

UM0492  
用户手册  
STM32F103xx

# 永磁同步电机磁场定向控制 (PMSM FOC) 软件库 V2.0

简体中文版

\*本手册由优易特电子公司根据 ST 公司的 UM0492 文档，组织翻译而成，仅为方便用户学习使用，如有错误，请以英文原稿为准。

\*优易特电子技术支持邮箱: [uet@u-easytech.com](mailto:uet@u-easytech.com)

## 永磁同步电机磁场定向控制(PMSM FOC)软件库 V2.0

### 简介

本用户手册介绍了永磁同步电机（PMSM）FOC 软件库，STM32F103xx 微控制器就是针对磁场定向控制（FOC）3 相永磁电机固件库开发的。

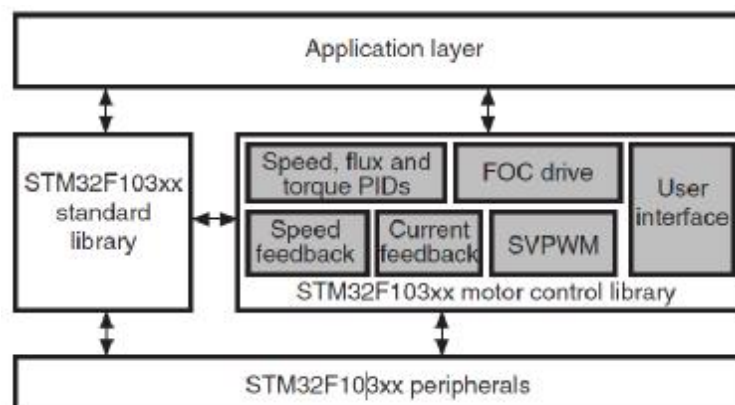
这些 32 位，ARM 的 Cortex™- M3 的芯 ST 微控制器（STM32F103xx）附带一套外围设备，使之适用于永磁交流同步电机和交流异步电机磁场定向控制。特别是，本手册介绍了 STM32F103xx 软件库发展到可以控制外置型或嵌入型或在扭矩和速度两个控制模式下控制正弦波驱动永磁电机。这些电机可能配备一编码器，具有三个霍尔传感器或者他们可能无传感器。在 UM0483 用户手册中介绍了具有编码器或测速发电机的交流感应电机的控制。

永磁同步电机磁场定向控制库是由几个 C 语言的模块构成，并列装 IAR EWARM 5.20, Keil 公司的 RealView MDK 3.22a 和 Green Hills 5.03 工作区。它是用来同时快速评估 MCU 和可用的工具。此外，当和 STM32F103xx 电机控制入门套件（STM3210B- MCKIT）及永磁电机一起使用时，可以让电机在很短的时间运转起来。它通过提供立即可用功能，也省去了 FOC 发展的耗时和速度调节的需要，让用户专注于应用层。而且，因为有了转子位置重构无传感器算法，它可能不需要任何速度传感器。

使用这个库的一个先决条件是要懂得 C 语言程序设计的基础知识，永磁电机驱动器和功率逆变器的硬件的基本知识。需要定制现有模块和添加一个新型的完整应用开发时才需要深入了解 STM32F103xx 功能。

图 1 显示了固件结构图。它广泛使用 STM32F103xx 标准库，但当需要优化执行速度和代码大小时它也直接作用于硬件设备的外围设备。

图1 固件结构图



## 目录

1 工具	7
1.1 工作环境	7
1.2 软件工具	7
1.3 源代码库	8
1.3.1 更新	8
1.3.2 文件结构	8
1.4 自定义 STM32F103xx 系列工作区	8
2 无传感器 FOC 的永磁电机介绍	10
2.1 永磁同步电机磁场定向控制驱动器简介	10
2.1.1 永磁电机结构	11
2.1.2 永磁同步电机基本方程	12
2.1.3 SM-PMSM 磁场定向控制 (FOC)	13
2.1.4 IPMSM 的最大转矩电流比 (MTPA) 的控制	14
2.1.5 前馈电流调节	16
2.2 无传感器的转子位置/速度反馈简介	17
2.3 弱磁控制简介	18
3 运行演示程序	20
3.1 转矩控制模式	20
3.2 速度控制模式	22
3.3 电流和速度调节器的调整	23
3.4 弱磁 PI 控制器的调节	24
3.5 观测器和 PLL 增益调节	25
3.6 DAC 的功能	26
3.7 电源反馈 (Power stage feedbacks)	26
3.8 故障信息	27
3.8.1 过流	27
3.8.2 过热	28
3.8.3 总线过压	28
3.8.4 总线欠压	28
3.8.5 启动失败	28
3.8.6 错误的速度反馈	28
3.9 在单分流拓扑结构和使用 MB459B 电路板的系统设置	29
3.10 使用 ICS 时的系统设置	29
3.10.1 PHASE_A_ADC_CHANNEL 和 PHASE_B_ADC_CHANNEL 的选择	30
3.11 使用编码器时的系统设置	30
3.12 使用霍尔传感器时的系统设置	32
3.13 先进无传感器系统开发	33
3.14 使用制动电阻时的系统设置	34
3.14.1 如何对制动电阻运行配置 FOC 软件库	35
3.14.2 如何对制动电阻运行修改 MB459 板	35
3.15 调试工具的注意事项	36
4 库的入门	37
4.1 库的配置文件: stm32f10x_MCconf.h	37

4.2 驱动控制参数: MC_Control_Param.h.....	40
4.3 增量编码器参数: MC_encoder_param.h.....	43
4.4 霍尔传感器参数: MC_hall_prm.h.....	44
4.5 状态观测器参数: MC_State_Observer_param.h.....	45
4.5.1 状态观测器参数.....	46
4.5.2 启动参数.....	47
4.5.3 统计参数.....	48
4.6 永磁同步电机 (PMSM) 参数: MC_PMSM_motor_param.h.....	49
4.6.1 基本电机参数.....	49
4.6.2 无传感磁场定向控制 (FOC) 电机参数.....	50
4.6.3 弱磁运行的额外参数.....	50
4.6.4 IPMSM 驱动优化 (MTPA) 的额外参数.....	51
4.6.5 用于前馈、高性能电流控制的额外参数.....	51
5 函数库.....	53
5.1 三相电阻拓扑电流采集和空间矢量 PWM 产生: stm32f10x_svpwm_3_shunt 模块.....	53
5.1.1 可用函数列表.....	53
5.1.2 空间矢量脉冲宽度调制 (SVPWM) 运行.....	56
5.1.3 三相分流拓扑电流采集和通用 A/D 转换.....	58
5.1.4 三相分流电阻拓扑结构中调节延时参数和定子电流采集.....	59
5.2 单相电阻拓扑电流读取和空间矢量 PWM 产生: stm32f10x_svpwm_1shunt module.....	63
5.2.1 可用函数列表.....	63
5.2.2 单相分流拓扑电流采集.....	67
5.2.3 定义干扰参数和边界区域.....	69
5.2.4 性能.....	71
5.3 隔离电流传感器读取和空间矢量的 PWM 产生: stm32f10x_svpwm_ics module.....	72
5.3.1 可用的函数列表和中断服务程序.....	72
5.3.2 隔离电流传感器拓扑和通用 A/D 转换的整合中电流采集.....	74
5.4 PMSM (SM-PMSM / IPMSM) 磁场定向控制: MC_FOC_Drive and MC_FOC_Methods 模块.....	75
5.4.1 可用 C 函数列表.....	76
5.5 参考坐标转换: MC_Clarke_Park 模块.....	82
5.5.1 可用 C 函数列表.....	83
5.5.2 详解参考坐标系转换.....	85
5.5.3 循环限制.....	87
5.6 编码器反馈处理: stm32f10x_encoder 模块.....	88
5.6.1 可用函数列表和中断服务程序.....	88
5.7 霍尔传感器反馈处理: stm32f10x_hall 模式.....	91
5.7.1 可用函数列表.....	91
5.7.2 速度测量实施.....	93
5.7.3 电角度推算方法.....	96
5.8 无传感器速度/位置检测: MC_State_Observe 和 MC_State_Observer_Interface 模块.....	96
5.8.1 可用 C 函数列表.....	96

5.9 PID 调节器: MC_PID_regulators 模块.....	100
5.9.1 可用函数列表.....	101
5.9.2 PID 调节器理论背景.....	101
5.9.3 调节器采样时间设定.....	102
5.9.4 电机频率与速度定义循环 Ki, Kp, Kd 的调节.....	103
5.10 通用目的时间基准: stm32f10x_Timebase 模块.....	105
5.10.1 可用函数列表与中断服务程序.....	105
5.11 功率级核查: MC_MotorControl_Layer 模块.....	107
5.11.1 函数列表.....	107
5.12 主中断服务程序: stm32f10x_it module.....	110
5.12.1 非空中断服务程序列表.....	110
附录 A 附加信息 .....	112
A.1 调节运行 FOC 算法相关负载.....	112
A.2 基于单相或三相分流电阻配置的 PWM 频率选择更新重复频率.....	113
A.3 固定点数字代表 .....	113
A.4 磁通和转矩电流 PI 增益优先决策.....	114
A.5 调节电流调节器 .....	116
A.6 静态观测增益优先策略.....	118
A.7 速度格式 .....	118
A.8 MMI (最大调制指数): 自动计算.....	119
A.9 参考文献 .....	121
修订史.....	122
<b>www.st.com</b> .....	124

