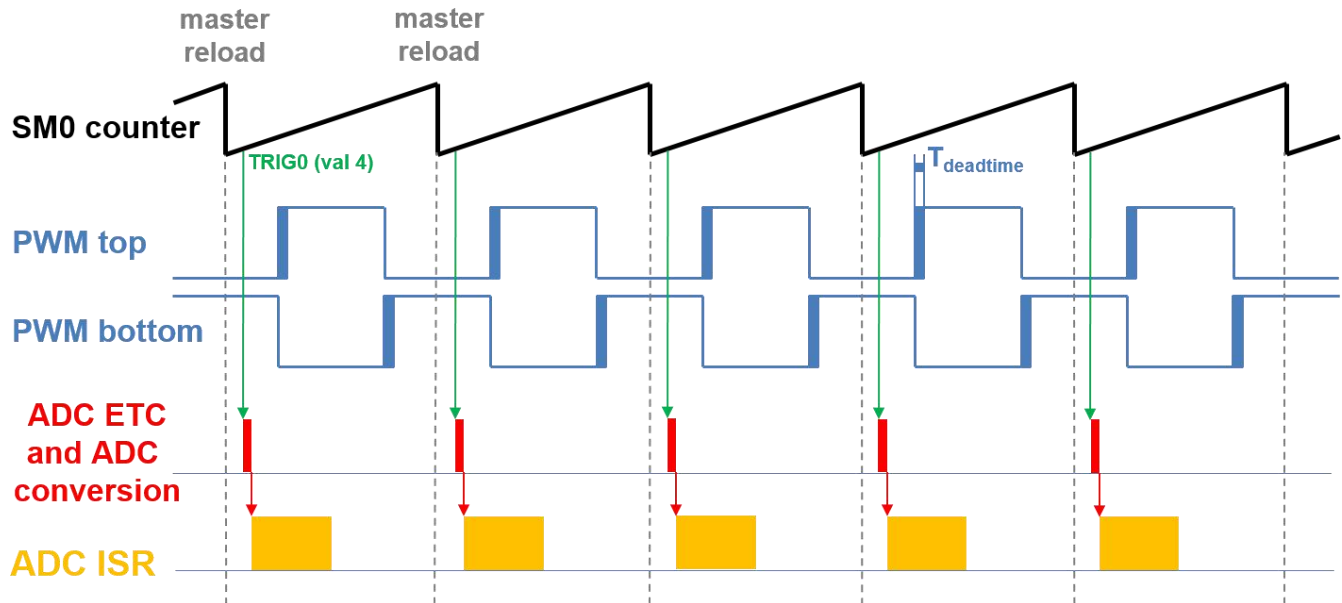


图2. i.MX RT1050/1060上的硬件时序和同步

- 顶部信号为eFlexPWM计数器（SM0计数器）。死区时间在PWM顶部和PWM底部信号处插入。SM0子模块会在每次机会时生成主重新加载。
- SM0为ADC_ETC（ADC外部触发控制）产生触发0（当计数器计数到等于TRIG4值）时，延迟约为 $T_{deadtime}/2$ 。该延迟可确保在接近100%的占空比下进行正确的电流采样。
- ADC_ETC开始ADC转换。
- ADC转换完成后，产生ADC ISR（ADC中断）。FOC计算在此中断中完成。

3.1.2 外设设置

本节介绍用于电机控制的外围设备。在i.MX RT1050/1060上，来自增强型FlexPWM（eFlexPWM）的三个子模块用于6通道PWM生成，两个12位ADC用于相电流和直流母线电压测量。eFlexPWM和ADC通过eFlexPWM的子模块0进行同步。以下设置位于 `mcdrv_evkbimxrt1050.c` (`mcdrv_evkimxrt1060.c`)和 `board.c` 文件及其头文件中。

3.1.2.1 时钟控制模块 (CCM)

CCM生成并控制设计中各个模块的时钟，并管理低功耗模式。该模块使用可用的时钟源生成时钟根。

电机控制应用中使用的时钟源为：

- PLL2，也称为系统PLL，频率为528 MHz。
- PLL3，也称为USB1 PLL，频率为480 MHz。

ARM时钟内核的工作频率为528 MHz，时钟源为PLL2。对于此设置，将设置以下寄存器：*clock_config.c*中的CBCMR [PRE_PERIPH_CLK_SEL]，CBCDR [PERIPH_CLK_SEL]和CBCDR [AHB_PODF]。ADC，XBAR和PWM由IPG_CLK_ROOT输出提供时钟，输出频率为132 MHz。必须为此设置CBCDR [IPG_PODF]寄存器。IPG_CLK_ROOT源自AHB_CLK_ROOT。LPUART来自PLL3，频率为480 MHz除以6。

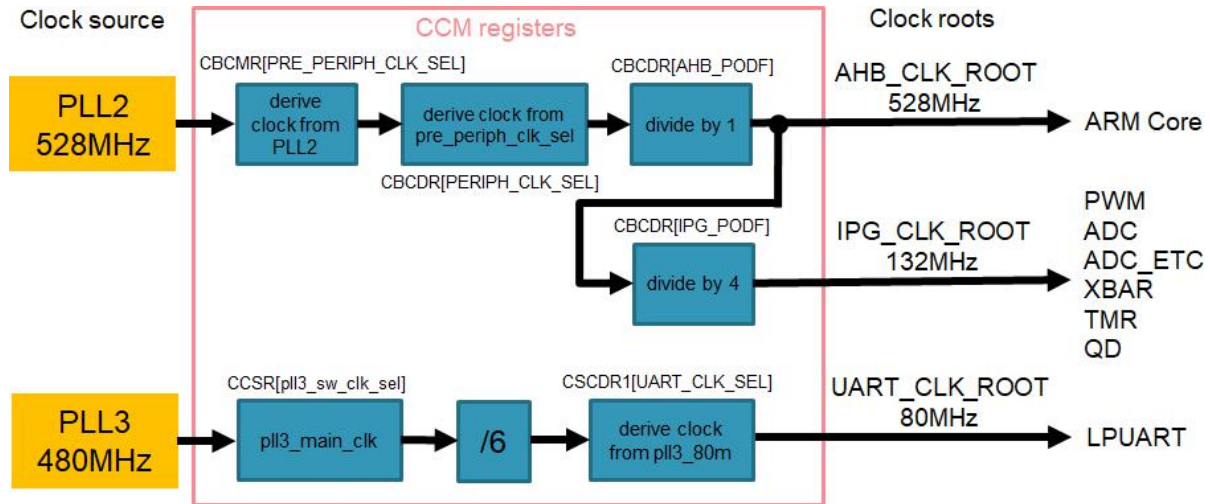


图3. i.MX RT1050/1060电机控制外设的时钟源

电动机控制所用外围设备的时钟源列在表1。

表1. i.MX RT1050/1060电机控制外设的时钟源

—	时钟源	时钟根	根时钟频率
Arm core	PLL2	AHB_CLK_ROOT	528 MHz
PWM	PLL2	IPG_CLK_ROOT	132 MHz
ADCs	PLL2	IPG_CLK_ROOT	132 MHz
ADC_ETC	PLL2	IPG_CLK_ROOT	132 MHz
XBAR	PLL2	IPG_CLK_ROOT	132 MHz
TMR	PLL2	IPG_CLK_ROOT	132 MHz
QD	PLL2	IPG_CLK_ROOT	132 MHz
LPUART	PLL3	UART_CLK_ROOT	80 MHz

有关更多详细信息，请参阅i.MX RT1050或i.MX RT1060参考手册。

3.1.2.2 PWM生成—PWM1

- eFlexPWM的时钟源为132-MHz IPG_CLK_ROOT。
- 三个子模块的六个通道用于生成三相PWM。子模块0根据用户定义的宏M1_FOC_FREQ_VS_PWM_FREQ，在事件发生时每n次机会生成一次主重新加载，
- 子模块1和3从子模块0获得时钟。
- 子模块1和3的计数器与子模块0的主重载信号同步（未使用子模块2）。
- 子模块0用于与ADC_ETC同步。当计数器计数到VAL4时，子模块在PWM重载后生成生成触发信号。
- 通过自动故障清除功能为子模块0、1、3的通道A和B启用了故障模式（故障输入恢复为零后，在第一个PWM重载时重新启用PWM输出）。
- PWM周期（频率）由计数器从INIT到VAL1计数所需的时间确定。默认情况下， $INIT = -MODULO / 2 = -6600$ ， $VAL1 = MODULO / 2 - 1 = 6599$ 。eFlexPWM时钟为132 MHz，因此周期为0.0001 s（10 kHz）。
- 启用了死区时间插入。死区时间长度由用户在M1_PWM_DEADTIME宏中定义。

3.1.2.3 ADC外部触发控制—ADC_ETC

ADC_ETC模块使多个用户能够以时分复用（TDM）方式共享ADC模块。外部触发器可以从Cross BAR（XBAR）或其他来源获得。ADC扫描通过ADC_ETC开始。

- 两个ADC都设置了自己的触发链。
- 触发链长度设置为2。启用背靠背ADC触发模式。
- 同步模式已打开。在SyncMode中，ADC1和ADC2由相同的触发源控制。触发源是PWM子模块0。

3.1.2.4 模拟感应—ADC1和ADC2

ADC1和ADC2用于电机控制相电流和直流母线电压的采样。

- ADC1和ADC2的时钟频率为66 MHz。它取自IPG_CLK_ROOT并除以2。
- ADC以10位工作，选择单端转换和硬件触发。eFlexPWM生成的触发从ADC_ETC触发ADC。
- 使能转换完成中断，并于ADC1完成最后一次采样后执行FOC快速环算法。

3.1.2.5 正交解码器模块—QD

模块用于检测编码器传感器的位置和速度

- 计数的方向由用户在M1_POSPE_ENC_DIRECTION宏中设置。
- 启用了模块计数和模块计数翻转/递减到增量/减量转数计数器。

3.1.2.6 i.MX RT1050的外设互联—XBARA1

交叉互联模块用于将触发信号从PWM互联到ADC_ETC，并将编码器（连接在GPIO）连接到QD。

- FLEXPWM1_PWM1_OUT_TRIG0_1输出触发器（由子模块0生成）连接到ADC_ETC_XBAR0_TRIG0。
- 编码器信号A相-IOMUX_XBAR_INOUT14输出分配给ENC1_PHASE_A_INPUT（在pinmux.c中将GPIO_AD_B0_00配置为XBARA1_INOUT14）。
- 编码器信号B相-IOMUX_XBAR_INOUT15输出分配给ENC1_PHASE_B_INPUT（在pinmux.c中将GPIO_AD_B0_01配置为XBARA1_INOUT15）。

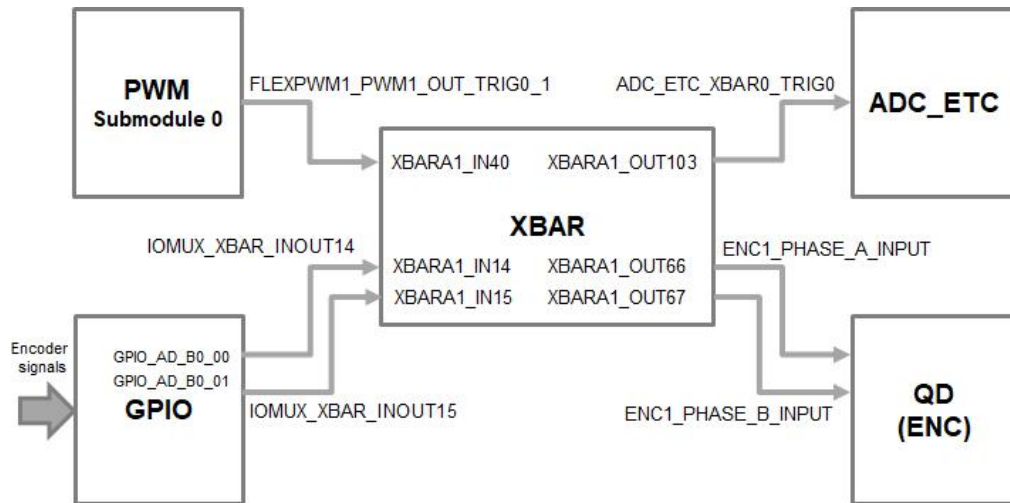


图4. i.MX RT1050的交叉互联

3.1.2.7 i.MX RT1060的外设互联—XBARA1

交叉开关矩阵用于将触发信号从PWM互连到ADC_ETC，并将编码器（连接在GPIO）连接到QD。

- FLEXPWM1_PWM1_OUT_TRIG0_1输出触发器（由子模块0生成）连接到ADC_ETC_XBAR0_TRIG0。
- 编码器信号A相-IOMUX_XBAR_INOUT16输出分配给ENC1_PHASE_A_INPUT（在pinmux.c中将GPIO_AD_B0_02配置为XBARA1_INOUT16）。
- 编码器信号B相-IOMUX_XBAR_INOUT17输出分配给ENC1_PHASE_B_INPUT（在pinmux.c中将GPIO_AD_B0_03配置为XBARA1_INOUT17）。

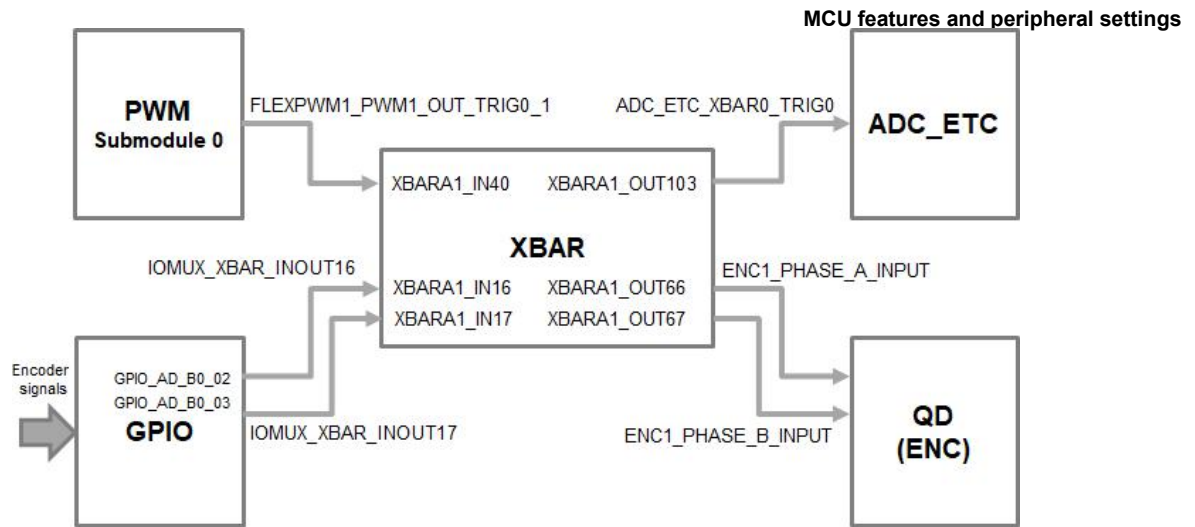


图5. i.MX RT1060的交叉互联

3.1.2.8 慢循环中断的产生—TMR1

QuadTimer模块TMR1用于生成慢循环中断。

- QuadTimerTMR1由IPG CLK ROOT除以16计时，因此TMR1的时钟频率为8.25 MHz。
- 慢循环通常比快循环慢十倍。因此，在计数器从CNTR0 = 0到COMP1 = IPG CLK ROOT / (16U * 速度环频率) 计数后中断产生。速度环频率在M1_SPEED_LOOP_FREQ宏中设置，等于1000 Hz。
- 在重载事件中启用并生成了一个中断（服务于慢循环周期）。

3.1.2.9 FreeMASTER通信—LPUART0

LPUART0（低功耗通用异步接收器和发送器）用于MCU板与PC之间的FreeMASTER通信。

- 波特率设置为115200 bit/s。
- 接收器和发送器均已启用。
- 其他设置设为默认值。

3.2 i.MX RT1020 EVK

i.MX RT1020 EVK是2层低成本通孔USB供电的PCB。它的核心是采用LQFP144封装的i.MX RT1020交叉处理器。

i.MX RT1020通过以低成本LQFP封装提供高性能功能集，扩展了i.MX RT交叉处理器系列，进一步简化了客户的电路板设计和布局。i.MX RT1020包含Arm Cortex-M7内核，该内核以高达500 MHz的速度运行，以提供较高的CPU性能和最佳的实时响应。

3.2.1 硬件定时和同步

正确和精确的时序对于电机控制应用至关重要。因此，专用于电机控制的外围设备要注意硬件层上的时序和同步。此外，可以将PWM频率设置为计算FOC算法的ADC中断（ADC ISR）频率的倍数。在这种情况下，PWM频率等于FOC频率。时序图如图6所示。

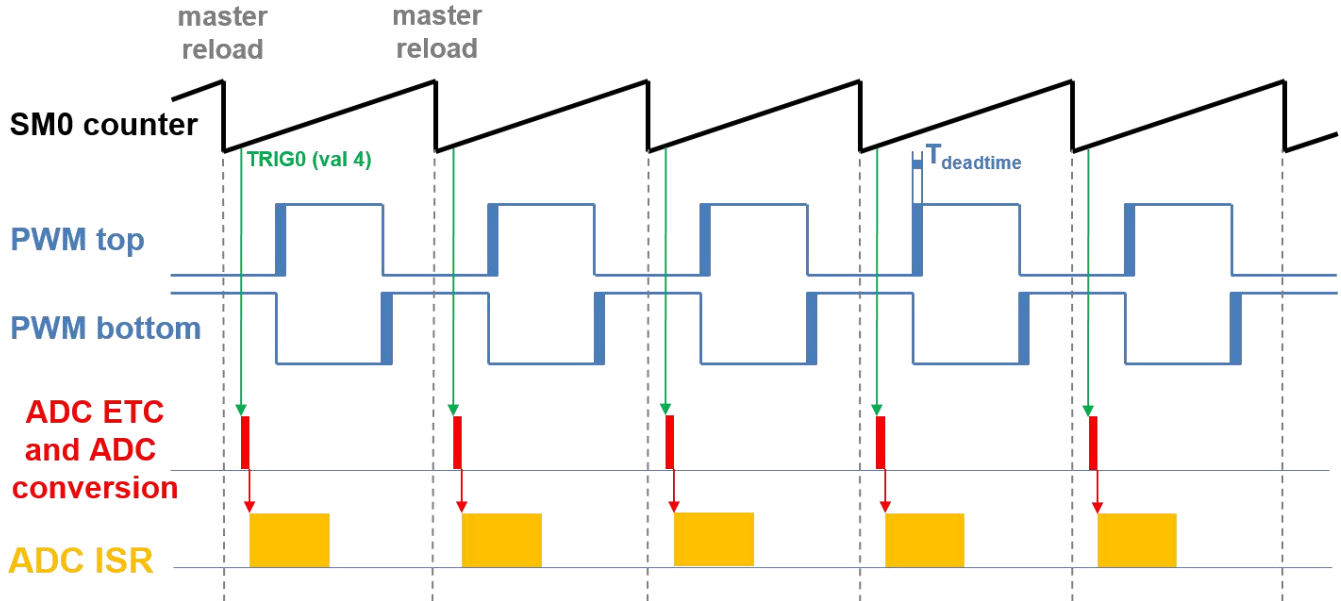


图6. i.MX RT1020上的硬件时序和同步

- 顶部信号显示eFlexPWM计数器（SM0计数器）。死区时间在PWM顶部和PWM底部信号处得到强调。SM0子模块会在每次机会时生成主重新加载。
- SM0为ADC_ETC（ADC外部触发控制）产生触发0（当计数器计数到等于TRIG4值）时，延迟约为 $T_{\text{deadtime}}/2$ 。该延迟可确保在接近100%的占空比下进行正确的电流采样。
- ADC_ETC开始ADC转换。
- ADC转换完成后，将输入ADC ISR（ADC中断）。FOC计算在此中断中完成。

3.2.2 外设设置

本节介绍用于电机控制的外围设备。在i.MX RT1020上，来自增强型FlexPWM（eFlexPWM）的三个子模块用于6通道PWM生成，两个12位ADC用于相电流和DC总线电压测量。eFlexPWM和ADC通过eFlexPWM的子模块0进行同步。以下设置位于 *mcdrv_imxrt1020.c* 和 *board.c* 文件及其头文件中。

3.2.2.1 时钟控制模块(CCM)

CCM生成并控制设计中各个模块的时钟，并管理低功耗模式。该模块使用可用的时钟源生成时钟根。

电机控制应用中使用的时钟源为：

- PLL6，也称为系统PLL，频率为500 MHz。
- PLL3，也称为USB1 PLL，频率为480 MHz。

Arm时钟内核的工作频率为500 MHz，时钟源为PLL6。对于此设置，将设置以下寄存器：*clock_config.c*中的CBCMR[PRE_PERIPH_CLK_SEL], CBCDR[PERIPH_CLK_SEL], and CBCDR[AHB_PODF]。ADC, XBAR和PWM由IPG_CLK_ROOT输出提供时钟，输出频率为125 MHz。必须为此设置CBCDR [IPG_PODF]寄存器。IPG_CLK_ROOT源自AHB_CLK_ROOT。LPUART来自PLL3，频率为480 MHz除以6。

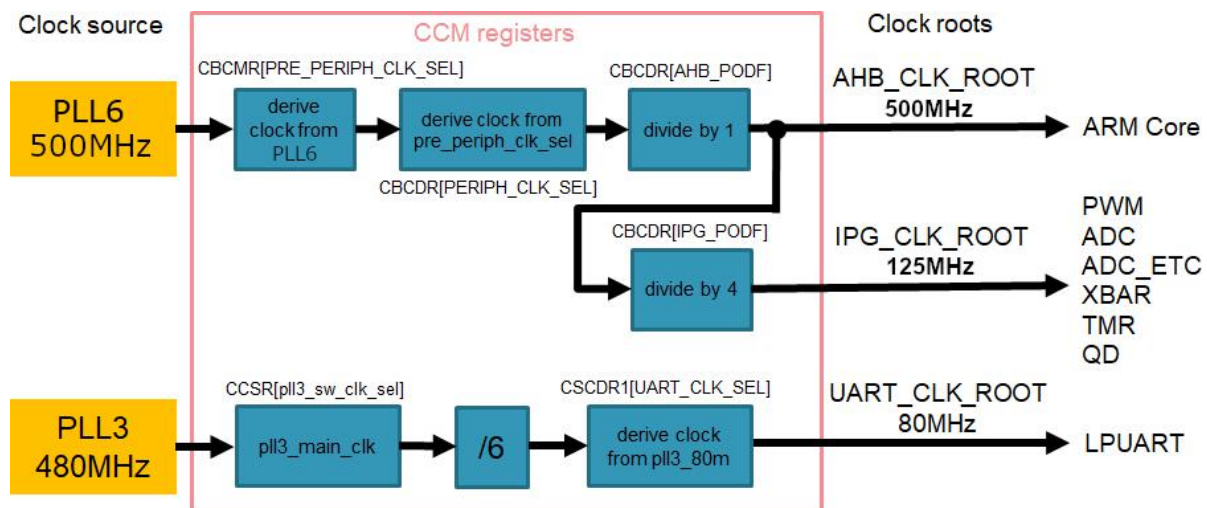


图7. i.MX RT1020用于电机控制外设的时钟源

电机控制所用外设的时钟源列在表2.

表2. i.MX RT1020用于电机控制所用外设的时钟源

—	时钟源	时钟根	时钟根频率
Arm core	PLL6	AHB_CLK_ROOT	500 MHz
PWM	PLL6	IPG_CLK_ROOT	125 MHz
ADCs	PLL6	IPG_CLK_ROOT	125 MHz
ADC_ETC	PLL6	IPG_CLK_ROOT	125 MHz
XBAR	PLL6	IPG_CLK_ROOT	125 MHz
TMR	PLL6	IPG_CLK_ROOT	125 MHz
QD	PLL6	IPG_CLK_ROOT	125 MHz
LPUART	PLL3	UART_CLK_ROOT	80 MHz

更多细节参见*i.MX RT1020 Processor Reference Manual* (文档[IMXRT1020RM](#))。

3.2.2.2 PWM子代—PWM2

- eFlexPWM的时钟根为125-MHz IPG_CLK_ROOT。
- 三个子模块的六个通道用于生成三相PWM。子模块0根据用户定义的宏M1_FOC_FREQ_VS_PWM_FREQ，在事件发生时每n次机会生成一次主设备重新加载。
- 子模块1和3从子模块0获得时钟。
- 子模块1和3的计数器与子模块0的主重载信号同步（未使用子模块2）。
- 子模块0用于与ADC_ETC同步。当计数器计数到VAL4时，子模块在PWM重载后生成输出触发。
- 通过自动故障清除功能为子模块0、1、3的通道A和B启用了故障模式（故障输入恢复为零后，在第一个PWM重载时重新启用了PWM输出）。
- PWM周期（频率）由计数器从INIT到VAL1计数所需的时间确定。默认情况下，INIT = -MODULO / 2 = -6250，VAL1 = MODULO / 2 - 1 = 6249。eFlexPWM时钟为125 MHz，因此需要0.0001 s（10 kHz）。
- 启用了死区时间插入。死区时间长度由用户在M1_PWM_DEADTIME宏中定义。

3.2.2.3 ADC外部触发控制—ADC_ETC

ADC_ETC模块使多个用户能够以时分复用（TDM）方式共享ADC模块。外部触发器可以从Cross BAR（XBAR）或其他来源获得。ADC扫描通过ADC_ETC开始。

- 两个ADC都设置了自己的触发链。
- 触发链长度设置为2。启用背靠背ADC触发模式。
- 同步模式已打开。在SyncMod中，ADC1和ADC2由相同的触发源控制。触发源是PWM子模

块0。

3.2.2.4 模拟感应—ADC1 and ADC2

ADC1和ADC2用于电流和DC总线电压的MC模拟感应。

- ADC1和ADC2的时钟频率为62.5 MHz。它取自IPG_CLK_ROOT并除以2。
- ADC以10位工作，选择单端转换和硬件触发。eFlexPWM生成的触发从ADC_ETC触发ADC。
- 使能转换完成中断，并服务于ADC1完成最后一次扫描后生成的FOC快速循环算法。

3.2.2.5 正交解码器模块—QD

QD模块用于从编码器传感器感应位置和速度。

- 启用了转换完成中断（该中断服务于ADC1完成最后一次扫描后生成的FOC快速循环算法）。
- 计数的方向由用户在M1_POSPE_ENC_DIRECTION宏中设置。
- 启用了模数计数和模数计数翻转/递减到增量/减量转数计数器。

3.2.2.6 外设互联—XBARA1

交叉开关矩阵用于将触发信号从PWM互联到ADC_ETC，并将编码器（连接在GPIO）连接到QD。

- FLEXPWM1_PWM1_OUT_TRIG0_1输出触发器（由子模块0生成）连接到ADC_ETC_XBAR0_TRIG0。
- 编码器信号A相-IOMUX_XBAR_INOUT14输出分配给ENC1_PHASE_A_INPUT（在pinmux.c中将GPIO_AD_B1_08配置为XBARA1_INOUT12）。
- 编码器信号B相-IOMUX_XBAR_INOUT15输出分配给ENC1_PHASE_B_INPUT（在pinmux.c中将GPIO_AD_B1_09配置为XBARA1_INOUT13）。

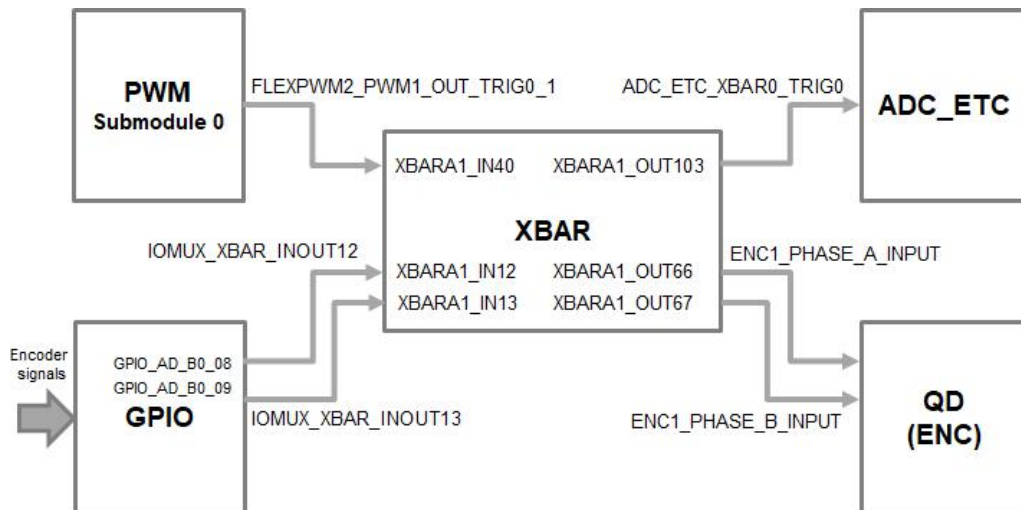


图8. i.MX RT1020的交叉互联

3.2.2.7 慢循环中断的产生—TMR1

QuadTimer模块TMR1用于生成慢循环中断。

- QuadTimerTMR1由IPG CLK ROOT除以16计时，因此TMR1的时钟频率为7.8125 MHz。
- 慢循环通常比快循环慢十倍。因此，在计数器从CNTR0 = 0到COMP1 = IPG CLK ROOT / (16U *速度环频率) 计数后中断产生。速度环频率在M1_SPEED_LOOP_FREQ宏中设置，等于1000 Hz。
- 在重载事件中启用并生成了一个中断（服务于慢循环周期）。

3.2.2.8 FreeMASTER通信—LPUART0

LPUART0（低功耗通用异步接收器和发送器）用于MCU板与PC之间的FreeMASTER通信。

- 波特率设置为115200 bit/s。
- 接收器和发送器均已启用。
- 其他设置设为默认值。

3.3 CPU负载和内存使用

以下信息适用于在调试RAM和发布FLASH配置中使用MCUXpresso IDE构建的应用程序。下列各表显示了内存使用情况和CPU负载。内存使用量是从链接器.map文件计算得出的，该文件包括在RAM中分配的2 KB FreeMASTER记录器缓冲区。CPU负载通过SysTick计时器测量。CPU负载取决于快速循环（FOC计算）和慢速循环（速度循环）的频率。在这种情况下，它适用于10 kHz的快速环路频率和1 kHz的慢速环路频率。使用以下等式计算CPU总负载：

$$CPU_{fast} = cycles_{fast} \cdot \frac{f_{fast}}{f_{CPU}} \cdot 100 [\%] \quad Eq. 1$$

$$CPU_{slow} = cycles_{slow} \cdot \frac{f_{slow}}{f_{CPU}} \cdot 100 [\%] \quad Eq. 2$$

$$CPU_{total} = CPU_{fast} + CPU_{slow} [\%] \quad Eq. 3$$

字符表示的含义是:

CPU_{fast} —快速循环占用的CPU负载 (ADC ISR)。

$cycles_{fast}$ —快速循环消耗的周期数。

f_{fast} —快速循环计算的频率 (10 kHz。 f_{CPU}

—CPU频率。

CPU_{slow} —慢循环占用的CPU负载 (TMR ISR)。 $cycles_{slow}$

—慢循环消耗的周期数。

f_{slow} —慢循环计算的频率 (1 kHz)。

CPU_{total} —电机控制占用的CPU总负载。

表3. i.MX RT1050的CPU负载和内存使用

	调试RAM		释放FLASH	
Board FLASH	-	-	51440 B	使用率0.08%
Board SRAM	-	-	-	-
SRAM_DTC	85572 B	使用率65.29%	23076 B	使用率17.61%
SRAM_ITC	-	-	10640 B	使用率8.12%
SRAM_OC	-	-	-	-
	速度控制	位置控制	速度控制	位置控制
CPU负载最大值	7.31%	7.84%	7.25%	6.84%

表 4. i.MX RT1060的CPU负载和内存使用

	调试RAM		释放FLASH	
Board FLASH	-	-	49028 B	使用率0.58%
Board SRAM	-	-	-	-
SRAM_DTC	8 KB	使用率6.25%	23060 B	使用率17.59%
SRAM_ITC	-	-	9672 B	使用率7.38%
SRAM_OC	83484 B	使用率10.62%	-	-
	速度控制	位置控制	速度控制	位置控制
CPU负载最大值	7.31%	7.84%	7.25%	6.84%

表 5. i.MX RT1020 CPU负载和内存使用

	调试RAM		释放FLASH	
Board FLASH	-	-	69376 B	使用率 0.83%
Board SRAM	-	-	-	-
SRAM_DTC	-	-	23080 B	使用率 35.22%
SRAM_ITC	-	-	-	-
SRAM_OC	88752 B	使用率 67.71%	-	-
	速度控制	位置控制	速度控制	位置控制
CPU负载最大值	7.90%	7.73%	7.70%	7.23%