## 永磁同步电机磁场定向控制软件库 2.0 版的特征 (CPU 运行在 72MHz)

- I 支持速度反馈:
  - 传感器
  - 间隔  $60^{\circ}$  或  $120^{\circ}$  配置霍尔传感器
  - 正交增量编码器
- I 电流取样方法:
  - 两个相互隔离的电流传感器 (ICS)
  - 单一, 通用直流环节分流电阻
  - 在三个逆变器底部间隔处放置三个分流电阻
- I 优化的永磁同步电动机和凸极永磁同步电动机驱动器
- I 磁场减弱
- I 前馈, 高性能的电流调节
- I 用于跟踪最重要的软件变量的 DAC 功能
- I 制动电阻的管理
- I 转速调节控制模式
- I 转矩调节控制模式
- I 16 位的空间矢量
  - PWM 频率易于调整
  - 中心型 PWM 模式
  - 17.6KH 时的 11 位分辨率
- I “先验”所有必要参数的规则, 对固件通用化是必须的
- I 在 3 分流/无传感器配置 (10KHz FOC 的采样率) 下 CPU 负载低于 22%
- I 在 3 分流/无传感器情况下为 LCD /操纵杆的管理, 配置代码大小约为 12.5Kbytes 外加 11.5Kbytes

# 1 工具

为了使用永磁同步电机磁场定向控制软件库，开发一个关于永磁同步电机的应用程序，您必须建立一个完整的开发环境，如下文所述。同时，您需要上运行 Windows XP 的 PC 机。

## 1.1 工作环境

当使用使用的主要的硬件板时，包括 STM3210B- MCKIT 入门套件（一个完整的逆变器和控制电路板），永磁同步电机磁场定向控制软件库完全的生效。该 STM3210B- MCKIT 入门套件为新建工程和使用软件库提供了一个理想工具包。因此，为了快速使用和评价用户手册介绍的软件，建议首先获取入门工具包。

该库由 IAR EWARM v. 5.20, KEIL RVMDK v. 3.22 和 Green Hills MULTI v. 5.0.3 工作区提供。使用 IAR 进行详细验证，同时可以使用其他工具链进行简单的功能验证。您可以针对任何其他工具链手动设置设置您的工作空间。

## 1.2 软件工具

一个完整的软件包包括：

- I 第三方集成开发环境（IDE）
- I 一个第三方 C 编译器
- I JTAG 调试和编程接口

利用微控制器的 JTAG 界面，使用大多数的工具链就能进入在线调试会话框。

每个工具链都可以由一个连接 PC 机和目标应用程序的界面来提供。

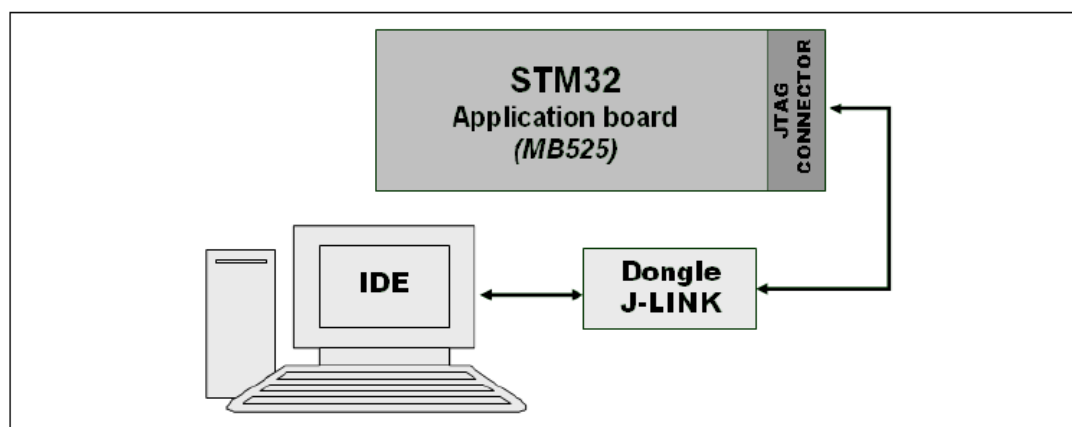


图2 用于调试和编程的JTAG界面

JTAG 接口也可用于在线 MCU 的编程。其他编程程序可以从第三方获得。

## 1.3 源代码库

### 1.3.1 更新

在新项目开始之前，强烈建议检查是否有最新的版本库，然后要随时检查所有版本注释，以便于了解到对您的项目有用的新特征。在 ST 网站上注册后可以自动获取更新。

### 1.3.2 文件结构

永磁同步电机（PMSM）磁场定向控制（FOC）软件库包含了前面提到的工具链的工作区间。点开文件，包含的库结构就会显现出来，如图 3 所示。

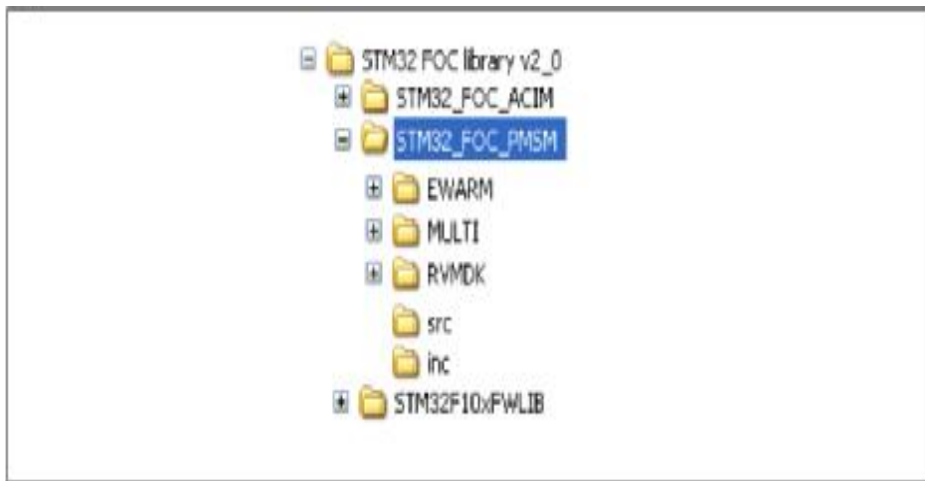


图3 文件结构

STM32 FOC 固件库 V2.0 文件夹包含 3 相（带传感器或无传感器）永磁同步电机（PMSM）和带传感器的交流感应电机固件库。

该 STM32F10xFWLIB 夹包含了 STM32F103xx 标准库。

该 inc 文件夹包含头文件，src 文件夹包含了控制电机的源文件库。

最后，EARM, RVMDK 和 MULTI 的每个文件夹包含各自的工具链结构文件，外加上一个 lib 文件夹，lib 文件夹包含两个编译的目标文件模块：MC\_State\_Observer 和 MC\_FOC\_Methods。

根据要求，从 ST 来的完整的源文件完全免费。不要犹豫联系离你最近的 ST 销售办事处或支持小组。

## 1.4 自定义 STM32F103xx 系列工作区

永磁同步电机（PMSM）磁场定向控制（FOC）软件库是针对 STM32F103VB6 的。但是，它应用在 STM32F103xx 系列产品上同样成功。



根据可用的功能（有些 I/ O 端口是不出现低电平），在使用不同类型的 STM32F103xx 系列时，可能需要对库进行一些修改。详情参阅单片机数据表。

另外，根据内存大小，工作区可能要进行配置，以适合您的 STM32F103xx MCU 系列。

## 2 无传感器 FOC 的永磁电机介绍

### 2.1 永磁同步电机磁场定向控制驱动器简介

该软件库是为了实现由完善的定向控制（FOC）策略提供飞高动态性能交流永磁同步电动机（PMSM）控制而设计的。

使用这种方法，可以说，通过控制定子电流精确变换而来的  $i_{qs}$  和  $i_{ds}$  两种电流，调节电磁转矩（ $T_e$ ），并在一定程度上达到弱磁能力就成为可能。

这类似于一个状态良好的直流电机，其中的角色相当于电枢电流和励磁电流。

因此，可以说，FOC 包括定子电流相位的控制和定向以及与转子磁通量正交。本定义明确指明测量定子电流和转子角度的必要性。

基于该算法结构（和库函数）的基本信息，如图 4 所示。

- I 空间矢量 PWM 模块（CALC SVPWM），实现了先进的降低电流谐波调制方法，从而优化了直流母线的利用
- I 使用低价分流电阻或市场有售的隔离电流霍尔传感器（ICS），电流读取模块可以保证系统正确测量定子电流
- I 转子的速度/位置反馈模块可以让系统通过操纵霍尔传感器或增量编码器的信号，从而得到正确的转子角速度或位置。此外，该固件库提供了无传感器检测转子速度/位置，如第 2.2 节所述。
- I PID 控制器块实现比例，积分和微分反馈控制（电流调节）
- I Clarke, Park, Reverse Park 和 Circle 限制模块实现 FOC 要求的精确转换。

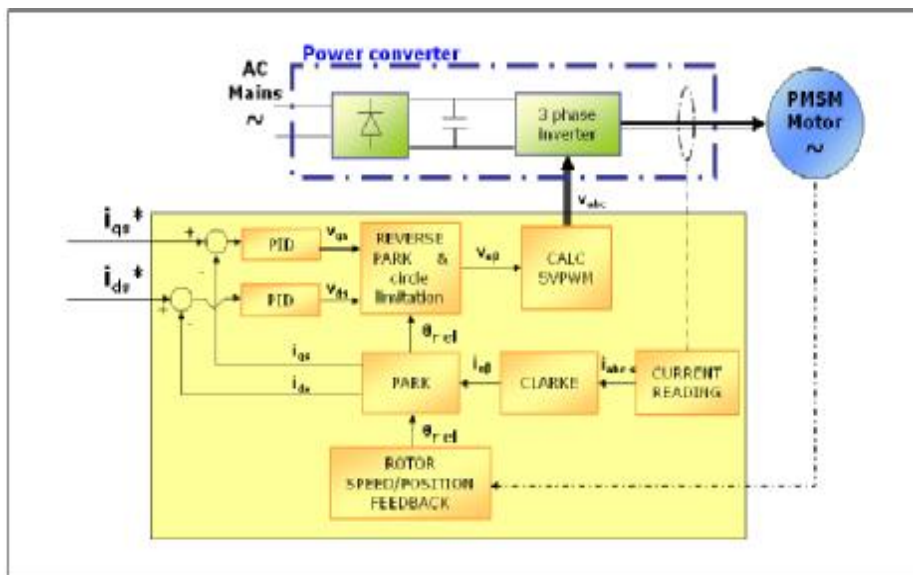
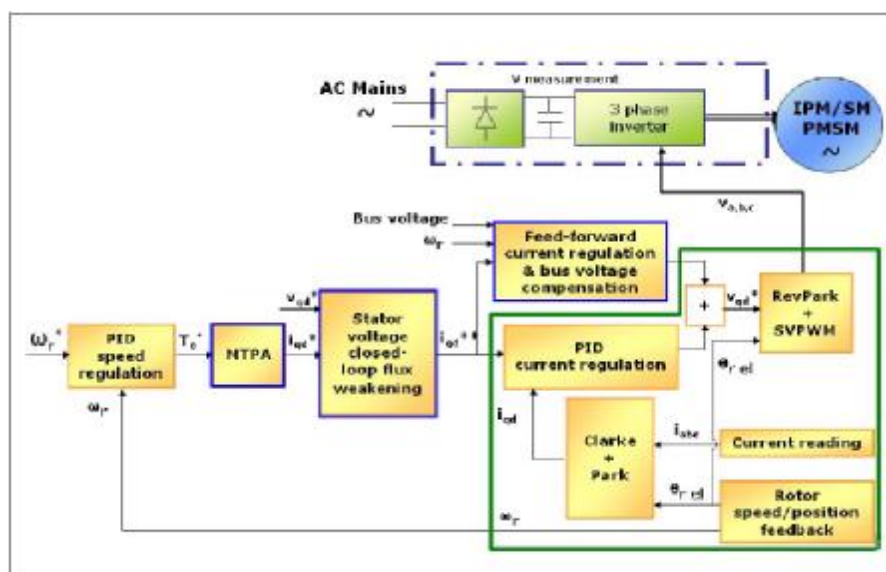


图 4 FOC 算法的基本结构，转矩控制



可以用电流  $i_{qs}$  和  $i_{ds}$  控制电磁转矩和磁通量。

另一方面,图 5 显示了速度控制回路和电机控制库提供的完整的具体特征设置。参阅第 2.1.4 节,第 2.3 节和第 2.1.5 节。第 2.1.4 节讲解了适用于永磁同步电动机 (IPMSM) 的 MTPA (最大转矩/安培) 优化策略。第 2.3 节介绍了弱磁控制,第 2.1.5 说明了如何利用前馈电流调节。

这些功能都可以根据电机的使用 and 用户需要，通过选项进行设置（见 4.1 节）。

### 2.1.1 永磁电机结构

主要有两种不同的永磁电机结构可供选择:

- 如图 6 (a) 所示, 磁铁粘在转子表面, 这就是为什么它被称为表贴式永磁同步电机 (SM-PMSM)
- 如图 6 (b) 和 (c) 所示, 磁铁被嵌在转子结构。这种结构被称为嵌入式永磁同步电机 (IPMSM)

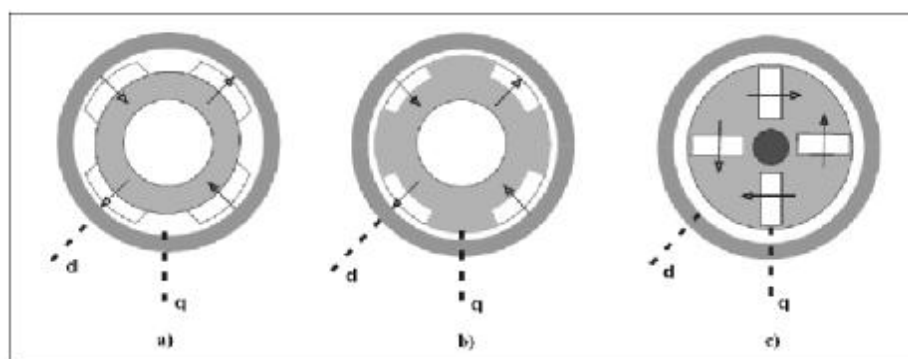


图 6 永磁电机的不同结构

表贴式永磁同步电机（SM - PMSM）本身具有各向同性的结构，即直接和正交电感  $L_d$  为和  $L_q$  是相同的。通常情况下，其机械结构产生更广泛的气隙，这反过来又意味着弱磁能力更低。

另一方面，IPMSMs 显示出各向异性结构（典型的， $L_d < L_q$ ），图 b）称为嵌入式永磁电机各向异性稍弱，图 c）称为埋入式或径向永磁电机，各向异性很强；这一特殊的磁场结构可以用来（如 2.1.4 节阐述的一样）产生更大的电磁转矩。它们都具有优良的机械结构，通常气隙很狭窄，从而弱磁能力更好。

这固件库适与表贴式永磁同步电机（SM- PMSM）和嵌入式永磁同步电机（IPMSM）连接已经优化。

## 2.1.2 永磁同步电机基本方程

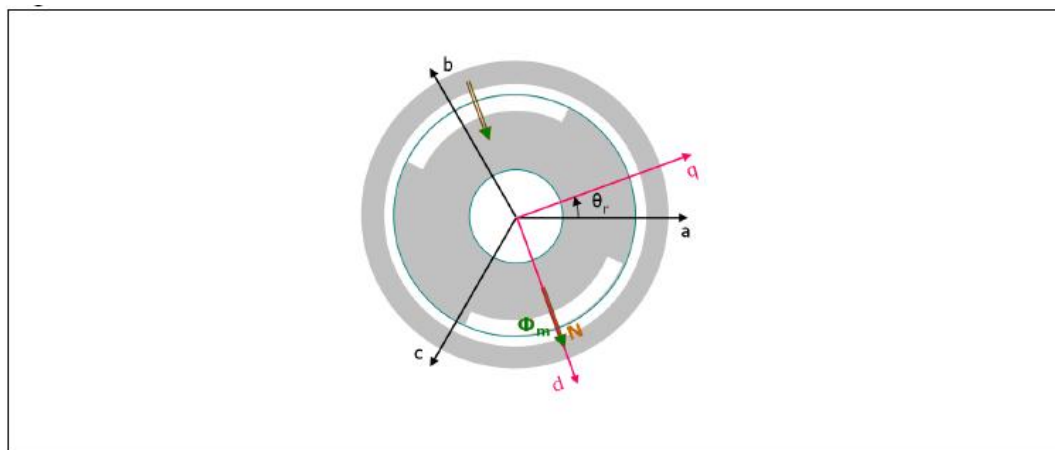


图 7 设定的永磁同步电动机的参考框架

参考图 7，结合永磁同步电动机的电压和磁链的方程（SMPMSM 或 IPMSM），一般表示为：

$$v_{abc_s} = r_s i_{abc_s} + \frac{d\lambda_{abc_s}}{dt}$$

$$\lambda_{abc_s} = \begin{bmatrix} L_{Is} + L_{ms} & -\frac{L_{ms}}{2} & -\frac{L_{ms}}{2} \\ -\frac{L_{ms}}{2} & L_{Is} + L_{ms} & -\frac{L_{ms}}{2} \\ -\frac{L_{ms}}{2} & -\frac{L_{ms}}{2} & L_{Is} + L_{ms} \end{bmatrix} i_{abc_s} + \begin{bmatrix} \sin\theta_r \\ \sin(\theta_r - \frac{2\pi}{3}) \\ \sin(\theta_r + \frac{2\pi}{3}) \end{bmatrix} \Phi_m$$

l  $r_s$  为定子相绕组电阻

l  $L_{Is}$  是定子相绕组自感

l  $L_{ms}$  是定子三相绕组互感；就永磁同步电动机（IPMSM）本身而言，本身磁场与互感磁场都有二次谐波分量  $L_{2s}$ ，当  $k=0 \pm 1$  时，除了恒量  $L_{ms}$  其它与  $\cos(2q_r + k \times 2p/3)$  成比例。

- l  $q_r$  是转子的电角度
- l  $f_m$  是永磁产生的磁通链

这些方程的复杂性是很明显的，因为三个定子磁链相互耦合，更甚者，因为他们都依赖于转子的位置，而转子的位置时时变化的，也是负载转矩和电磁的函数。

参照结构理论改变定子变量 abc（即可以看作沿着间隔  $120^\circ$  的轴线）成为 qd 分量，直接沿着间隔  $90^\circ$  的轴，与转子同步旋转，反之亦然（更多详情参见 5.5 节），简化了永磁电机方程。在 d “径直” 轴与转子磁通对齐，而 q “正交” 轴在位置滚动方向成  $90$  度方向。