4 电机控制外设的初始化

在MCU启动期间以及使用外围设备之前，通过调用MCDRV_Init_MI()函数来初始化电动机控制外围设备。所有初始化功能都在mcdrv_imxrt10xx.c源文件和mcdrv_imxrt10xx.h头文件中。这些文件中也包含用户指定的定义。该功能提供的功能是三相PWM生成和三相电流测量，以及直流母线电压和辅助量测量。Sensorless PMSM Field-Oriented Control（文档[DRM148](#)）中介绍了使用空间矢量调制（SVM）技术进行三相电流测量和PWM生成的原理。

The mcdrv_imxrt10xx.h头文件提供了几个可由用户定义的宏：

- MI_MCDRV_ADC—该宏指定使用哪个ADC外设。如果选择不受支持的外围设备，则会发出预处理器错误。
- MI_MCDRV_PWM3PH—该宏指定使用哪个PWM外设。如果选择不受支持的外围设备，则会发出预处理器错误。
- MI_MCDRV_QD_ENC—该宏指定使用哪个QD外设。如果选择不受支持的外围设备，则会发出预处理器错误。
- MI_PWM_FREQ—此定义的值设置PWM频率。
- MI_FOC_FREQ_VS_PWM_FREQ—使您能够在每个第一次、第二次、第三次或第n次PWM重载时调用快速循环中断。当PWM频率必须高于最大快速环路中断时，这很方便。
- MI_SPEED_LOOP_FREQ—此定义的值设置速度环频率（TMR1中断）。
- MI_PWM_DEADTIME—PWM死区的值，以纳秒为单位。
- MI_PWM_PAIR_PH[A..C]—这些宏可以将物理电机相位简单分配给PWM外围通道（或子模块），以此方式更改电动机相位的顺序。
- MI_ADC[1,2]_PH[A..C]—这些宏用于分配ADC通道用于相电流测量。一般规则是，在两个ADC转换器上至少必须测量一个相电流，而在不同的ADC转换器上必须测量两个剩余的相电流。其原因是要测量的相电流对的选择取决于当前的SVM扇区。如果该规则被破坏，则发出预处理器错误。有关三相电流测量的更多信息，请参见 Sensorless PMSM Field-Oriented Control（文档[DRM148](#)）。
- MI_ADC[1,2]_UDCB—此定义用于选择用于测量直流总线电压的ADC通道。

Motor control peripheral initialization

在电机控制软件中，可以使用以下API服务ADC和PWM外设：

- ADC的可用API有：
 - *mcdrv_adc_t*—MCDRV ADC结构数据类型。
 - *bool_t MI_MCDRV_ADC_PERIPH_INIT()*—默认情况下，在由*MCDRV_Init_MI()*函数调用的ADC外设初始化过程中调用此函数，并且在完成外设初始化后不应再次调用此函数。
 - *bool_t MI_MCDRV_CURR_3PH_CHAN_ASSIGN(mcdrv_adc_t*)*—调用此函数可为基于SVM扇区的下一个三相电流测量分配适当的ADC通道。该函数始终返回true。
 - *bool_t MI_MCDRV_CURR_3PH_CALIB_INIT(mcdrv_adc_t*)*—此函数初始化相电流通道偏移测量。此函数始终返回true。
 - *bool_t MI_MCDRV_CURR_3PH_CALIB(mcdrv_adc_t*)*—此函数从静止电动机的无动力相读取当前信息，并通过移动平均滤波器对其进行滤波。目的是获得测量偏移的值。默认情况下，用于移动平均滤波器的窗口的长度设置为八个样本。此函数始终返回true。
 - *bool_t MI_MCDRV_CURR_3PH_CALIB_SET(mcdrv_adc_t*)*—此函数将相电流测量偏移值声明给内部寄存器。在足够数量的*MI_MCDRV_CURR_3PH_CALIB()*调用之后，调用此函数。此函数始终返回true。
 - *bool_t MI_MCDRV_ADC_GET(mcdrv_adc_t*)*—此函数读取并计算三相电流、直流母线电压和辅助量的实际值。此函数始终返回true。
- PWM的可用API有：
 - *mcdrv_pwm3ph_t*—MCDRV PWM结构数据类型。
 - *bool_t MI_MCDRV_PWM_PERIPH_INIT*—默认情况下，在由*MCDRV_Init_MI()*函数调用的PWM外设初始化过程中调用此函数。
 - *bool_t MI_MCDRV_PWM3PH_SET(mcdrv_pwm3ph_t*)*—该函数根据*MI_MCDRV_PWMIO_DUTY*变量中存储的所需值更新PWM相位占空比。此函数始终返回true。
 - *MI_MCDRV_PWM3PH_EN(mcdrv_pwm3ph_t*)*—调用此函数将启用所有PWM通道。此函数始终返回true。
 - *bool_t MI_MCDRV_PWM3PH_DIS(mcdrv_pwm3ph_t*)*—调用此函数将禁用所有PWM通道。此函数始终返回true。
 - *bool_t MI_MCDRV_PWM3PH_FLT_GET(mcdrv_pwm3ph_t*)*—此函数返回过流故障标志的状态并自动清除这些标志（如果已设置）。当发生过流事件时，此函数返回true。否则，它返回false。
- 正交编码器的可用API有：
 - *mcdrv_qd_enc_t*—MCDRV QD结构数据类型。

- `bool_t MI_MCDRV_QD_PERIPH_INIT()`—默认情况下，在由`MCDRV_Init_MI()` 函数调用的QD外设初始化过程中调用此函数。
- `bool_t MI_MCDRV_QD_GET(mcdrv_qd_enc_t*)`—此函数返回实际位置和速度。此函数始终返回true。
- `bool_t MI_MCDRV_QD_SET_DIRECTION(mcdrv_qd_enc_t*)`—此函数设置正交编码器的方向。此函数始终返回true。
- `bool_t MI_MCDRV_QD_CLEAR(mcdrv_qd_enc_t*)`—此函数清除内部变量和解码器计数器。此函数始终返回true。

5 调整和控制应用

本节提供有关用于控制传感器/无传感器PMSM磁场定向控制（FOC）应用程序的工具和推荐过程的信息。该应用包含FreeMASTER实时调试监视器和数据可视化工具的嵌入式驱动程序，用于与PC进行通信。它支持非侵入式监视以及实时修改目标变量，这对于算法调整非常有用。除目标端驱动外，FreeMASTER工具还需要安装PC应用程序。有关更多信息请访问 www.nxp.com/freemaster。

PMSM传感器/无传感器FOC应用可以使用PMSM的电机控制应用优化（MCAT）插件轻松控制和调整。MCAT for PMSM是一个用户友好的模块化页面，可在FreeMASTER中运行，要启动它只需执行项目旁边的.pmp文件。该工具由选项卡菜单，调整模式选择器和Figure 9所示的工作区组成。选项卡菜单中的每个选项卡都代表一个子模块，您可以调整或控制应用程序的不同方面。除了PMSM的MCAT页面之外，FreeMASTER项目文件中还预定义了项目树中的多个作用域、记录器和变量，以进一步简化电动机参数的调试和调试。可以使用“Basic（基本）”和“Expert（专家）”调整模式。选择“Expert”模式将授予您修改MCAT中所有可用参数和字段的权限。

对于没有经验的用户，建议使用“Basic”模式。当FreeMASTER没有连接到目标时，“App ID”行显示为“offline”。建立使用正确软件与目标MCU的通信后，“App ID”行将显示板子名称“pmsm_evk_imxrt10xx”，并且给定MCU的所有存储参数均已加载。

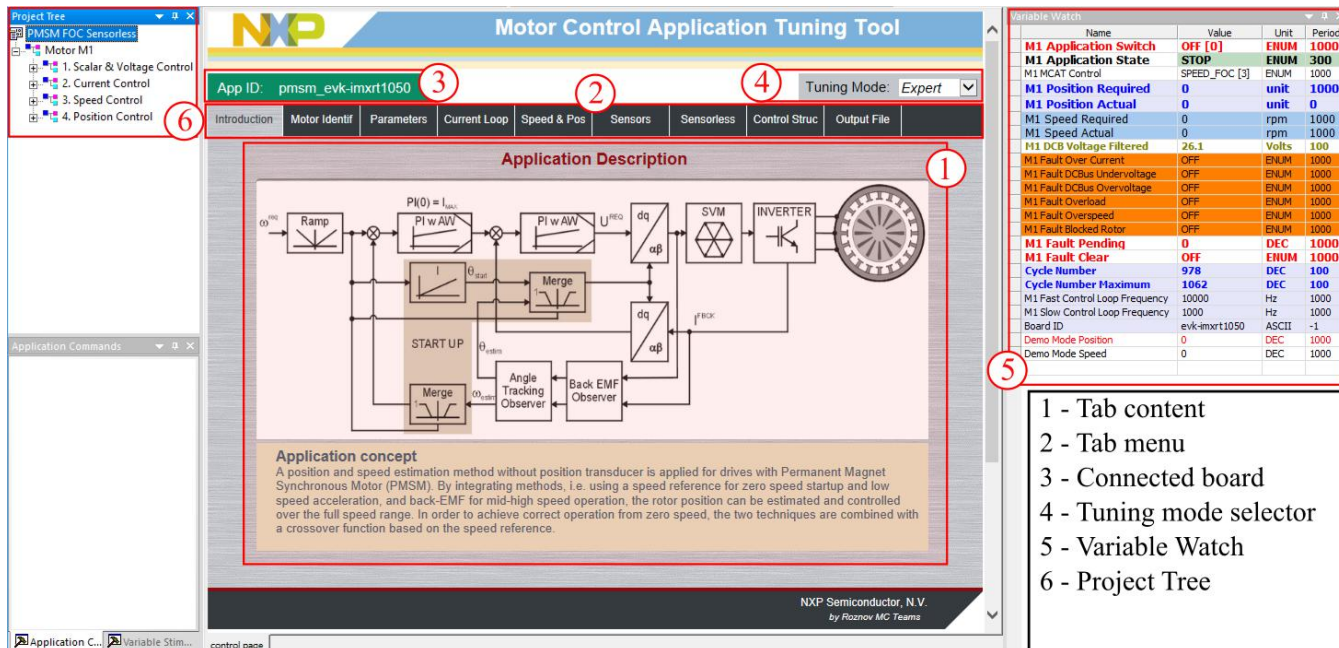


图9. MCAT界面布局

在默认配置中，以下选项卡可用：

- “Introduction（介绍）”—欢迎页面，其中包含PMSM传感器/无传感器FOC图解以及应用的简短说明。
- “Motor Identif（电机标识）”—PMSM半自动参数测量控制页面。PMSM参数识别将在本文档中进一步详细介绍。
- “Parameters（参数）”—您可以在此页面修改电动机参数、硬件和应用的规格、校准和故障极限。
- “Current Loop（电流回路）”—电流回路PI控制器的增益和输出限制。
- “Speed & Pos（速度和位置）”—此选项卡包含用于指定速度控制器比例和积分增益以及输出限制和速度斜坡参数的说明。位置比例控制器常数也在此处设置。
- “Sensors（传感器）”—此页面包含编码器参数和位置观察器参数。
- “Sensorless（无传感器）”—在此页面您可以调整BEMF观察器、跟踪观察器和开环启动的参数。
- “Control Struc（控制结构）”—在此应用程序控制页面您可以使用不同的技术（标量-伏/赫兹控制，电压FOC，电流FOC，速度FOC和位置FOC）选择和控制在PMSM。应用程序状态也显示在此选项卡上。
- “Output file（输出文件）”—此选项卡显示PMSM传感器/无传感器FOC应用程序所需的所有计算出的常数。也可以生成m1_acim_appconfig.h文件，然后在项目重建时将其用于永久预设所有应用程序参数。
- “App page（应用程序页面）”—此选项卡包含图形元素，例如速度表、DC总线电压测量条和各种开关，可实现简单快速和用户友好的应用程序控制。也可以在此页面控制故障清除和演示模式（随时间设置各种预定义的所需速度和位置）。

大多数选项卡提供了使用“更新目标”按钮立即将MCAT中指定的参数加载到目标中、并使用“重新加载数据”和“存储数据”按钮从硬盘驱动器文件中保存（或还原）它们的可能性。

以下各节提供了如何识别已连接的PMSM电动机的参数以及如何适当调整应用程序的简单说明。

5.1 PMSM 参数识别

由于PMSM驱动器的基于模型的控制方法是最有效和最实用的，因此获得准确的电动机模型是驱动器设计和控制的重要组成部分。对于已实现的FOC算法，有必要知道定子电阻 R_s ，直流电感 L_d ，正交电感 L_q 和BEMF常数 K_e 的值。

5.1.1 功率级表征

每个逆变器都会引入总误差电压 U_{error} ，该误差电压是由空载时间、电流钳位效应和晶体管压降引起的。总误差电压 U_{error} 取决于相电流 i_s ，并且在功率级表征过程中测量了这种依赖性。逆变器误差特性的示例如图10所示。功率级特性描述是MCAT的一部分，可以通过“电机标识”选项卡进行控制。为了进行表征，将电动机与已知的定子电阻 R_s 连接，并在“Calib R_s ”字段中设置该值。然后指定“校准范围”，即执行 U_{error} 测量的定子电流 i_s 的范围。按下“校准”按钮开始表征。表征过程逐步执行65个 i_{sd} 电流步长（从 $i_s = -I_{s,calib}$ 到 $i_s = I_{s,calib}$ ），每步花费300 ms，因此请注意，该过程大约需要20秒，并且电动机必须承受此负载。所获取的表征数据将保存到文件中，并随后用于 R_s 测量过程中的相电压校正。可以使用最大电流 $I_{s,calib}$ 进行以下 R_s 测量。建议使用具有低 R_s 的电动机进行表征。

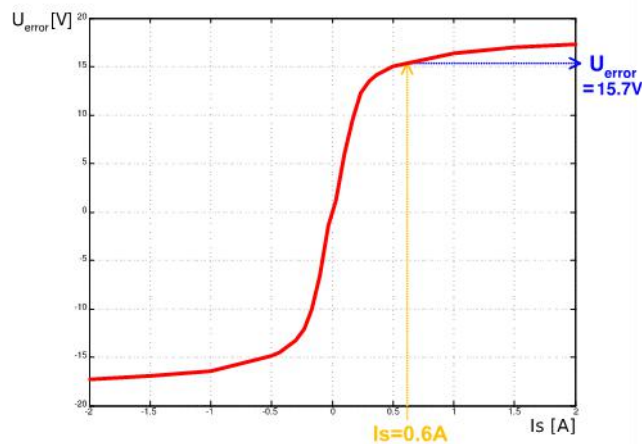


图10. 功率级表征示例

功率级表征仅对于用户硬件板需要。当恩智浦功率级（TWR、FRDM或HVP）与应用程序一起使用时，可以省略表征过程。所获取的特征数据将保存到文件中，因此对于给定的硬件仅需执行一次。

5.1.2 定子电阻测量

定子电阻 R_s 是用直流电流 I_{phN} 值测量的，该电流在电机上施加的时间为1200 ms。直流电压 U_{DC} 使用电流控制器保持。保守选择其参数以确保稳定性。定子电阻 R_s 使用欧姆定律计算为：

$$R_s = \frac{U_{DC} - U_{error}}{I_{phN}} [\Omega] \quad \text{Eq. 4}$$

5.1.3 定子电感

为了识别定子电感 L_s ，将正弦测量电压施加到电机上。在 L_s 测量期间启用电压控制。正弦电压的频率和幅度是在实际测量之前以及调谐过程中获得的。调谐过程从施加于电动机的0-V振幅和 F_{start} 起始频率开始。幅度通过 $U_d inc$ 逐渐增加，直到DC总线电压（DCbus/2）的一半，直到达到 $I_d ampl$ 。如果即使在DCbus/2且启动 F_{start} 的情况下仍未达到 $I_d ampl$ ，则将测量信号的频率逐渐降低 F_{dec} ，再降低至 F_{min} ，直到达到 $I_d ampl$ 为止。如果仍未达到 $I_d ampl$ ，则以DCbus/2和 F_{min} 继续测量。调整过程如图11所示。