电机的电压和磁通量方程简化为：

$$\begin{cases} v_{qs} = r_s i_{qs} + \frac{d\lambda_{qs}}{dt} + \omega_r \lambda_{ds} \\ v_{ds} = r_s i_{ds} + \frac{d\lambda_{ds}}{dt} - \omega_r \lambda_{qs} \\ \lambda_{qs} = L_{qs} i_{qs} \\ \lambda_{ds} = L_{ds} i_{ds} + \Phi_m \end{cases}$$

对于表贴式型永磁同步电机（SM-PMSM），在 d-和 q-轴线的电感参数是相同的（参照 2.1.1 节），那么就有：

$$L_s = L_{qs} = L_{ds} = L_{Is} + \frac{3L_{ms}}{2}$$

另一方面，嵌入内永磁同步电机（IPMSM）具有凸出磁场结构，所以它们的电感参数可写为：

$$\begin{aligned} L_{qs} &= L_{Is} + \frac{3(L_{ms} + L_{2s})}{2} \\ L_{ds} &= L_{Is} + \frac{3(L_{ms} - L_{2s})}{2} \end{aligned}$$

### 2.1.3 SM-PMSM 磁场定向控制（FOC）

下面的公式描述了一个 SM-PMSM 电磁转矩：

$$T_e = \frac{3}{2} \bar{P} (\lambda_{ds} i_{qs} - \lambda_{qs} i_{ds}) = \frac{3}{2} \bar{P} (L_s i_{ds} i_{qs} + \Phi_m i_{qs} - L_s i_{qs} i_{ds})$$

$$T_e = \frac{3}{2} \bar{P} (\Phi_m i_{qs})$$

最后一个方程表明，这个乘积分量  $i_{qs}$  与转矩呈线性关系，而电流分量  $i_{ds}$  对转矩没有影响（如上所述，这些方程是对 SM-PMSMs 适用）。

因此，如果  $I_s$  是电动机的额定电流，那么它的最大扭矩是在  $i_{qs} = I_s$  和  $i_{ds} = 0$ （事实上  $I_s = \sqrt{i_{qs}^2 + i_{ds}^2}$ ）时产生的。在任何情况下使用 SM-PMSM，都可以在  $i_{ds} = 0$  时得到最优化的转矩/电比，这一点是很明确的。因此，这种选择对应于各向同性电机的 MTPA（最大扭矩电流比）控制。在 4.1 节可以找到如何设置库的配置和执行这一方法。

另一方面，磁通量可以通过调节直轴电流  $i_{ds}$  进行削弱；这样扩大了速度变化范围，但会降低最大的交轴电流  $i_{qs}$ ，并因而降低提供给负载的电磁转矩（有关弱磁策略的详情见 2.3 节）。

总之，通过调节电机电流分量  $i_{qs}$  和  $i_{ds}$ ，FOC 方法可以调节 PMSM 转矩和磁通；电流调节是靠通常所说的“同步帧的 CR - PWM”的方式实现。

#### 2.1.4 IPMSM 的最大转矩电流比（MTPA）的控制

嵌入式永磁同步电动机（IPMSM）的电磁转矩公式为：

$$T_e = \frac{3}{2} \bar{P} (\lambda_{ds} i_{qs} - \lambda_{qs} i_{ds}) = \frac{3}{2} \bar{P} (L_{ds} i_{ds} i_{qs} + \Phi_m i_{qs} - L_{qs} i_{qs} i_{ds})$$

$$T_e = \frac{3}{2} \bar{P} (\Phi_m i_{qs} + \frac{3}{2} \bar{P} L_{ds} - L_{qs} i_{qs} i_{ds})$$

在这个表达式的第一项是 PM 的励磁转矩。第二项是所谓的磁阻转矩，由于内在凸磁结构，它代表一个额外的组成部分。此外，由于通常  $L_d < L_q$ ，仅当  $i_{ds} < 0$  时磁阻转矩和励磁转矩有相同的方向。

考虑到转矩方程，它可以指出，电流  $i_{qs}$  和  $i_{ds}$  都直接影响转矩的产生。

该 MTPA（最大扭矩电流比）控制的目的是计算参考电流（ $i_{qs}$ ,  $i_{ds}$ ），极限与产值与电磁转矩和铜损失有关。（条件： $I_s = \sqrt{i_{qs}^2 + i_{ds}^2} \leq I_n$ ）

因此，给定一组电机参数（磁极对数，垂直和正交电感  $L_d$  为和  $L_q$ ，磁铁磁链，额定电流），MTPA 轨迹可以看作电流（ $i_{qs}$ ,  $i_{ds}$ ）的轨迹，从而降低所需的力矩消耗的电流（见图 8）。

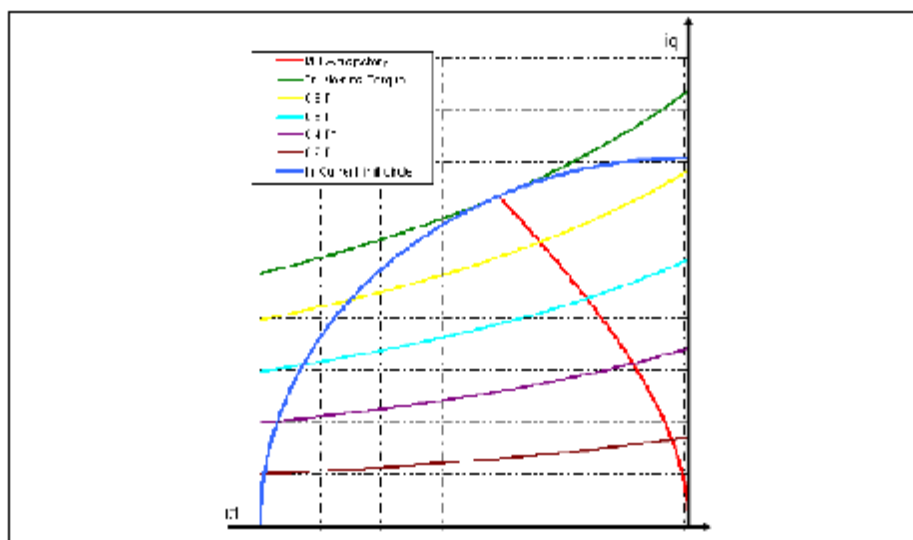


图 8 MTPA 轨迹

4.1 节介绍如何激活此功能。把电机参数输入到包含在固件库包的电子表格中(该文件的位置是 STM32MCKIT\design tools \PMSM\_MTPA\_FEEDFORWARD.xls), 就可以预先计算的 MTPA 轨迹并将计算结果作为 8 区间线性内插系数, 插入到正确的参数头文件中去(详情见第 4.6.4)。

图 9 显示了一个实际运行情况。该 MTPA 方法运行在速度控制回路中。在这种情况下,  $i_q^*$  (PI 调节器输出) 反馈给 MTPA 功能,  $i_d^*$  则选择进入线性插补轨迹。

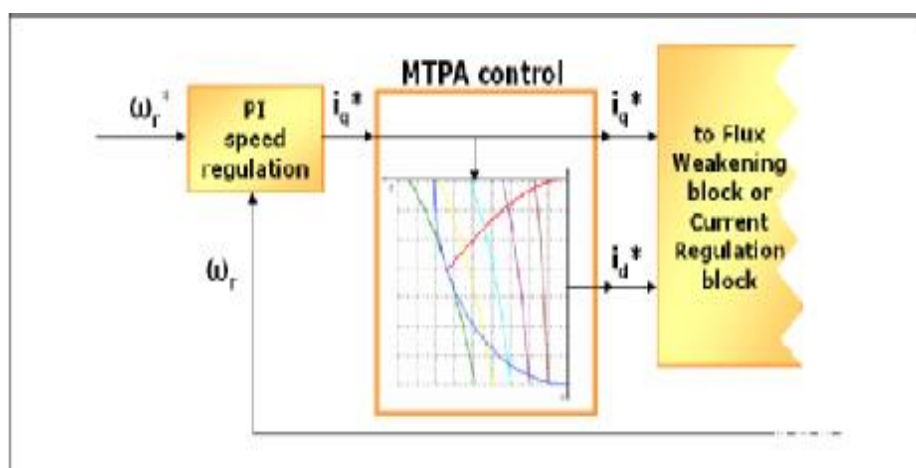


图 9 MTPA 控制

在任何情况下, 通过作用直轴电流  $i_{ds}$ , 磁通可以被削弱, 扩大速度变化范围。进入此运行区域, MTPA 轨迹变成左边(关于弱磁策略的详情见 2.3 节)。

总之, 通过调节分量电流  $i_{qs}$  和  $i_{ds}$  调节电机电流, FOC 可以控制 PMSM 转矩和磁通。然后通过“同步帧的 CR-PWM”的方法就实现了电流调节。

### 2.1.5 前馈电流调节

前馈功能是由固件库提供的，旨在改进电机驱动器 CR-PWM 部分的性能（脉冲宽度调制电流调节）。

基本上前馈功能预先计算定子电压  $V_q^*$  和  $V_d^*$  命令要求，参考电流  $i_q^{**}$  和  $i_d^{**}$ ，再送入电机中去。这样就会对标准 PID 电流调节有利（见图 10）。

前馈功能工作同在同步参照系的同时，也需要很好的了解机器参数，如绕组的电感  $L_d$  和  $L_q$ （或使用 SM-PMSM 时  $L_s$ ）以及电机恒定电压  $K_e$ 。

前馈算法的设计是为了弥补依赖频率的反电动势和永磁电机的交流耦合降压。因此，q 轴和 d 轴的 PID 电流控制回路成为线性的，就得到高性能的电流控制方法。

计算的定子电压  $v_q^*$  和  $v_d^*$  是根据当时测量的直流电压进行补偿的，上述方法还对总线电压进行纹波补偿。

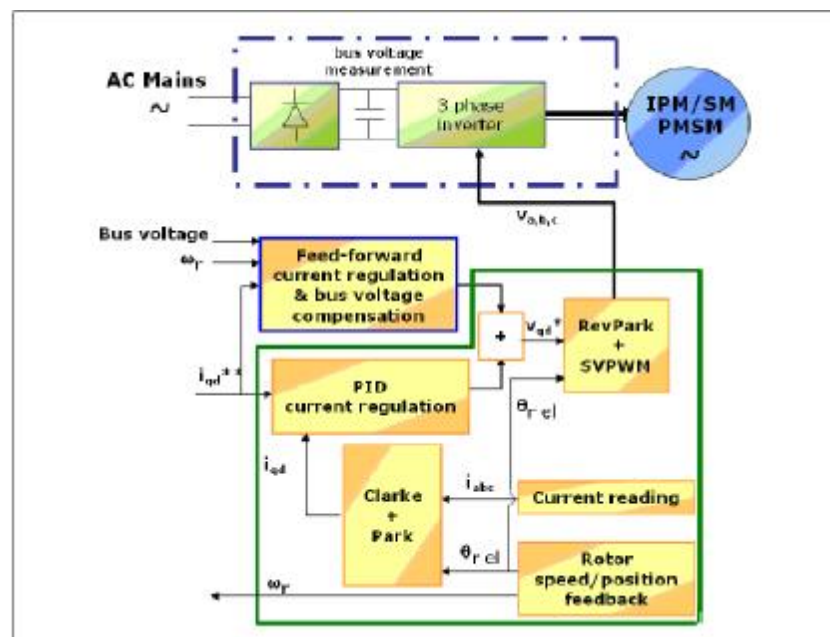


图 10 前馈电流调节

根据整个系统的一些参数，如直流电容大小，电频率，电机参数等，前馈功能对电机驱动器作用可能很大，或者很小。因此，建议用户仅当在测量有价值的结果时评估系统性能，启用前馈功能。

启动前馈功能见 4.1 节，4.6 节说明了如何在恰当的头文件中插入所需参数（使用 STM32MC-KIT\design tools\提供的 PMSM\_MTPA\_FEEDFORWARD.xls 电子表格），5.4 节描阐述了此功能及可用函数类型。



## 2.2 无传感器的转子位置/速度反馈简介

2.1 节已经表明,在永磁同步电机磁场定向控制中转子位置/速度测量有着至关重要的作用。为了测量转子位置/速度,霍尔传感器或编码器广泛应用于控制链中。

人们普遍认为无位置传感器关于转子位置/速度反馈的算法非常有用，而且都有各种理由：降低应用总成本，通过冗余性提高可靠性等。

固件库基于状态观测理论，为无传感器检测转子位置/速度反馈提供了完整的解决方案。此算法可用于 SM - PM 和 IPM 同步电动机，如参考文献[5]解释的一样（附录 A.9：参考文献）。运行的转子磁通观察器和经典第六评估程序[6]（附录 A.9：参考文献）之间的理论与实践对比已经指出观察器的优势，明显减少对定子电阻的变化和参数变化整体稳固性的依赖。

在控制理论中一个状态观察器就是一个体系，一个提供了评估内部状态的实时系统，并测量其输入和输出值。

在本文例子中，电机的内部状态包括反电动势和相电流，而输入由相电压供给，输出由相电流压供给（见图 11）。

直流母线电压测量接收电压命令，并向电机施加相电压。

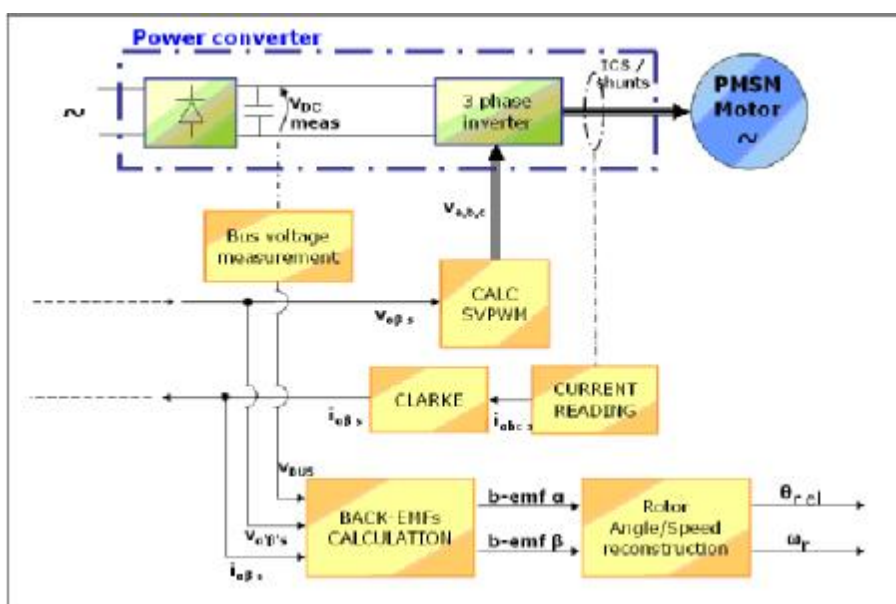


图 11 普通无传感器算法模块图

特别是，所观察到的状态通过相电流与实时系统一致性进行对比，其结果通过增益向量（K1，K2）来调整模块。

电机反电动势定义如下:

$$e_g = \Phi_m p \omega_r \cos(p \omega_r t)$$

$$e_{\beta} = -\Phi_m p \omega_r \sin(p \omega_r t)$$

可以看出，其中包含转子电角度。接着，反电动势送到作为 PLL 的模块，这个模块能够重构转子电角度和速度。

图 12 是电机在定向控制运行时（顺时针转动）局部截图；黄色的和红色的波形（C1，C2）分别代表观察到的反电动势  $\alpha$  和  $\beta$ ，蓝方波（C3）是安装在 a 轴上的霍尔传感器电池传来的信号，绿色正弦波是电流  $i_a$ （C4）。

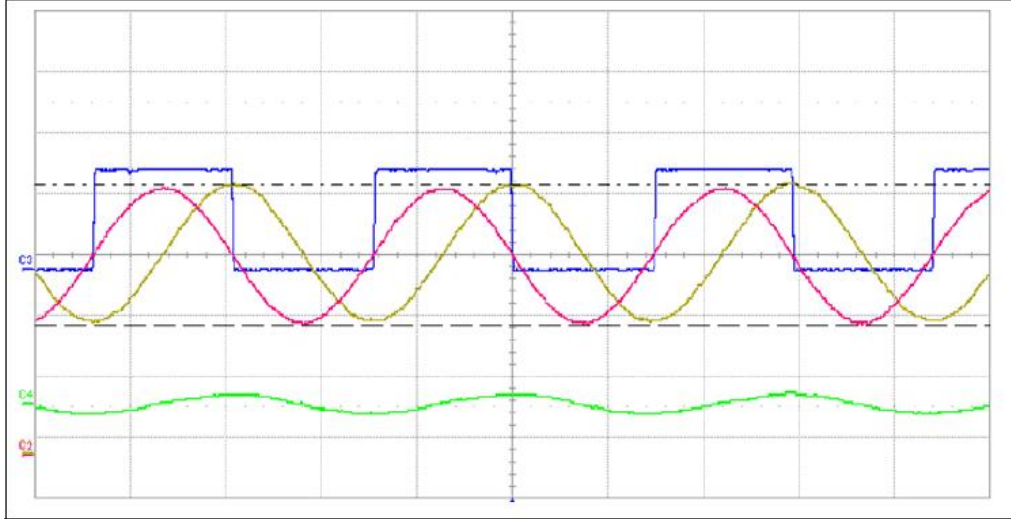


图 12 无传感器的状态观测算法检测的 PMSM 反电动势

更多关于如何设置参数使固件适应用户电机的信息见 4.5 节。

## 2.3 弱磁控制简介

在很多应用场合下永磁电机负载要比额定负载低，这时弱磁功能使永磁电机运转速度超过额定速度，从而达到扩大运行速度范围的目的。在这里，额定转速是在电机可以提供最大扭矩情况下的最高速度。

控制直轴电流  $i_d$  可以削弱磁通；给定电机额定电流  $I_n$ ，例如  $I_n = \sqrt{i_q^2 + i_d^2}$ ，如果让  $i_d \neq 0$ ，则最大可用正交电流  $i_q$  就会降低。类似的在 SM-PMSM 例子中，最大可传递电磁转矩也下降了，如 2.1.3 节展示的一样。另一方面，对于 IPM 电机来说，单独控制  $i_d$  会引起 MTPA 路径偏差（如 2.1.4 节说明一样）。

现在技术已经达到“闭环”弱磁控制，而且不需要知道电机参数具体值，这样就大大降低了对参数偏差的敏感性（见附录 A.9：参考文献中[3]-[4]）。这种方案对 IPMSMs 和 SM-PMSMs 都适用。