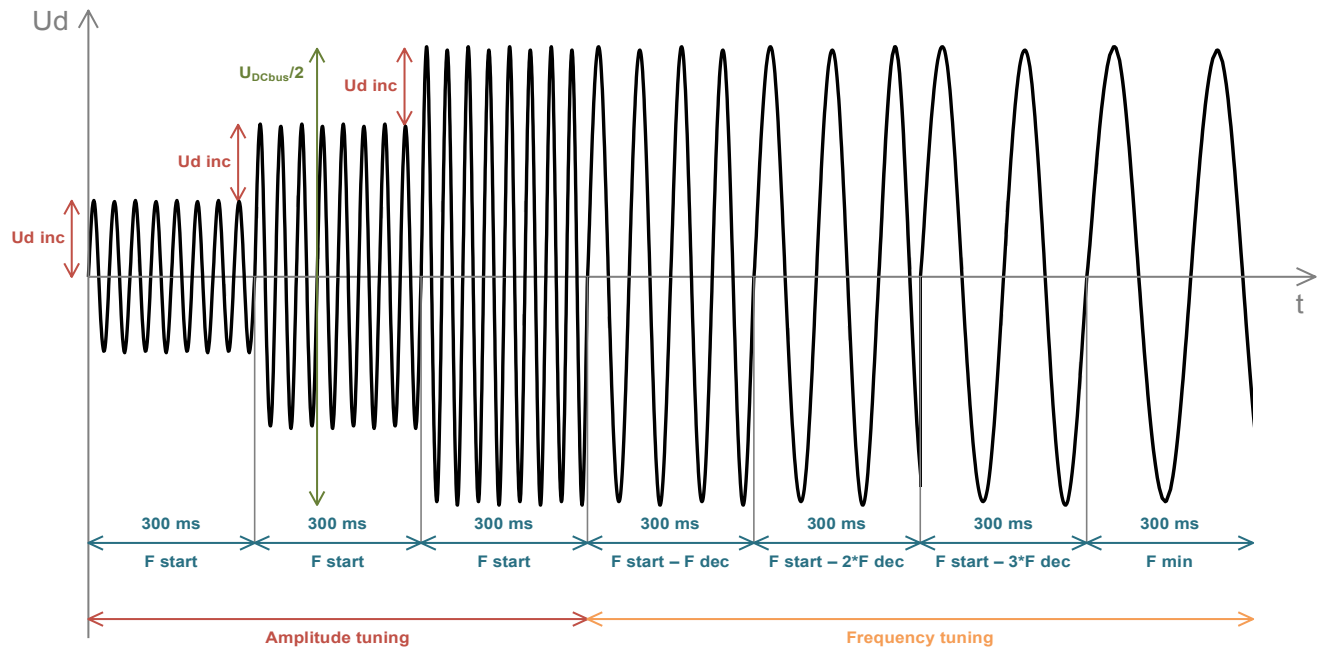


图11. 调整 L_s 测量信号

整定过程完成后，将正弦测量信号（具有在整定过程中获得的幅度和频率）加在电动机上，然后根据电压和电流幅度计算RL电路的总阻抗，并根据RL电路的总阻抗计算 L_s 。

$$Z_{RL} = \frac{U_d}{I_{d\ amp}} [\Omega] \quad \text{Eq. 5}$$

$$X_{Ls} = \sqrt{Z_{RL}^2 - R_s^2} [\Omega] \quad \text{Eq. 6}$$

$$L_s = \frac{X_{Ls}}{2\pi f} [\Omega] \quad \text{Eq. 7}$$

直流电感 L_d 和正交电感 L_q 的测量方法与 L_s 相同。在进行 L_d 和 L_q 测量之前，将直流电流施加到D轴，该轴对齐转子。对于 L_d 测量，在D轴上施加正弦电压。对于 L_q 测量，在Q轴上施加正弦电压。

5.1.4 BEMF常数测量

在实际测量BEMF常数 (K_e) 之前，MCAT工具会根据先前测量的 R_s 、 L_d 和 L_q 计算当前控制器和BEMF观测器常数。要测量 K_e ，电动机必须旋转。通过“ I_d meas”控制 I_d ，并通过对所需速度进行积分来生成电气开环位置，该速度取自 N_{nom} 。当电动机达到所需速度时，对由BEMF观测器获得的BEMF电压进行滤波并计算 K_e ：

$$K_e = \frac{U_{BEMF}}{\omega_{el}} [\Omega] \quad \text{Eq. 8}$$

在测量 K_e 时，用户必须目视检查以确定电动机是否在正常旋转。如果电动机旋转不正常，请执行以下步骤：

- 确保 pp 数正确。 K_e 测量所需的速度也由 pp 计算得出。因此， pp 的误差会导致所得 K_e 的误差。
- 在开环期间旋转时，增加 $I_{d\text{ meas}}$ 以产生更高的扭矩。
- 减小 N_{nom} 以降低 K_e 测量所需的速度。

5.1.5 极对数

没有位置传感器则无法测量极对数，但有一个简单的助手可以确定极对数 (pp)。 pp 助手的编号执行一次转动，停止几秒钟，然后重复进行。因为 pp 值是电气速度和机械速度之间的比率，所以可以将其确定为每机械旋转一圈的停止次数。建议不要在第一次机械旋转期间计算挡块，因为对齐会在第一次旋转期间发生，并且会影响挡块的数量。在 pp 测量期间，启用电流环路并将 I_d 电流控制为 $I_{d\text{ meas}}$ 。通过积分开环速度生成电气位置。如果在启动 pp 助手数量后转子仍未移动，请停止助手，增加 $I_{d\text{ meas}}$ ，然后重新启动助手。

5.1.6 机械参数测量

在速度加速度测试期间，产生转矩 T 和加载转矩 T_{load} 已知，惯性矩 J 和粘滞摩擦 B 可以通过Eq. 12确定。

$$\frac{d\omega_m}{dt} = \frac{1}{J} (T - T_{load} - B\omega_m) [\text{rad/s}^2] \quad \text{Eq. 9}$$

等式中的 ω_m 是机械速度。机械参数识别软件使用扭矩曲线，如图12所示。在整个测量过程中，为简单起见，加载扭矩为0，仅考虑摩擦力和电机产生的扭矩。

在测量的第一阶段，施加恒定转矩 T_{meas} ，并且在时间 t_1 中，电动机加速到其额定速度的50%。

这些积分是在 t_0 （速度估算足够准确）到 t_1 的时间段内计算的：

$$T_{int} = \int_{t_0}^{t_1} T dt \text{ [Nms]} \tag{Eq. 10}$$

$$\omega_{int} = \int_{t_0}^{t_1} \omega_m dt \text{ [rad/s]} \tag{Eq. 11}$$

在第二阶段中，转子仅通过摩擦而没有产生转矩地自由减速。这使您可以简单地测量机械时间常数 $\tau_m = J/B$ ，因为转子从其原始值减速的时间为63%。

最终的机械参数估算可以通过积分 Eq. 12来计算：

$$\omega_m(t_1) = \frac{1}{J} T_{int} - B\omega_{int} + \omega_m(t_0) \text{ [rad/s]} \tag{Eq. 12}$$

惯性矩为：

$$J = \frac{\tau_m T_{int}}{\tau_m [\omega_m(t_1) - \omega_m(t_0)] + \omega_{int}} \text{ [kgm}^2\text{]} \tag{Eq. 13}$$

然后从机械时间常数和惯性矩之间的关系得出粘滞摩擦。要使用机械参数测量，必须设置电流控制回路带宽 $f_{0,Current}$ ，速度控制回路带宽 $f_{0,Speed}$ 和机械参数测量转矩 Trq_m 。

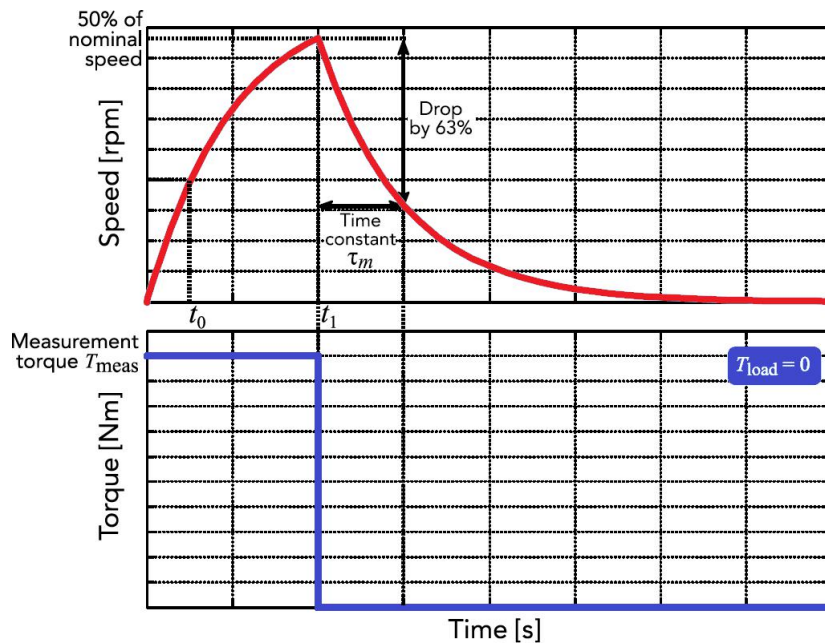


图12. PMSM识别标签

5.1.7 PMSM机电参数测量过程

电动机识别过程可以在MCAT“Motor Identif”选项卡中控制和设置，如图13所示。要测量您自己的电机，请执行以下步骤：

- 选择您的硬件板。在标准恩智浦硬件之间选择或使用您自己的硬件。如果您使用自己的硬件，请指定其标度（“Parameters”菜单选项卡中的“ I_{max} ”和“ $U_{DCB\ max}$ ”）。
- 如果您不知道电动机的极对数，请使用第5.1.5节“极对数”中所述的极对数辅助。
- 如果是第一次使用自己的硬件，请执行第5.1.1节“功率级表征”中所述的功率级表征。
- 输入电动机测量参数（取决于“Basic/Expert”模式），然后按“Measure electrical（电气测量）”或“Measure mechanical（机械测量）”按钮开始测量。您可以在“Status（状态）”栏中观察正在测量的参数。

The screenshot shows the 'Parameters Measurement' section of the NXP Motor Control Application Tuning Tool. It is divided into several sub-sections, each marked with a red circle and a number from 1 to 7. The sub-sections are:

- 1 - HW selection:** Inverter settings, HW board (User HW), Generate HW Data File.
- 2 - Power stage characterization:** Calib R_s (0.5 [Ω]), Calib range (2 [A]), Calibrate.
- 3 - Electrical measurement conditions:** Electrical parameters measurement, N nom (4000 [rpm]), I_s DC (1.8 [A]), I_s AC (0.9 [A]), Freq start (999 [Hz]), Freq min (400 [Hz]), U_d inc (0.5 [V]), Freq dec (100 [Hz]), Measure electrical.
- 4 - Mechanical measurement conditions:** Mechanical parameters measurement, f_0 Current (280 [Hz]), f_0 Speed (20 [Hz]), TRQ_m (0.05 [Nm]), Measure mechanical.
- 5 - Start/stop pp assistant:** Number of pp assistant, Start, Stop.
- 6 - Measured parameters:** Measured Motor Parameters, pp (3 [-]), R_s (0.56 [Ω]), L_d (0.000196 [H]), L_q (0.00023 [H]), K_e (0.0595 [V/Hz]), J (0.000023 [kgm²]), B (0.0002 [-]).
- 7 - Measurement status:** Status, Ready for measurement ...

At the bottom of the interface, there are buttons for 'Reload Data' and 'Store Data'. The footer of the application reads 'NXP Semiconductor, N.V. by Raznov MC Teams'.

图13. PMSM识别标签

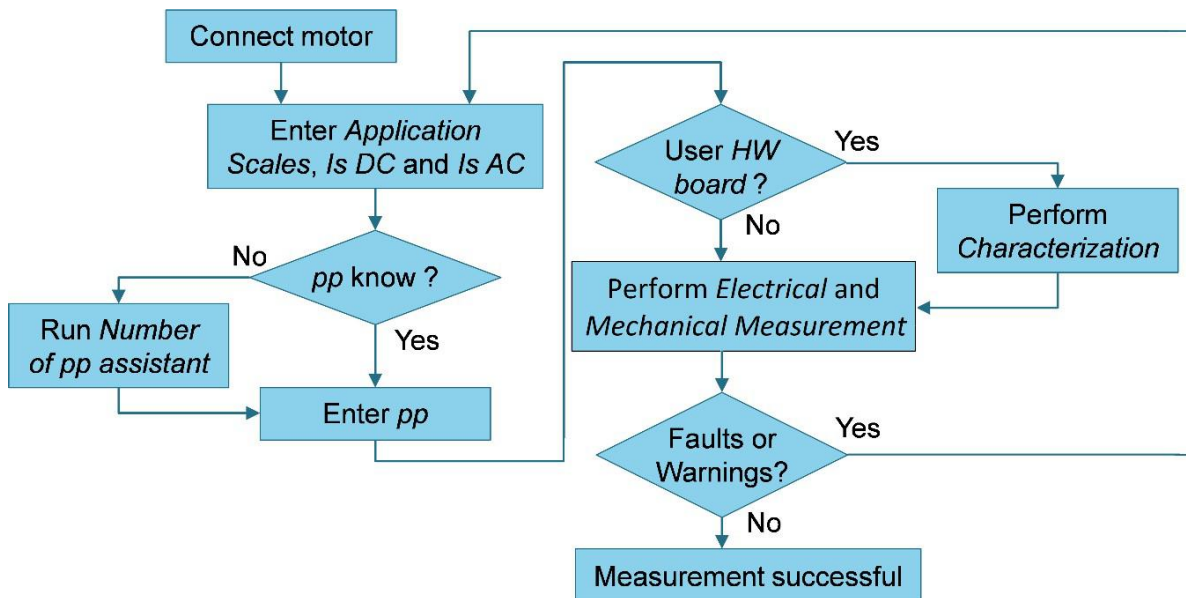


图14.测量流程图

在测量过程中，可能会发生测量故障和警告。请勿将这些故障与过流、欠压等应用故障混淆。这些故障的列表及其说明和可能的故障排除方法如表6所示。

表6. 测量故障和警告

故障序号	故障描述	故障原因	故障排除
1	电机未连接。	使用可用的直流总线电压无法达到 $I_a > 50 \text{ mA}$ 。	检查电机是否已连接。
2	R_s 太高，无法校准。	可用的直流母线电压无法达到校准要求。	使用具有较低 R_s 的电机进行功率级表征。 Use a motor with a lower for the power stage characterization.
3	未达到直流电流测量量 $I_s DC$ 。	未达到用户定义的 $I_s DC$ ，因此以较低的 $I_s DC$ 进行测量。	升高直流总线电压以达到 $I_s DC$ 或降低 $I_s DC$ 以避免此警告。
4	未达到交流电流幅值测量量 $I_s AC$ 。	未达到用户定义的 $I_s AC$ ，因此以较低的 $I_s AC$ 进行测量。	升高直流母线电压或降低 F_{min} 以达到 $I_s AC$ ，或降低 $I_s AC$ ，以避免此警告。
5	特征数据错误。	用于电压校正的特性数据与实际功率级不对应。	选择用户硬件并执行校准。
6	机械测量超时。	机械测量时间太长。	使用其他设置重复测量过程。

5.2 使用MCAT的PMSM传感器/无传感器应用程序控制和调整

FreeMASTER（已通过MCAT页面启用）可用于完全控制和轻松调整ACIM无传感器FOC应用程序。此处将更详细地描述用于PMSM子模块（第5节“调整和控制应用程序”中列出）的MCAT选项卡。

5.2.1 使用MCAT进行应用程序控制

可以通过“Control Struc”选项卡来控制该应用程序，如图15所示。屏幕左侧的状态控制区域显示了当前应用程序状态，并使您能够打开或关闭主应用程序开关（打开正在运行的应用程序关闭会禁用所有PWM输出）。“Cascade Control Structure（级联控制结构）”区域位于屏幕的右侧。在这里，您可以使用适当的按钮在标量控制和FOC控制之间进行选择。通过选择“Voltage（电压）FOC”，“Current（电流）FOC”，“Speed（速度）FOC”（传感器/无传感器）或“Position（位置）FOC”（传感器），可以启用FOC级联结构的选定部分。这对于应用程序调整和调试很有用。