这种控制循环基于对定子电压进行监控（如图 13 所示）。

电流调节器输出  $V_s$  已经对照门槛值（“电压等级”参数）来进行校核。如果  $V_s$  超过限额，调节控制信号  $i_{fw}^*$  自动进入弱磁区域， $i_{fw}^*$  累积起来就是  $i_{ds}^*$ 。这可以通过 PI 调节器（增益可以在实时系统中进行调节，见 3.4 节）的方式实现，

也为了防止调节电流达到饱和值。很明显，设定的电压等级越高（通过保持电流规则），获得的效率也就越高，能实现的最大速度也越大。

如果  $V_s$  比所选的门槛电压小，那么  $i_{fw}$  就会减小到零，MTPA 模块就会重新恢复控制。

弱磁控制器输出电流的  $i_{ds}^{**}$  必须与  $i_{ds}$  进行核对比较，以避免电机的退磁。

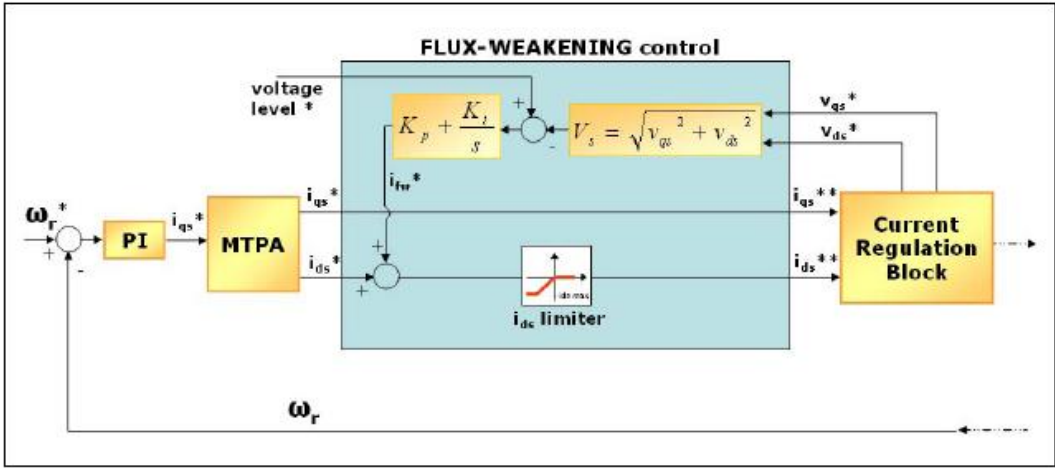


图 13 弱磁运行图

弱磁功能的应用见 4.1 节，在恰当的头文件中插入所需的参数见 4.6 节和功能描述及可用函数的类型见 5.4 节。

### 3 运行演示程序

#### 3.1 转矩控制模式

图 14，图 15 和图 16 显示了一些在转矩控制模式下的参数设置 LCD 菜单。红色突出显示的参数是可以设置的，其值也是可以可通过操纵杆键修改的。

按操作按钮向上/向下选择控制模式（图 14 例子是转矩控制）。一旦电机启动命令已发出（按下 JOY 或 KEY 键），这个参数就不能再修改了。当电机停止才能再次修改它。



图 14 液晶屏幕上的转矩控制设置

从前面的屏幕上（图 14）可以看出，如果操纵杆移动到右侧，目标电流分量  $I_q$  就会突出显示（变成红色）。这时可以向上/下移动操纵杆修改参数。一旦电机启动命令已经发出，测量电流分量  $I_q$  可以在运行时改变，同时测得  $I_q$  电流分量在测量区域显示出来。



图 15 液晶屏幕上的目标  $I_q$  设置

从前面的屏幕（图 15）可以看出，如果操纵杆移到右侧，目标电流分量  $I_d$  会突出显示（变成红色）。这个参数就可以通过上/下移动操纵杆进行修改。一旦电机启动命令已经发出，目标电流分量  $I_d$  可以在运行时改变，同时测得  $I_d$  电

流分量在测量区域显示出来。



图 16 液晶屏幕上的目标 Id 设定

通过按下 **KEY** 按钮或者操纵杆，电机就会停止运行（电机的主要状态从运行状态切换到停止状态）。

根据速度/位置反馈所利用的结构不同，在转矩控制模式下就会使用不同的电机斜坡上升方法：

- I 在配置文件 `stm32f10x_MCconf.h` 中取消了对 `ENCODER` 和 `VIEW_ENCODER_FEEDBACK` 的注释。在这种情况下，转子必须进行相位预置（也称为校准），以便通过正交编码器反馈相对位置信息。只有在测微控制器故障或复位之后首次启动才需进行校准。对此程序更深层次的描述参见 4.3 节。转子预定位完成后，如果编码器还未被注释，变量包括目标值 `iq` 和 `id` 定子电流分量（分别为 `hTorque_Reference` 和 `hFlux_Reference`）在 `MC_Control_Param.h` 头文件中用 `PID_TORQUE_REFERENCE` 和 `PID_FLUX_REFERENCE` 值进行初始化；主状态机从启动切换到运行状态。另一方面，如果取消了对 `VIEW_ENCODER_FEEDBACK` 的注释，与无传感器运行有关的斜坡式操作策略在定位结束后就开始运行。
- I 在配置文件 `stm32f10x_MCconf.h` 中取消了对 `HALL_SENSORS` 的注释。在这种例子中不能进行转子预定位，`hTorque_Reference` 软件由头文件 `MC_Control_Param.h` 中定义的 `PID_TORQUE_REFERENCE` 进行初始化。该软件中的变量包含转子电角度，转子电角度是根据三个霍尔传感器输出的数值进行的初始化，电机主状态从启动切换到运行状态。
- I 在 `stm32f10x_MCconf.h` 中取消了对 `NO_SPEED_SENSORS` 的注释。就无传感器电机驱动而言，为了使转子运转起来，需要特定的斜坡上升，这样无传感器算法才收敛于转子实际位置。4.5 节中会更详细的描述斜坡上升程序。



## 3.2 速度控制模式

图 17 和图 18 是两个 LCD 显示菜单，显示的是在速度控制模式下设置控制的参数。以红色显示的参数是可以通过操纵杆设置或修改其值的。

从图 17 所示的菜单屏幕显示来看，在电机停止运转时，可以通过操纵杆上/下移动实现从转矩控制切换到速度控制（反之亦然）。



图 17 速度控制的主要设置

从图 18 所示菜单画面来看，移动操纵杆到右边，可以选择目标速度（以红色显示的参数）。一旦选定，就可以通过上/下移动操纵杆递增/递减参数。按下操纵杆就可以启动电机。当电机开始运行后，仍然可以修改目标速度。



图 18 设定目标速度的液晶屏幕

和转矩控制模式一样，按下操纵杆或按键就可以启动或停止电机。

由于是在速度控制模式下，转矩和磁链参数（目标  $I_q$  和目标  $I_d$ ）是转矩和磁链控制器的输出，不能直接设置这些参数。但是，PID 调节器却可以实时调节，下面将会详细阐述。

根据速度/位置反馈利用的各种配置不同，电机速度控制模块使用了不同的斜坡上升方法：

- I 在配置文件 `stm32f10x_MCconf.h` 中未注释掉 `ENCODER` 或者 `VIEW_ENCODER_FEEDBACK`。正如在前面所述一样，在这种情况下需要对转子

进行位相预定（也称为校准）是必要的。详细描述参考 4.3 节。

转子预定位完成后，如果 ENCODER 未被注释，包含目标电流分量  $I_q$  和  $I_d$ （分别是 `hTorque_Reference` 和 `hFlux_Reference`）的变量由转矩和磁通控制模块驱动，电机主要状态从启动切换到运行状态。另一方面，如果 `VIEW_ENCODER_FEEDBACK` 未被注释掉，与无传感器相关的斜坡式方法相刚在预定位结束后就开始启动。

- I 在配置文件 `stm32f10x_MCconf.h` 中取消对 `HALL_SENSORS` 的注释。那么从启动指令发出时，`hTorque_Reference` 软件变量就由通链和转矩控制器模块驱动。同时包含转子电角度的软件变量也在三个霍尔传感器输出数值的基础上被初始化。最后，电机主状态从启动切换到运行状态。
- I 在 `stm32f10x_MCconf.h` 中取消对 `NO_SPEED_SENSORS` 的注释。在无传感器电机驱动情况下，一个特定的斜坡上升是必要的，以便于转子启动，保证无传感器算法收敛到实际转子位置。关于斜坡上升程序详细描述见 4.5 节。

### 3.3 电流和速度调节器的调整

正如在 2.1 节揭露，在 `stm32f10x_MCconf.h` 中取消对 `DIFFERENTIAL_TERM_ENABLED` 定义的注释，就可以屏蔽两个 PID 控制器的微分酌，调节这两个 PID 控制器就可以实现调节  $I_q$  和  $I_d$  电流。下面两个图展示的是两个 LCD 菜单，菜单上允许实时的比例、积分调节，图中是正在进行微分增益调节。

图 19 显示的是选择转矩 PID 系数，同时图 20 显示的是选择磁通量 PID 的系数。从两个屏幕上来看，通过向右/左移动操纵杆就可以选择 P，I 或 D 系数（当存在），选中了就以红色显示。然后，再上/下按动操纵杆就可以改变某个值（增加或减小）。