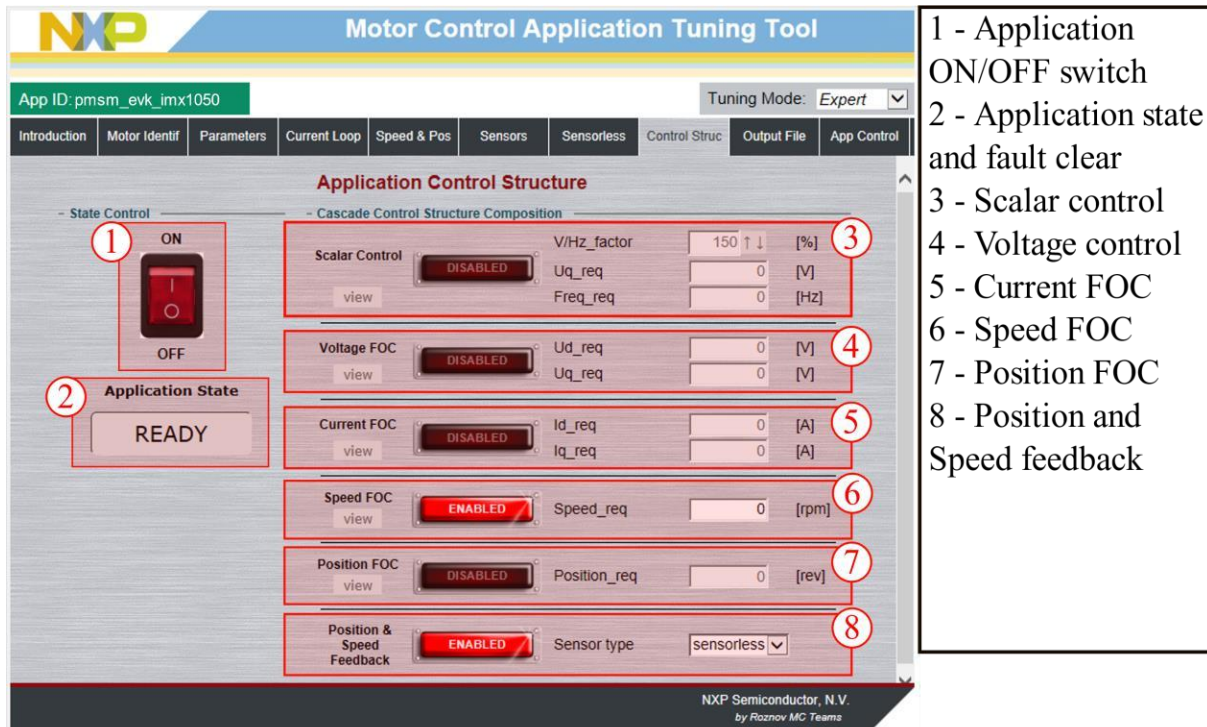


图15. PMSM控制页面的MCAT

标量控制图如图16所示。它是最简单的电动机控制技术。定子电压大小和频率之间的比率必须保持在标称比率。因此，该控制方法有时称为Volt per Hertz 或V/Hz。在“Expert”调整模式中输入所需的电压和频率时需要注意。在“Basic”模式下，该比率保持恒定，唯一需要的输入是电压和频率。即使未直接使用估算的位置信息，位置估算BEMF观测器算法和跟踪观测器算法（有关更多信息，请参阅 *Sensorless PMSM Field-Oriented Control* (document [DRM148](#))) 也在后台运行。这对于BEMF观测器调试非常有用。

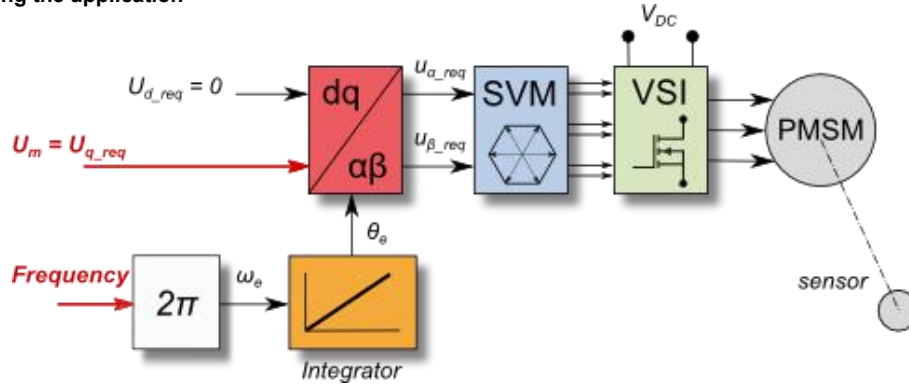


图16. 标量控制模式

电压FOC的框图如图17所示。与标量控制不同，位置反馈使用BEMF观测器关闭，并且定子电压幅值与电动机速度无关。可以使用“Ud_req”和“Uq_req”字段指定d轴和q轴定子电压。此控制方法对于BEMF观察器功能检查很有用。

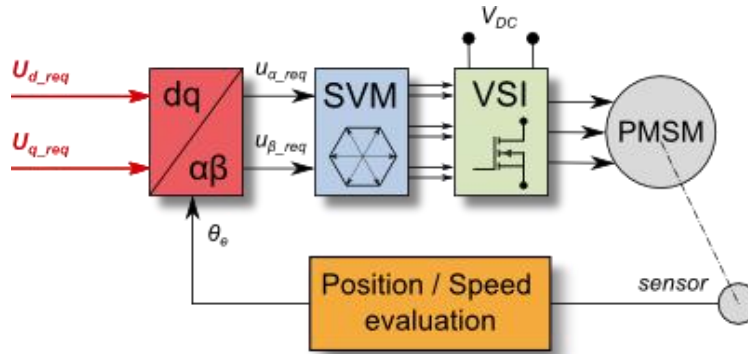


图17. 电压FOC控制模式

当前的FOC（或转矩）控制需要转子位置反馈并将电流转换为d-q参考系。如图18的框图所示，有两个参考变量（“Id_req”和“Iq_req”）可用于电动机控制。d轴电流分量 i_{sd_req} 负责转子磁通量控制。电流 i_{sq_req} 的q轴电流分量会产生转矩，并且通过施加转矩，电动机将开始运行。通过更改电流 i_{sq_req} 的极性，电动机可以更改旋转方向。假设正确调整了BEMF观测器，则可以使用当前的FOC控制结构来调整当前的PI控制器。

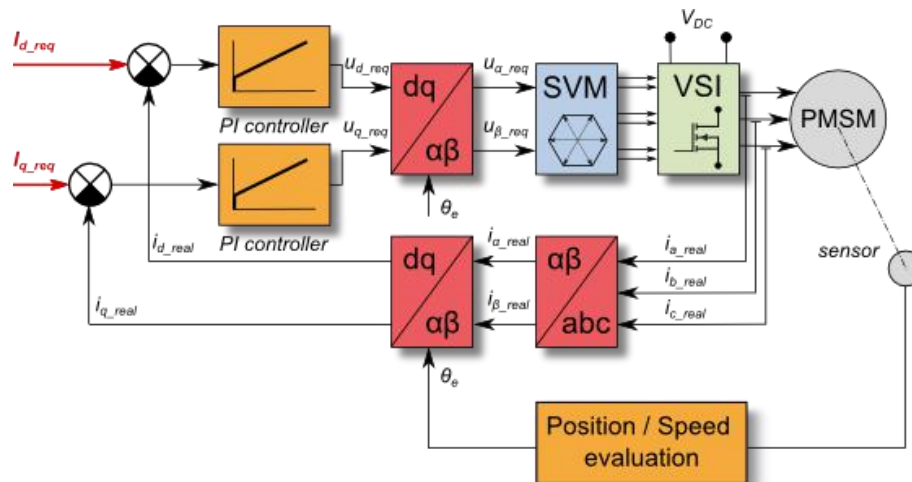


图18. 电流（转矩）控制方式

通过启用速度FOC控制结构来激活速度PMSM传感器/无传感器FOC（流程图如图19所示）。在“Speed_req”字段中输入所需的速度。在整个FOC操作期间，d轴电流参考量保持为0。

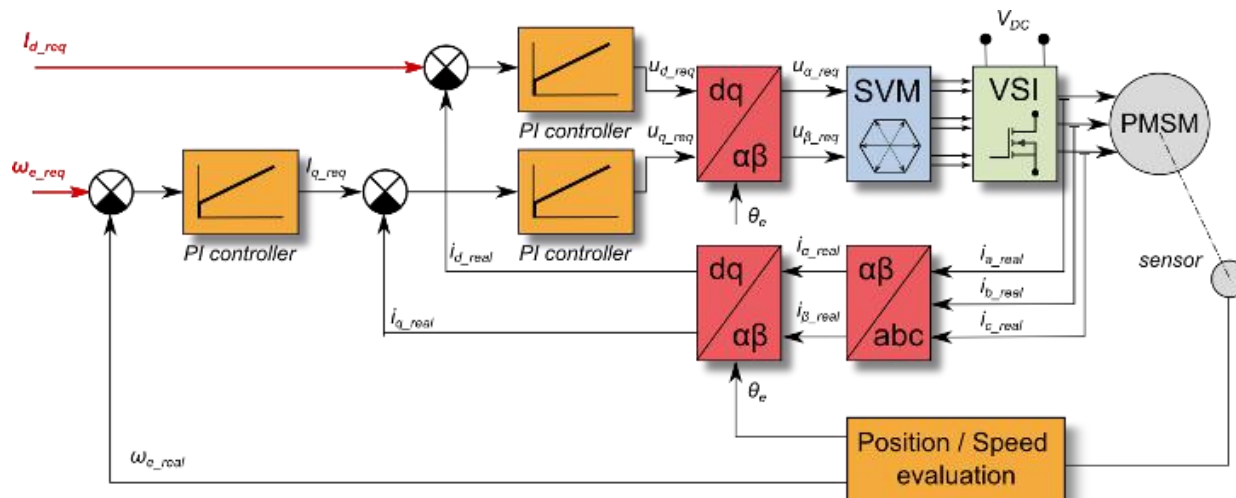


图19. 速度 FOC控制模式

位置PMSM传感器FOC如图20所示。使用P控制器的位置控制可以在“Speed & Pos（速度和位置）”菜单选项卡中进行调整。反馈需要编码器传感器。没有传感器，位置控制将不起作用。FRDM-MC-LVPMSM板上缺少制动电阻。因此，需要设置一个缓慢的速度斜坡（在“Speed & Pos”菜单选项卡中），因为在制动快速旋转的轴时，DC总线上的电压可能会升高。可能会导致过压故障。

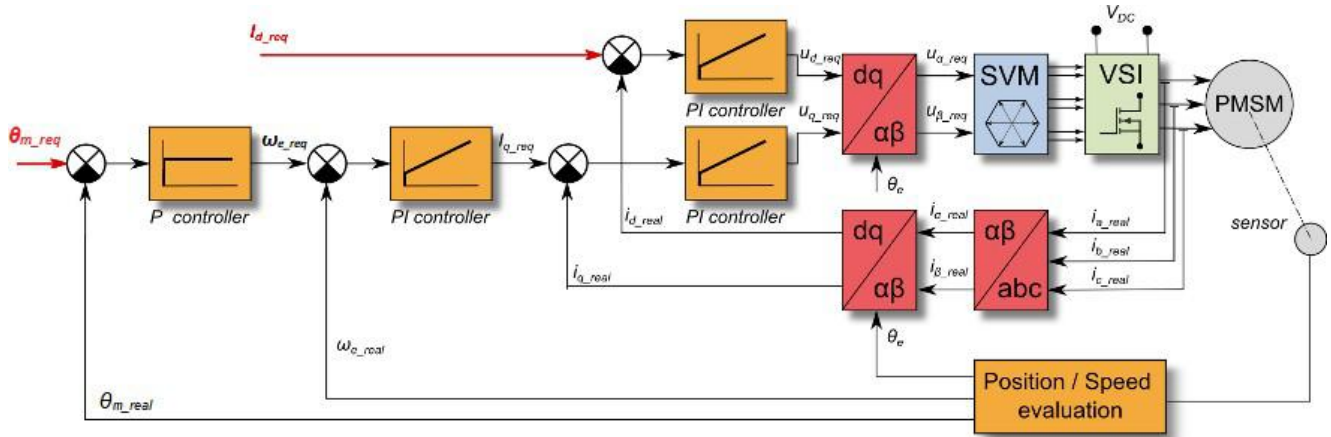


图20. 位置FOC控制模式

5.2.2 使用MCAT进行PMSM传感器/无传感器应用程序调整

本节提供有关如何分几步运行电动机的指南。强烈建议仔细执行所有步骤，以消除调试过程中可能出现的任何问题。图21所示的状态图描述了典型的PMSM传感器/无传感器控制调整过程。以下各节将更详细地描述每个调整阶段。

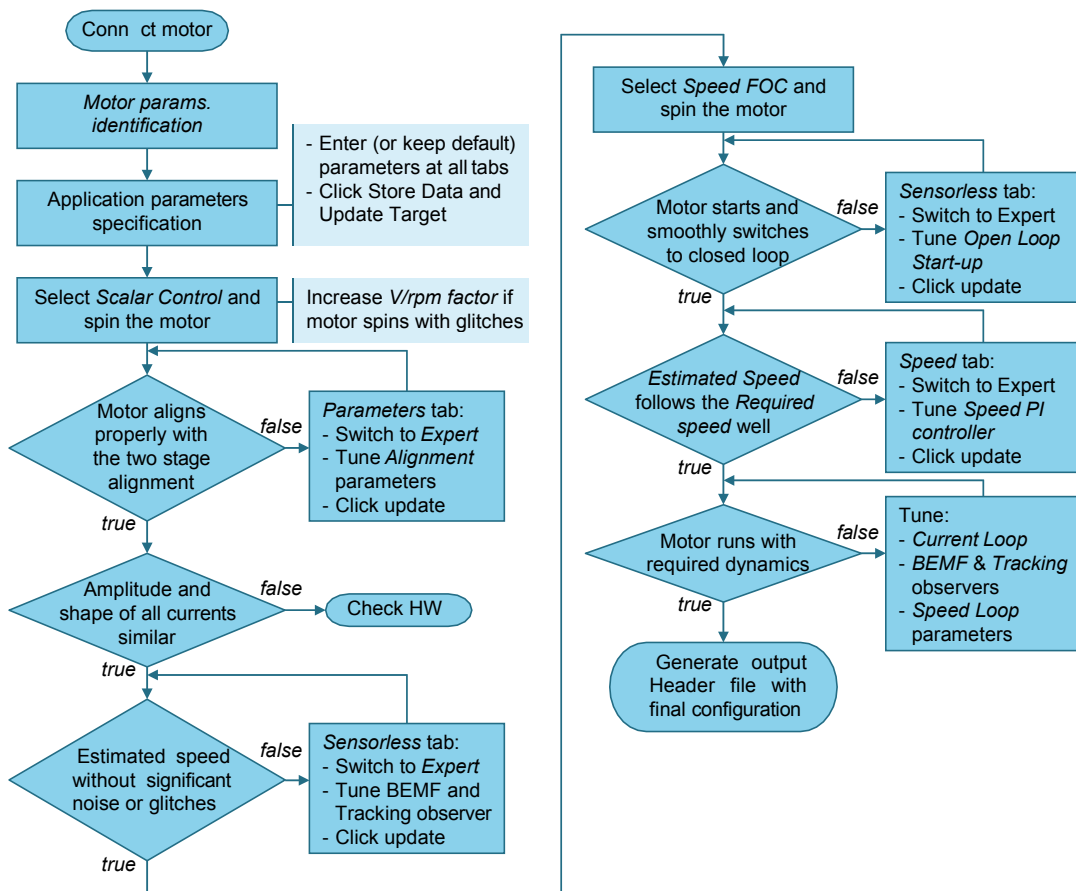


图21. 运行一个新PMSM

5.2.3 初始配置设置和更新

1. 打开包含专用MCAT插件模块的PMSM控制应用程序FreeMASTER项目。
2. 选择“Basic”模式（建议不熟悉电机控制理论的用户使用）。所需输入参数的数量将减少。
3. 选择“Parameters”选项卡。
4. 保留测量的电机参数或手动指定参数。电机参数可以从电机数据表中获得，也可以使用 *PMSM Electrical Parameters Measurement* (document [AN4680](#))中介绍的PMSM参数测量程序获得。表7中提供的所有参数均可在“Basic”和“Expert”模式下访问。电机惯量J表示整个系统的惯量，可以通过机械测量获得。J参数用于计算速度控制器常数。但是，手动控制器调整也可以用于计算该常数。

表7. MCAT电机参数

参数	单位	描述	典型范围
pp	[-]	电机极对数	1 – 10
Rs	[Ω]	一相定子电阻	0.3 – 50
Ld	[H]	一相直流电感	0.00001 – 0.1
Lq	[H]	一相正交电感	0.00001 – 0.1
Ke	[V.sec/rad]	BEMF常数	0.001 – 1
J	[kg.m ²]	系统惯量	0.000001 – 1
Iph nom	[A]	电机额定相电流	0.5 – 8
Uph nom	[V]	电机额定相电压	10 – 300
N nom	[rpm]	电机额定转速	1000 – 2000

5. 设置硬件量度—当使用标准功率级板的参考时，无需修改这两个字段。这些量度表示最大可测量的电流和电压模拟量。
6. 检查故障极限—这些字段在“Basic”模式下不可访问，并且使用电动机参数和硬件量度计算得出（请参见表8）。

表8. 故障极限

参数	单位	描述	典型范围
U DCB trip	[V]	外部制动电阻器开关接通的电压值	U DCB Over ~ U DCB max
U DCB under	[V]	检测到欠压故障的触发值	0 ~ U DCB Over
U DCB over	[V]	检测到过电压故障的触发值	U DCB Under ~ U max
N over	[rpm]	检测到超速故障的触发值	N nom ~ N max
N min	[rpm]	无传感器控制的最小实际速度值	(0.05~0.2) *N max

7. 检查应用量度—这些字段在“Basic”模式下不可访问，是通过电动机参数和硬件量度计算出来的。

表9. 应用量度

参数	单位	描述	典型范围
N max	[rpm]	转速量度	>1.1 * N nom
E max	[V]	反电动势量度	ke * Nmax
kt	[Nm/A]	电机转矩常数	—

8. 检查对齐参数—这些字段在“Basic”模式下不可访问，并使用电动机参数和硬件标尺进行计算。这些参数表示在转子对准及其持续期间施加到电动机的所需电压值。

- 单击“Store Data(存储数据)”按钮，将修改后的参数保存到内部文件中。

5.2.4 控制结构模式

- 通过单击“Scalar Control（标量控制）”部分中的“DISABLED（禁用）”按钮来选择标量控件。按钮的颜色变为红色，文本变为“ENABLED（启用）”。
- 打开应用程序开关。应用程序状态更改为“RUN（运行）”。
- 在“Freq_req”字段中设置所需的频率值，例如“Scalar Control（标量控制）”部分中为15 Hz。电动机开始运行（请参见图22）。

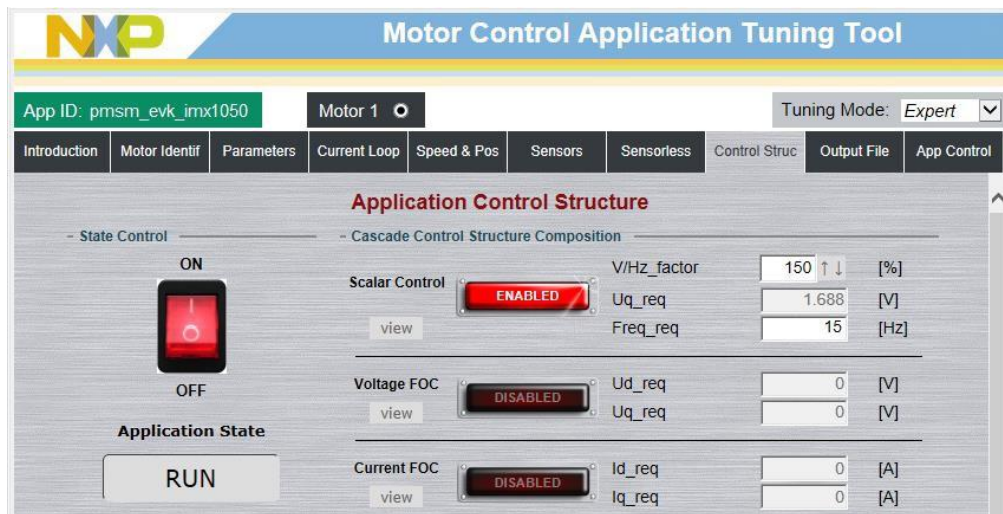


图22. MCAT标量控制

- 从“Scalar & Voltage Control（标量和电压控制）”FreeMASTER项目树中选择“Phase Currents（相电流）”记录器。
- 可以通过直接更改V/Hz因子或使用“UP/DOWN”按钮来找到V/Hz曲线的最佳比率。电动机电流的形状应接近正弦形状（见图23）。

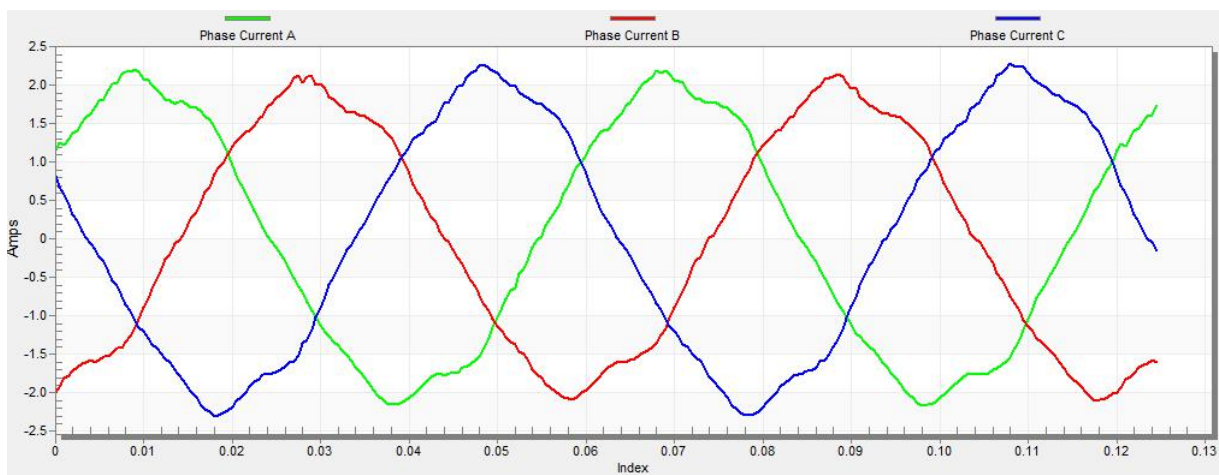


图23. 相电流

- 选择“Position（位置）”记录器以检查观测器功能。为了使反电动势位置和速度观测器正常工作，“Position Electrical Scalar（位置电子标量）”和“Position Estimated（预计位置）”之间的差异应最小（见图24）。位置差取决于电动机负载。负载越高，负载角度导致的位置之间的差异越大。

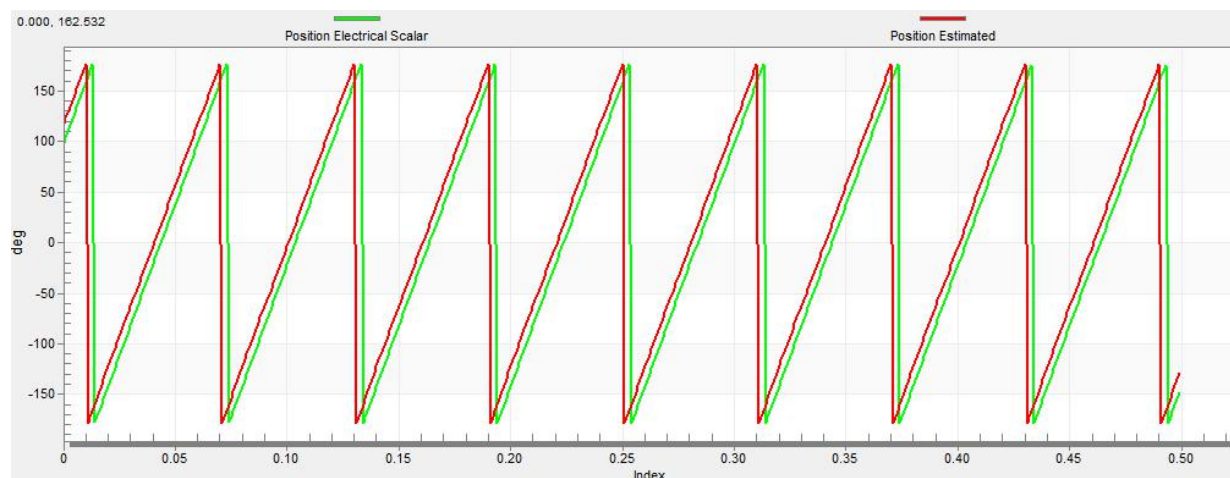


图24. 生成与估计的位置

- 如果需要相反的速度方向，请在“Freq_req”字段中设置负的速度值。
- 在此步骤中，适当的观察器功能和模拟量的测量被预期。
- 在主应用程序开关关闭的情况下，通过单击“Voltage FOC”部分中的“DISABLED（禁用）”按钮来启用电压FOC模式。
- 打开主应用程序开关，并在“Uq_req”字段中设置非零值。FOC算法通过已估计的位置来运行电动机。

5.2.5 编码器传感器设置

编码器传感器设置位于“Sensors（传感器）”选项卡中。您可以通过编码器传感器计算速度和位置，以进行传感器速度和位置控制。为了正确进行编码器计数，请设置每转一圈的编码器脉冲数和正确的计数方向。编码器脉冲数基于其制造商提供的有关编码器的信息。如果编码器传感器每转具有更多脉冲，速度和位置计算将更加准确。计数方向可以通过将编码器信号连接到NXP Freedom板或通过连接电机相位来提供。旋转方向可以通过如下步骤确定：

- 单击“Control Struct”选项卡的“Scalar Control”中的“DISABLED”按钮，选择标量控制。按钮颜色变为红色，文本变为“ENABLED”。
- 打开应用开关。应用状态更改为“RUN”。
- 在“Freq_req”字段中设置所需的频率值；例如“Scalar control”部分中的10 Hz。电动机开始运行。
- 检查编码器方向。从“Scalar & Voltage Control（标量和电压控制）”项目树中选择“Encoder Direction Scope（编码器方向范围）”。如果编码器方向正确，估计速度等于测得的机械速度。如果测得的机械速度与估计速度相反，则“Sensor”选项卡中的方向必须从0更改为1（反之亦然）。

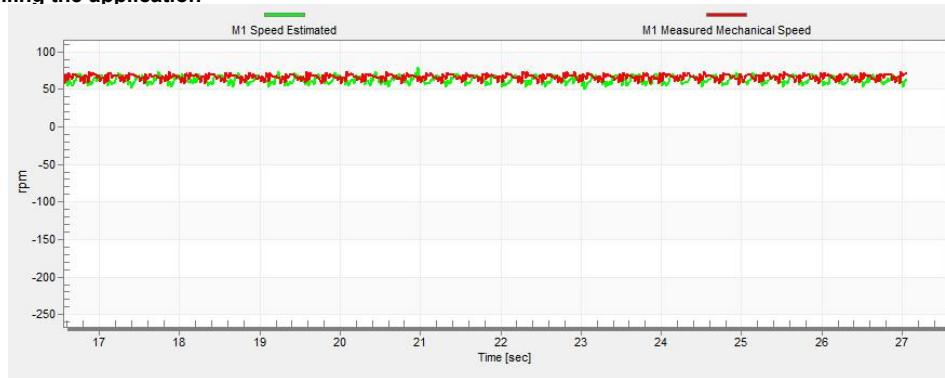


图25. 编码器方向–正确方向

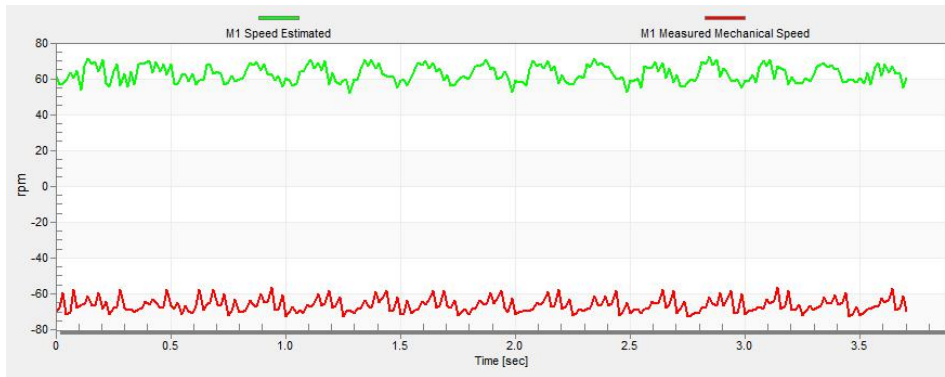


图26.编码器方向–错误方向

5.2.6 对齐调整

对于对齐参数，导航至“Tab”菜单，然后选择“Parameters”。对准过程将转子设置到准确的初始位置，您可以向电动机施加完整的启动转矩。转子对准参数可在“Expert”模式下进行编辑。主要对于高启动负载（压缩机、垫圈等）需要正确的初始位置。对准的目的是使转子处于稳定的位置，启动前不会有任何振荡。

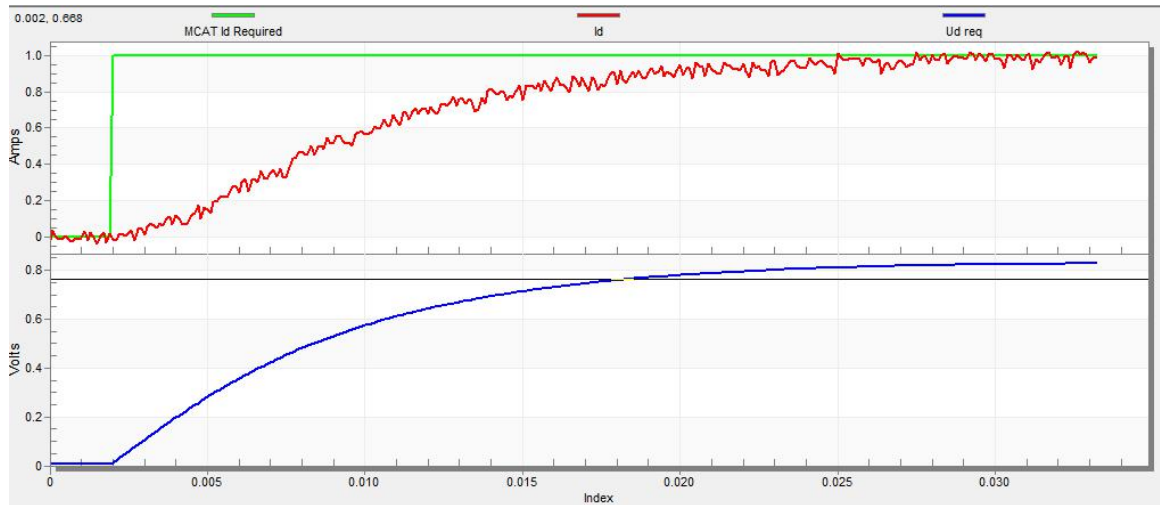
1. 对准电压是对准期间施加到d轴的值。增大此值可获得更高的轴负载。
2. 对齐持续时间表示调用对齐例程的时间。调整此参数可使转子在对准过程结束时没有振荡或运动。

5.2.7 电流回路调节

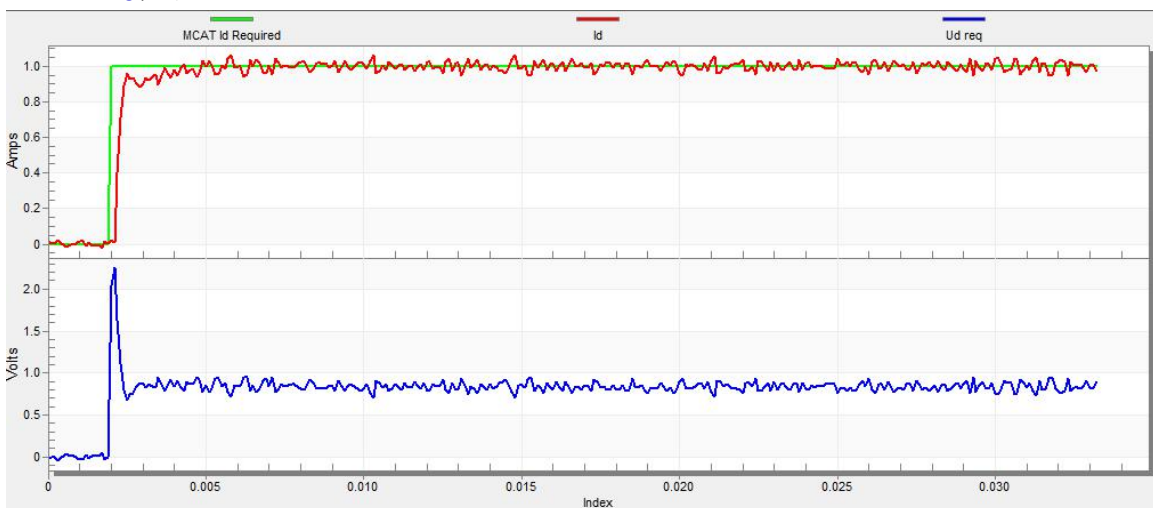
在“Basic”模式下使用电动机参数可以完全计算出电流D, Q PI控制器的参数，在此模式下无需采取任何措施。如果计算出的环路参数不符合所需的响应，则可以调整带宽和衰减参数。

1. 将调整模式切换为“Expert”。
2. 锁定电机轴。
3. 设置所需的环路带宽和衰减，然后单击“Current Loop”选项卡中的“Update Target”按钮。调谐环路带宽参数定义环路响应的速度，而调谐环路衰减参数定义实际数量的过冲幅度。