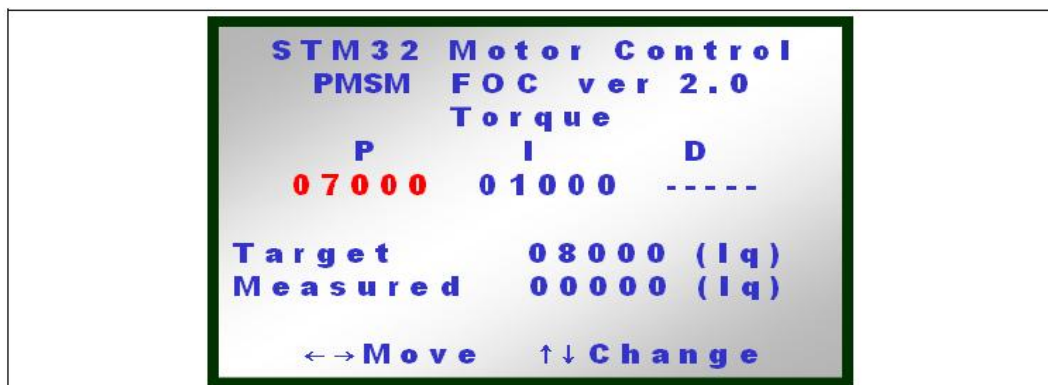


图 19LCD 显示的转矩 PID 的 P 项设置



图 20 LCD 显示的磁通量 PID 的 P 项设置

此外，为了实现在速度控制模块中的速度控制，在转矩和磁通控制模块还施行 PI（D）控制。由专门的 LCD 菜单也可以实时调节相关增益：

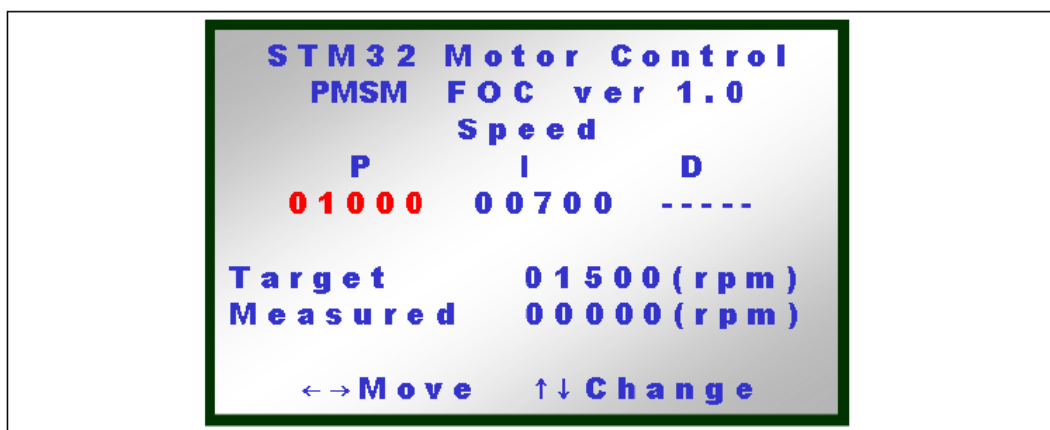


图 21 LCD 显示的速度 PID 的 P 项设置

在上面的菜单中，左/右移动操纵杆就可以选择 P，I 和 D（当可选时）系数，选中了就以红色显示。上/下操纵杆就可以改变（递增或递减）所需的值。

### 3.4 弱磁 PI 控制器的调节

如果弱磁功能已在 stm32F10x\_MCconf.h 文件中激活，下面的菜单就是可用的（见 2.3 和 4.1 节）。

下面的菜单可以实时调节弱磁模块内部 PI 调节器的比例和积分增益。

无论是 P 系数，I 系数还是定子目标电压 Vs 都可以通过左/右移动操纵杆选择选中就以红色显示。上/下移动操纵杆就可以改变（递增或递减）所需的值。图 22 显示了调节界面。



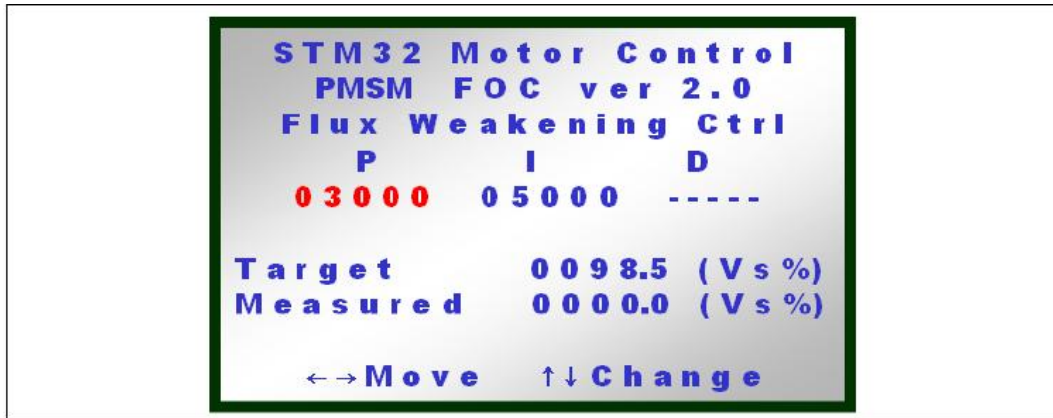


图 22 LCD 显示的弱磁 PI 的 P 项设置

在屏幕下部显示的是目标电压和测量定子电压  $V_s = \sqrt{n_a^2 + n_b^2}$  与最大可用相电压的百分比。

### 3.5 观测器和 PLL 增益调节

在固件库的默认配置中是禁止调节无传感器算法的。然而，当 stm32F10x\_MCconf.h 配置头文件未注释 OBSERVER\_GAIN\_TUNING 定义时，下图展示在就是专门的 LCD 菜单。



图 23 LCD 显示的磁通 PID 的 P 设置

在图 23 所示的菜单中，可以向左/右移动操纵杆来浏览不同增益值。向上/下移动下操纵杆可以递增/递减红色标示的增益。

此菜单可以实时改变观测器和 PLL 增益。当与 DAC 功能结合或者与霍尔效应传感器或编码器固件配置结合使用时，此功能会非常有用。在这种方式下，通过转子电角度的观察和测量和调整增益来取消两波形之间的误差就可以修改观测器和 PLL 增益。

### 3.6 DAC 的功能

在 stm32F10x\_MCconf.h 中启用 DAC 功能后，DAC 就是一个功能强大的调试工具，它允许使用专用的菜单最多同时可以追踪多两个软件的实时变量。



图 24 LCD 显示的磁通 PID 的 P 设置

在图 24 所显示的菜单中，可以左/右移动操纵杆来选择所需的微控制器引脚。要更改软件变量输出，可以向上/下移动操纵杆（可用变量列表取决于所选固件配置）。至于所有其他的菜单，按下操纵杆或 Key 按钮会启动/停止电机。

下面的代码可以将两个用户定义的变量添加到默认列表：

```
...  
#include "stm32f10x_MClib.h"  
...  
MCDAC_Update_Value(USER_1, variable_name1);  
MCDAC_Update_Value(USER_2, variable_name2);  
...
```

如果在显示屏上选择了“User1”或“User2”，就会输出显示这些变量。如果演示程序在开始/运行状态，就可以启动变量跟踪。FOC 算法采样频率可以修复更新频率（详情见 4.2）。

使用四个 TIM3 输出比较通道中的两个（PB0 和 PB1 引脚），调节 30kHz PWM 循环信号，就能在目前的固件库中运行 DAC 功能。在没有启用波形延时情况下，为了恰当的过滤产生的信号，建议使用恰当的一阶低通滤波器（如 10kΩ 的电阻和 22nF 的电容器构成的滤波器）。此外，如果使用具有高密度的性能线（即 STM32F103xC，STM32F103xD 或 STM32F103xE 衍生工具）的 MCU，用户可以适当修改 stm32f10x\_MCdac.c 文件来开发内置 2 通道、12 位 D/A 转换器。

### 3.7 电源反馈（Power stage feedbacks）

为了显示直流母线电压值和 STM3210B-MCKIT 电源板散热器温度，必须专门

设计一个显示菜单，如下图所示：

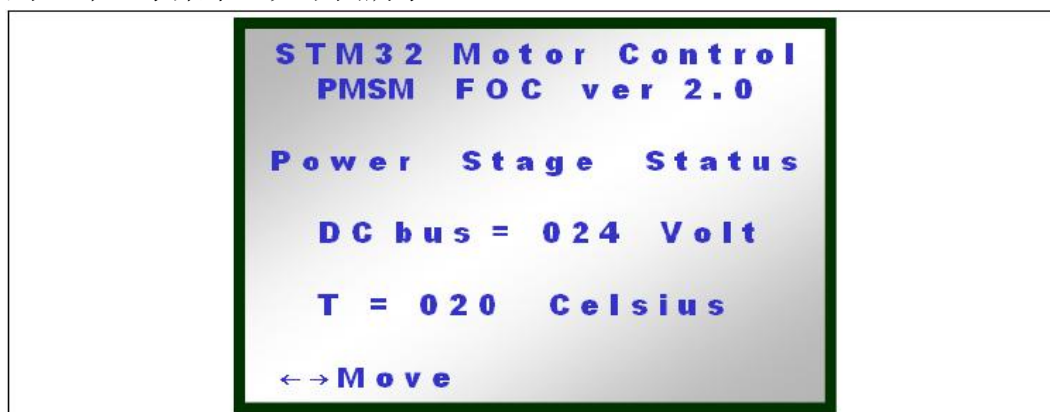


图 25 电源状态

### 3.8 故障信息

本节主要介绍所在软件库和 STM3210B- MCKIT 同时使用时，可能检测到的故障信息。图 26 显示了一个会在 LCD 上显示的典型错误消息。



图 26 欠压故障时会显示的错误信息

只有在故障源消除后，才会显示“按‘Key’返回主菜单”信息。在这种情况下，按下‘Key’键会使电机从故障状态切换到正常状态。

当固件库与 STM3210B-MCKIT 同时使用时，会有六种不同的故障：

#### 3.8.1 过流

PWM 外设专用引脚（BKIN）可以检测到低电平。如果这时正在使用 STM3210B-MCKIT，这意味着可能是硬件过热保护或硬件过流保护被触发了。详情参阅 STM3210B-MCKIT 用户手册。