4. 选择“Current Controller ID”记录器。
5. 选择“Control Structure”选项卡，切换到“Current FOC”，将“ I_q _req”字段设置为非常低的值（例如0.01），然后在“ I_d _req”中设置所需的步骤。记录器中显示了控制回路响应（请参见图10）。
6. 调整环路带宽和衰减，直到获得所需的响应。示例波形显示了电流回路参数的正确和错误设置：
 - 环路带宽很低（110 Hz）， I_d 电流的建立时间很长（见图27）。

图27. I_d 电流控制器的慢速阶跃响应

- 环路带宽（400 Hz）是最佳的，并且 I_d 电流的响应时间足够（参见图28）。

图28. I_d 电流控制器的最佳阶跃响应

- 环路带宽很高（700 Hz）， I_d 电流的响应时间非常快，但会出现振荡和过冲（见图29）。

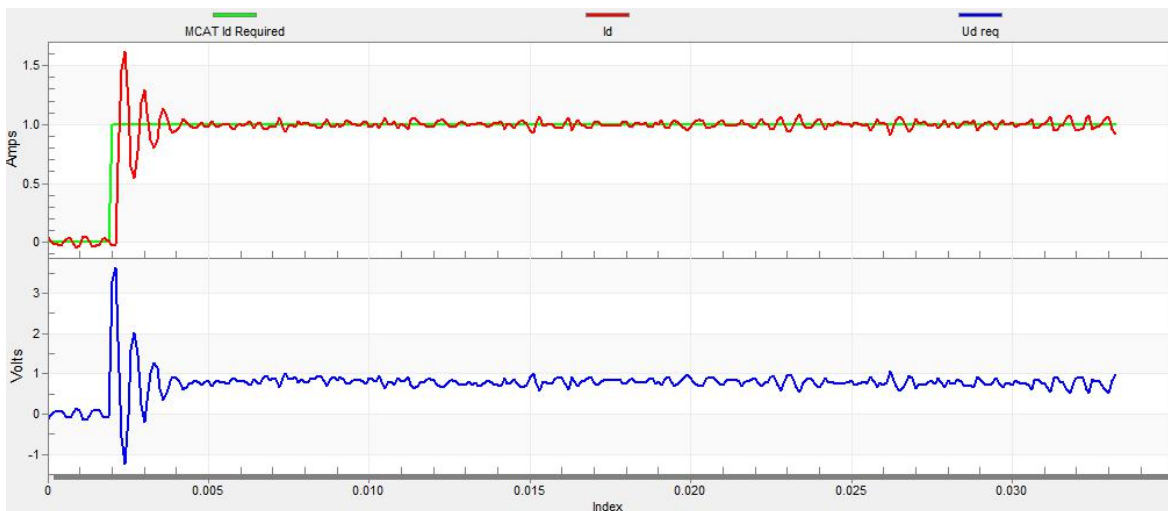


图29. Id电流控制器的快速阶跃响应

5.2.8 速度斜坡调整

速度命令通过速度斜坡施加到速度控制器。斜坡函数包含两个增量（向上和向下），表示电动机每秒的加速度和减速度。如果增量非常大，则会在加速过程中引起过电流故障，在减速过程中引起过电压故障。在“Speed”范围内，您可以看到“Speed Actual Filtered（速度实际滤波）”波形形状是否与“Speed Ramp（速度斜坡）”曲线相同。

增量对于标量和速度控制是常见的。增量字段位于“速度和位置”选项卡中，并且在两种调整模式下均可访问。单击“更新目标”按钮将更改应用到MCU。速度曲线示例如图30所示。斜坡下降的增量设置为500 rpm/sec，而上升的增量设置为3,000 rpm/sec。

启动斜坡增量位于“Sensorless（无传感器）”选项卡中，其值通常高于速度环斜坡的值。

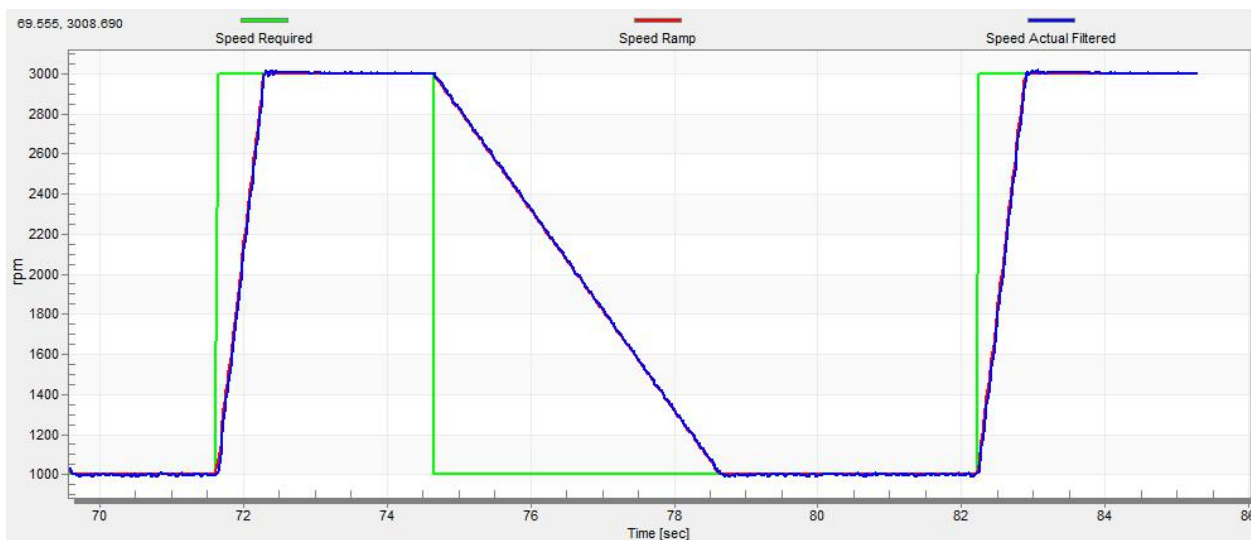


图30. 速度曲线

5.2.9 开环启动

可以通过“Sensorless”选项卡中的一组参数来调整启动过程。在两种调谐模式下均可访问其中两个参数（斜坡增量和电流）。除标量控制外，还可以在所有控制模式下处理启动调试。设置最佳值可正确启动电动机。低动态驱动器（风扇、泵）的启动状态示例如图31所示。

1. 从FreeMASTER项目树中选择“Startup（启动）”记录器。
2. 通常将启动斜坡增量设置为高于速度环斜坡增量的值。
3. 根据所需的启动转矩设置启动电流。对于风扇或泵等驱动器，启动转矩不是很高，可以设置为额定电流的15%。
4. 设置所需的合并速度—开环和估计位置合并开始时，阈值通常设置在标称速度的5%~10%的范围内。
5. 设置合并系数—在位置合并过程中，100%对应电气旋转的一半。值越高，合并越快。对于要求高启动转矩以及开环和闭环之间平稳过渡的驱动器，将其设置为接近1%的值。
6. 单击“Update Target（更新目标）”按钮以将更改应用到MCU。
7. 切换到“Control Structure（控制结构）”选项卡，然后启用“Speed FOC”。
8. 将所需速度设置为高于合并速度。
9. 在记录器中检查启动响应。
10. 调整启动参数，直到获得最佳响应。
11. 如果转子没有开始运行，请增加启动电流。
12. 如果合并过程失败（转子卡住或停止），请减小启动斜坡增量，增加合并速度，并将合并系数设置为5%。

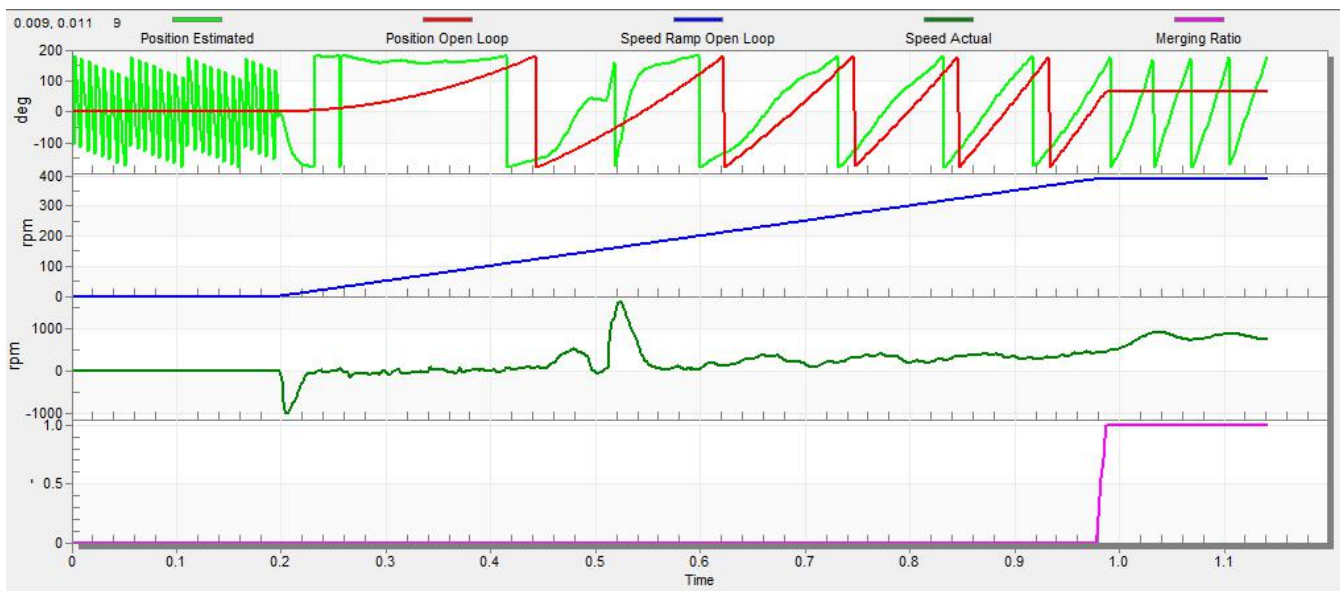


图31. 电机启动

5.2.10 BEMF观测器调整

在“Basic”模式下，将通过电动机参数完全计算出BEMF观测器和跟踪观测器的参数，在此模式下无需采取任何措施。如果计算出的环路参数与最佳响应不符，则可以调整带宽和衰减参数。

1. 将调整模式切换为“Expert”。
2. 从FreeMASTER项目树中选择“Observer”记录器。
3. 设置所需的带宽和BEMF观测器的衰减—带宽通常设置为接近当前环路带宽的值。
4. 设置所需的带宽和跟踪观测器的衰减—对于大多数低动力驱动器（风扇、泵），带宽通常设置为10 – 20 Hz。
5. 单击“Update Target（更新目标）”按钮以将更改应用到MCU。
6. 在记录器中检查观察者的响应。

5.2.11 PI速度控制器调整

电动机速度控制回路是一阶函数，其机械时间常数取决于电动机惯量摩擦力。第5.1.6节“机械参数测量”和第5.1.7节“PMSM电气参数测量过程”中介绍了使用PMSM电气和机械参数测量过程获得这些机械常数。如果这些机械常数可用，则可以使用环路带宽和衰减来调整PI控制器常数。否则，可以手动调节速度控制器的P和I部分以获得所需的速度响应（请参见图32中的示例响应）。有数十种方法可用于调整PI控制器常数。以下步骤提供了以下示例之一，以设置和调整PM同步电动机的速度PI控制器：

1. 从FreeMASTER项目树中选择“Speed Controller”选项。
2. 选择“Speed & Pos（速度和位置）”选项卡。
3. 选中“Manual Constant Tuning（手动常数调整）”选项，即禁用“Bandwidth（带宽）”和“Attenuation（衰减）”字段，启用“SL_Kp”和“SL_Ki”字段。
4. 调整比例增益：
 - a) 将“SL_Ki”积分增益设置为零。
 - b) 将速度斜坡设置为1000 rpm/sec（或更高）。
 - c) 切换到“Control Structure”选项卡，以合适的速度（约为标称速度的30%）运行电动机。
 - d) 将所需速度设置为 N_{nom} 的40%。
 - e) 切换回“Speed loop（速度环）”选项卡。
 - f) 调整比例增益“SL_Kp”，直到系统正确响应所需的值，并且没有任何振荡或过大的过冲：
 - 如果“SL_Kp”字段设置为低，则系统响应缓慢。
 - 如果“SL_Kp”字段设置为高，则系统响应更紧密。
 - 当“SL_Ki”字段等于0时，则系统很可能没有达到所需的速度。

- g) 单击“更新目标”按钮以将更改应用到MCU。
5. 调整积分增益：
- 缓慢增加“SL_Ki”字段，以将所需速度和实际速度之间的差异最小化为0。
 - 调整“SL_Ki”字段，使在应用所需的速度步长时看不到实际速度值的任何振荡或较大的过冲。
 - 单击“Update Target（更新目标）”按钮以将更改应用到MCU。
6. 调整环路带宽和衰减，直到收到所需的响应为止。下图显示了具有正确和错误设置速度环参数的示例波形：
- “SL_Ki”值很低，并且“Speed Actual Filtered（已过滤的实际速度）”无法达到“Speed Ramp（速度斜坡）”（参见图32）

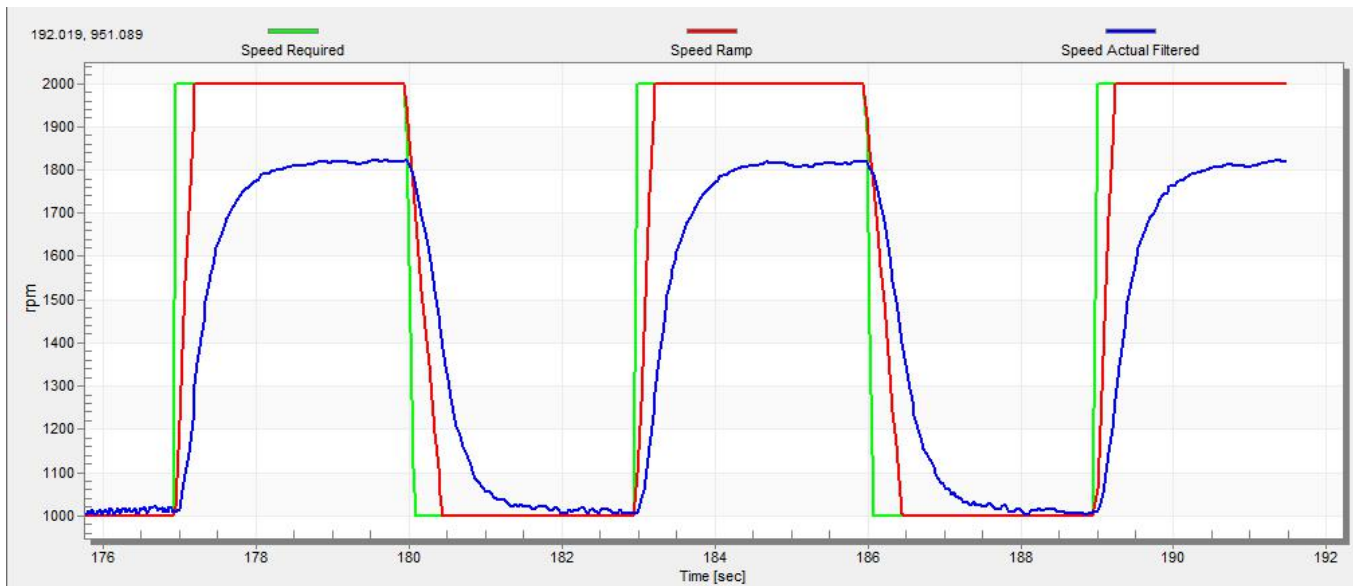


图32. 速度控制器响应—SL_Ki值低，未实现速度斜坡

- “SL_Kp”值低，“Speed Actual Filtered（实际速度已滤波）”大幅度过冲，并且不需要长时间的稳定时间（请参见图33）。

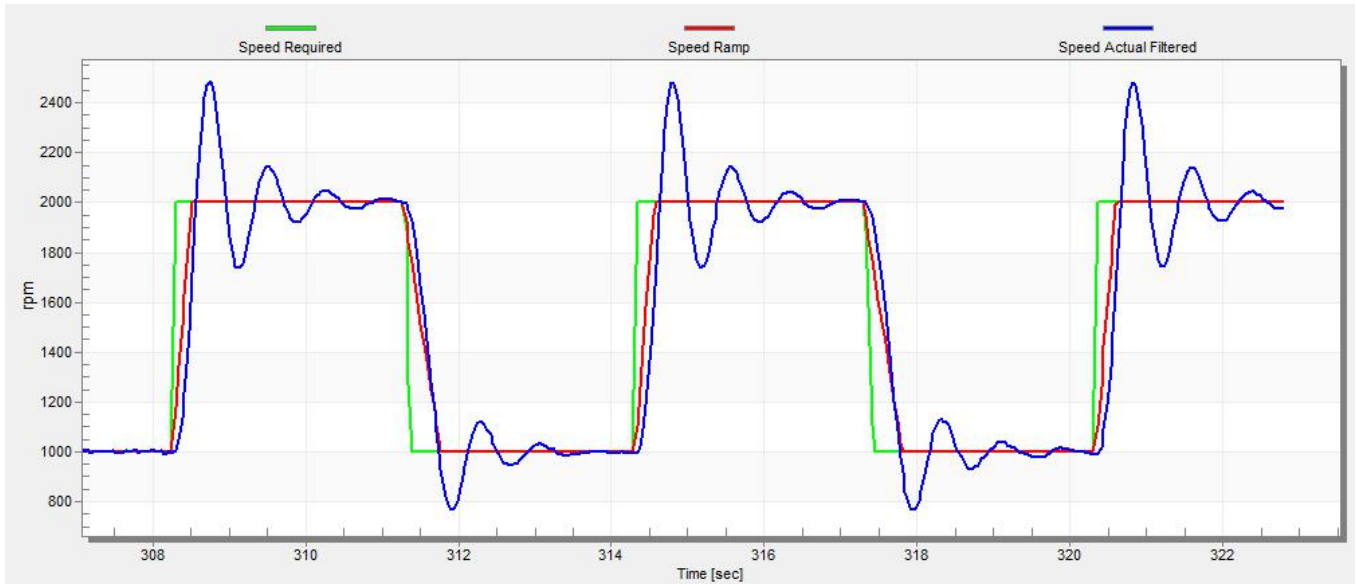


图33. 速度控制器响应— SL_Kp 值很低，速度实际滤波大幅度过冲

- 速度环响应具有较小的过冲，并且“Speed Actual Filtered（速度实际滤波）”建立时间已足够。可以将这种响应视为最佳（请参见图34）。

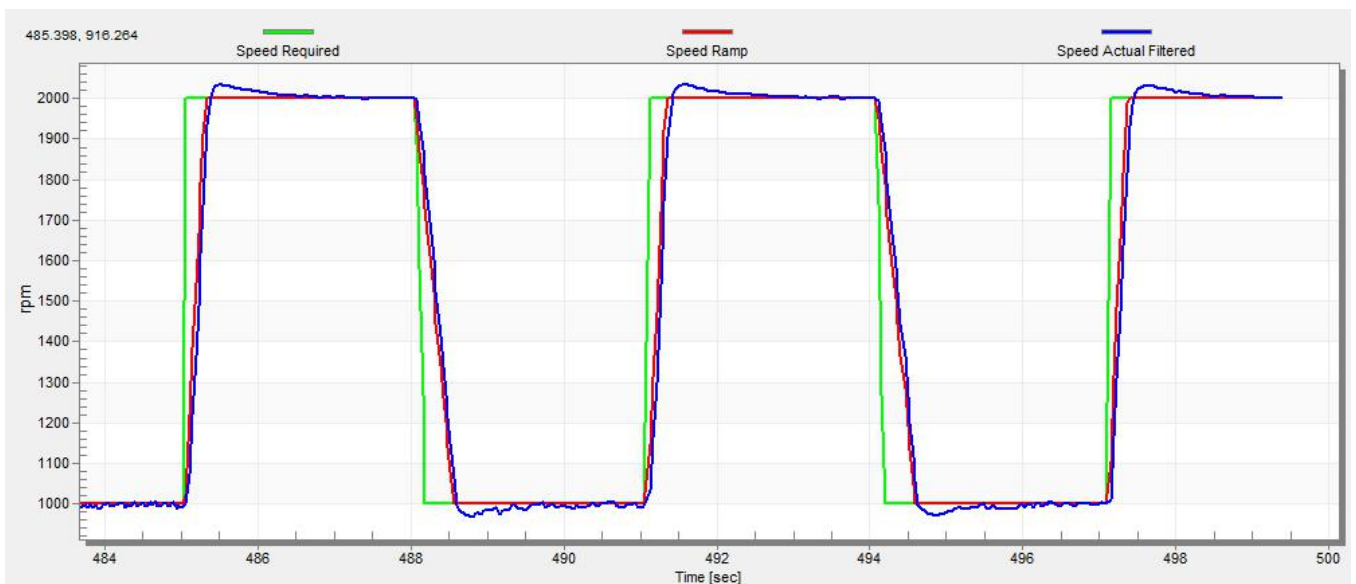


图34. 速度控制器响应-具有较小过冲的速度环响应

5.2.12 位置P控制器调整

可以通过“Speed & Pos”菜单选项卡中的比例增益“ PL_Kp ”来调整位置控制环。这是一个比例控制器，可用于不假装的位置控制系统。最佳位置响应的关键是控制器的适当值，该值只需将误差乘以比例增益（ Kp ）即可获得控制器输出。TG驱动电机的预定义基本值在用户指南中进行了说明，但是对于不同的电机，可以手动更改 必须将编码器传感器用于工作位置控制。以下步骤提供了如何设置PM同步电动机的位置P控制器的示例：

1. 从FreeMASTER项目树中选择“Position Controller（位置控制器）”。

2. 选择“Speed & Pos（速度和位置）”选项卡
3. 调整位置P控制器常数中的比例增益：
 - a) 设置一个小的值“PL_Kp”（M1位置环Kp增益），然后单击“Update Target”按钮。
 - b) 切换到“Control structure”，选择位置控制并将所需位置（举个例子）设置为10转。
 - c) 选择“Position Controller”并观察实际位置响应。
4. 重复前面的步骤，直到获得所需的位置响应。
“PL_Kp”值低，并且所需位置上的实际位置响应非常慢。

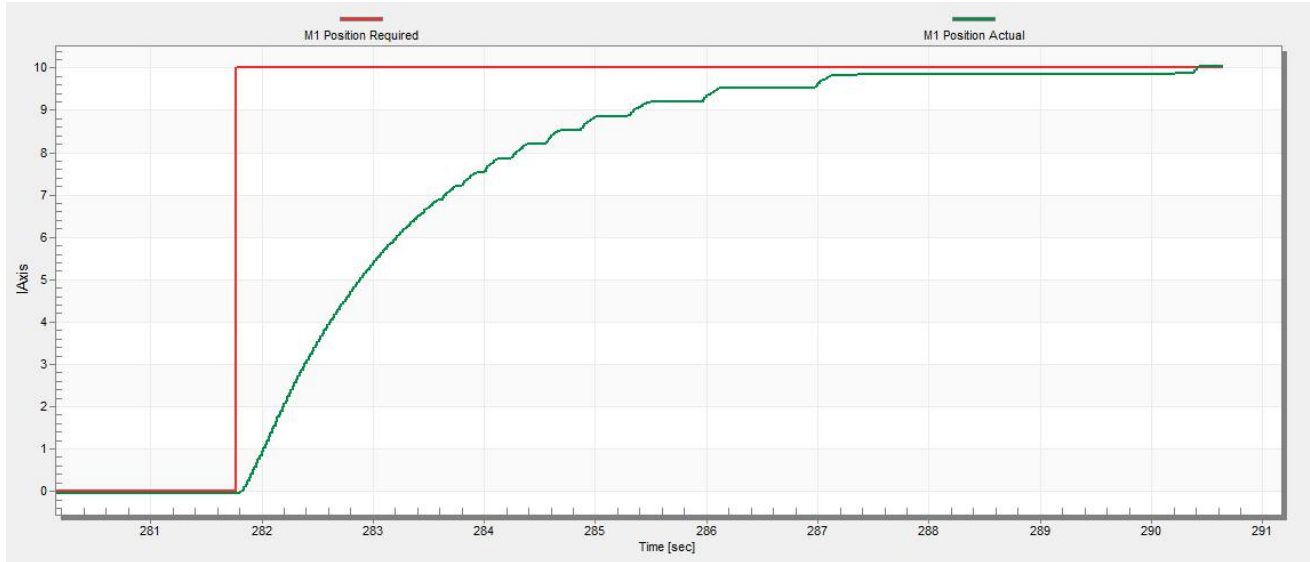


图35.位置控制器响应—PL_Kp值低，实际位置响应很慢

“PL_Kp”值太高，实际位置超出了所需位置。

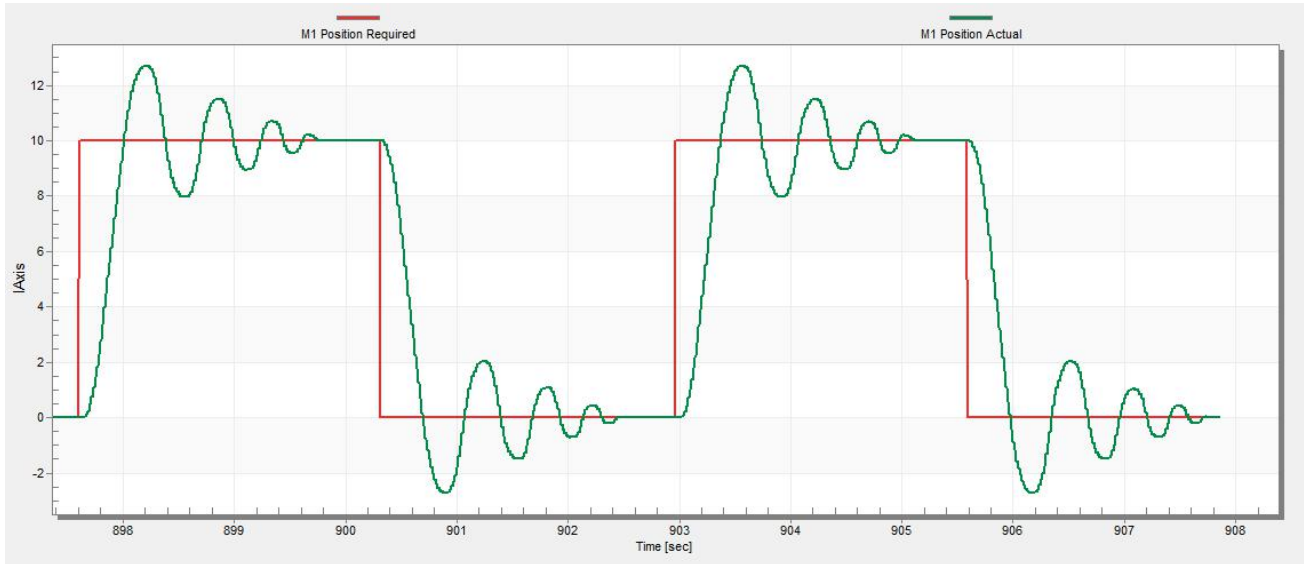


图36. 位置控制器响应-PL_Kp值太高，实际位置过冲

“PL_Kp”值是最佳的，实际位置响应是最佳的。

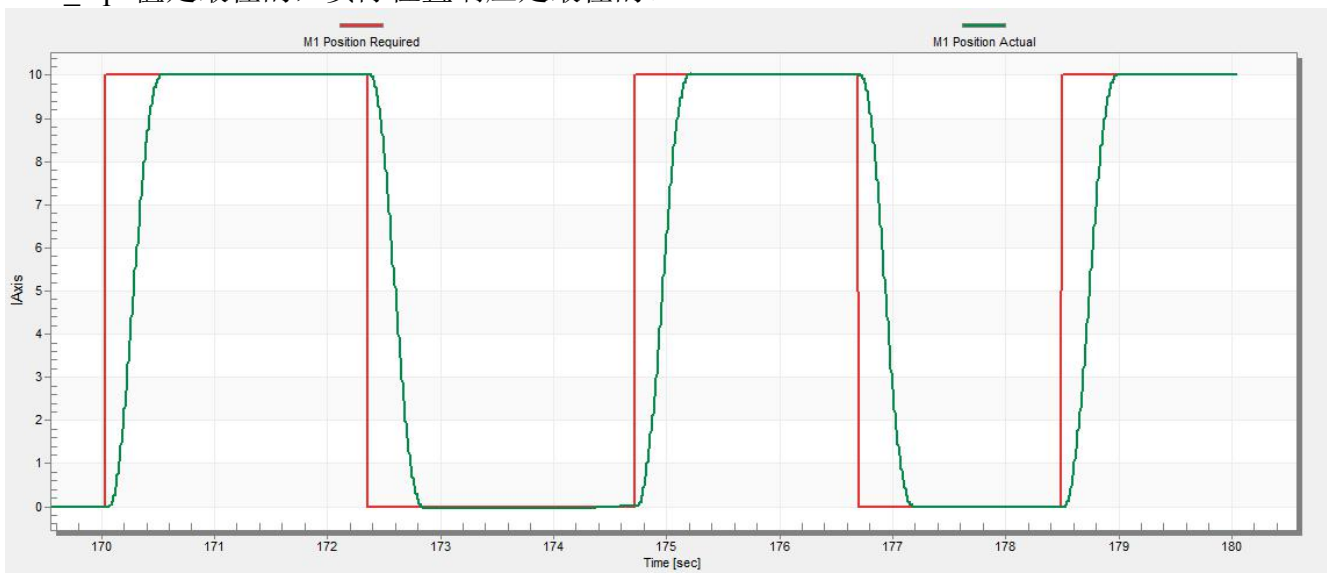


图37.位置控制器响应-实际位置响应良好

5.2.13 MCAT输出文件生成

当您成功完成应用程序的调整并希望将所有计算出的参数存储到嵌入式应用程序中时，导航至“Output File（输出文件）”选项卡，可以在此处查看MCAT生成的所有定义列表。单击“Generate Configuration File”按钮将覆盖包含这些定义的*m1_pmsm_appconfig.h*文件的旧版本。请注意，为了正确生成电动机参数文件，必须提供文件的正确路径。要更改路径，请将鼠标光标导航到MCAT屏幕的右上角，直到出现带有螺丝刀和扳手的符号。当您单击该符号时，将出现“Application Settings Page（应用程序设置页面）”。在“Project Path Selection（项目路径选择）”区域中，可以将路径修改为*m1_pmsm_appconfig.h*。

6 结论

本应用笔记介绍了使用FRDM-MC-LVPMSM NXP Freedom开发平台在NXP MIMXRT10xx EVK上实现三相PMSM的传感器和无传感器磁场定向控制的方法。第3节“MCU功能和外设设置”中介绍了控制软件的硬件相关部分（包括详细的外设设置、电机控制外设初始化和应用时序）。电机参数识别理论和识别算法本身在第5.1节“PMSM参数识别”中进行了介绍。本文档的最后一部分介绍并说明了基于FreeMASTER通信接口的电机控制应用优化（MCAT）工具所代表的用户界面。

7 缩略语与缩写

表10. 缩略语与缩写

缩略语	含义
ADC	Analog-to-Digital Converter
ACIM	Asynchronous Induction Motor
ADC_ETC	ADC External Trigger Control
AN	Application Note
CCM	Clock Controller Module
CPU	Central Processing Unit
DC	Direct Current
DRM	Design Reference Manual
ENC	Encoder
FOC	Field-Oriented Control
GPIO	General-Purpose Input/Output
LPUART	Universal Asynchronous Receiver/Transmitter
MCAT	Motor Control Application Tuning tool
MCDRV	Motor Control Peripheral Drivers
MCU	Microcontroller
PI	Proportional Integral Controller
PLL	Phase-Locked Loop
PMSM	Permanent Magnet Synchronous Machine
PWM	Pulse-Width Modulation
QD	Quadrature Decoder
TMR	Quad Timer
USB	Universal Serial Bus
XBAR	Inter-Peripheral Crossbar Switch

8 参考文献

以下参考文献可在www.nxp.com查阅：

1. *Sensorless PMSM Field-Oriented Control* (document [DRM148](#)).
2. *Motor Control Application Tuning (MCAT) Tool for 3-Phase PMSM* (document [AN4642](#)).
3. *Sensorless PMSM Field-Oriented Control on Kinetis KV* (document [AN5237](#)).
4. *PMSM Field-Oriented Control on MIMXRT1050 EVK* (document [AN12169](#)).
5. *i.MX RT1050 Processor Reference Manual* (document [IMXRT1050RM](#)).
6. *i.MX RT1020 Processor Reference Manual* (document [IMXRT1020RM](#)).
7. *i.MX RT1060 Processor Reference Manual* (document [IMXRT1060RM](#)).

How to Reach Us:**Home Page:**www.nxp.com**Web Support:**www.nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: www.nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, C 5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. Arm, AMBA, Arm Powered, Artisan, Cortex, Jazelle, Keil, SecurCore, Thumb, TrustZone, and μ Vision are registered trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. Arm7, Arm9, Arm11, big.LITTLE, CoreLink, CoreSight, DesignStart, Mali, Mbed, NEON, POP, Sensinode, Socrates, ULINK and Versatile are trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2018 NXP B.V.

Document Number: AN12214

Rev. 0
12/2018