### 3.8.2 过热

专用模拟通道可以检测过热。请在 `MC_Control_Param.h` 头文件中声明干预阈值 (`NTC_THRESHOLD_C`) 和相关滞后 (`NTC_HYSTERESIS_C`)。详情参阅 STM3210B-MCKIT 用户手册。

### 3.8.3 总线过压

只有在 `stm32f10x_MCconf.h` 配置头文件中注释 `BRAKE_RESISTOR` 定义(默认设置)，总线过压功能才可用。这意味着专用模拟通道才能检测过压。请在 `MC_Control_Param.h` 头文件中声明干预阈值 (`OVERVOLTAGE_THRESHOLD_V`)。详情参阅 STM3210B-MCKIT 用户手册。

注意：如果在 `stm32f10x_MCconf.h` 中取消了对 `BRAKE_RESISTOR` 定义的注释，这就会假定一个具有高功率耗散的电阻通过开关与总线电容并联。在这种情况下过压就不会产生故障提示，因为高功率耗散电阻通过总线电容会具有过电压。制动电阻原理详情参阅 3.14 节。

### 3.8.4 总线欠压

低于 20V 等效直流电压就是总线欠压。在 `MC_Control_Param.h` 头文件是由 `UNDERVOLTAGE_THRESHOLD_V` 参数设置的这个阈值。见 STM3210B-MCKIT 用户手册。

### 3.8.5 启动失败

仅当取消对 `NO_SPEED_SENSORS` 注释时才会产生启动失败提示。启动失败指在电机斜坡上升时没有检测到启动输出条件（在 `MC_State_Observer_param.h` 中的 `FREQ_START_UP_DURATION`）。见 4.5.2 节。

### 3.8.6 错误的速度反馈

同样要注意速度/位置反馈错误。根据使用的反馈不同，这可能意味着：

- I 如果正在使用编码器，测量速度连续增，等于或超过 `MAXIMUM_ERROR_NUMBER`（可以在 `MC_encoder_param.h` 中找到这些参数）时，就会超出所允许的范围（`[MINIMUM_MECHANICAL_SPEED_RPM; MAXIMUM_MECHANICAL_SPEED_RPM]`）。比如，编码器失去连接就会产生反馈错误。见 4.3 节。
- I 在霍尔传感器的配置下，利用定时器与三个霍尔效应传感器连接，就会连续溢出 `HALL_MAX_OVERFLOW`s 几次（`MC_hall_param.h`），正如在 4.4 节提到和

在第 5.7 解释的一样。这通常表示信息已经丢失（霍尔传感器超时）或者速度急剧下降。

- I 在无传感器运行情况下，速度测量就均值分布而言效果不理想。这通常意味着要么观察器调试不当，要么就是速度很低，以至于不可能得到良好的诱导 B-emf 观测数据值（例如转子锁定）。见 4.5.3 节。

### 3.9 在单分流拓扑结构和使用 MB459B 电路板的系统设置

为了将 MB459B 电路板设置为单分流拓扑结构，唯一所需的操作是将移动 W4，W5 和 W10 中导线的占位到图 27 所示：

- I W4 断开
- I W5 连接
- I W10 的 1-2 之间（与丝网不同）

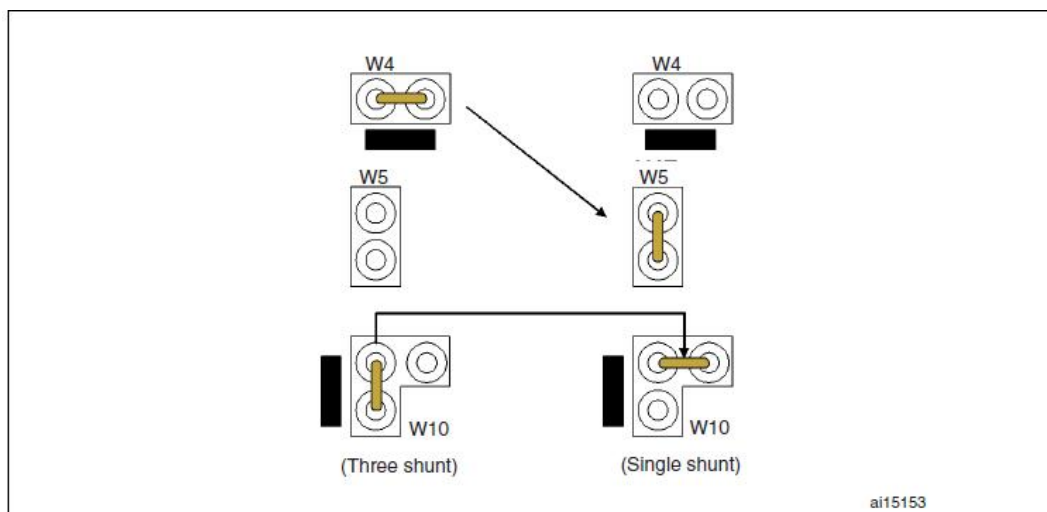


图 27 单分流拓扑结构的 MB459B 电路板设置

### 3.10 使用 ICS 时的系统设置

默认配置提供使用三个分流电阻器，不提供速度传感器。本节为您提供有关具有 ICS 反馈信号的 STM32F103xx 讯息以及如何正确配置固件的信息。

注意事项：

当使用两个 ICS 读取定子电流时，必须确保传感器输出信号条件范围与 STM32F103xx 供给电压兼容。

为了实现 FOC 算法的正常工作，必须确保该 stm32f10x\_svpwm\_ics 软件模块和两个 ICS 硬件连接相匹配。

如图 28 所示，由 STM32F103xx PWM 的信号 PWM1（A 相）和 PWM2（B 项）驱动的逆变器输出相电流，两个 ICS 必须作为变换器作用于这电机相电流。特别是，

从逆变器 A 项输出的电流必须由一路 ICS 读取，ICS 的输出必须由 PHASE\_A\_ADC\_CHANNEL 参数 MC\_pwm\_ics\_prm.h 发送到模拟通道。同样，从逆变器 B 项输出的电流必须由其它 ICS 读取，其它 ICS 的输出必须由 PHASE\_B\_ADC\_CHANNEL 参数 MC\_pwm\_ics\_prm.h 发送到模拟通道。

关于电流正向惯例，会在 PHASE\_X\_ADC\_CHANNEL 产生正半波，对应的正半波从相关逆变器的管脚输出（见图 28 中 I 向）。

注意：ICS 的电流测量固件激活见 4.1 节。使用 ICS 允许 MMI 设置为 100%（#define MAX\_MODULATION\_100\_PER\_CENT，见 4.2 和 5.5.3 节）。

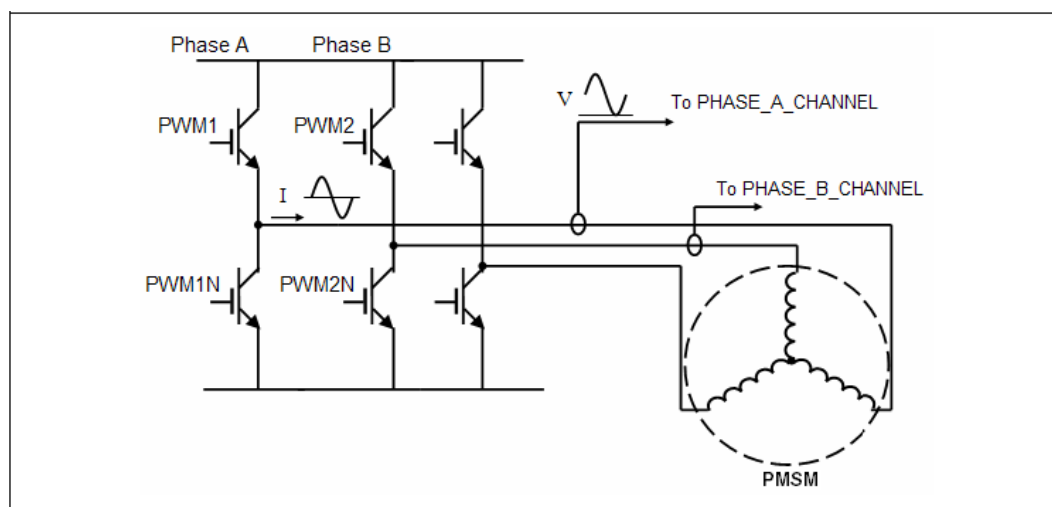


图 28 ICS 的硬件连接

### 3.10.1 PHASE\_A\_ADC\_CHANNEL 和 PHASE\_B\_ADC\_CHANNEL 的选择

PHASE\_A\_ADC\_CHANNEL 和 PHASE\_B\_ADC\_CHANNEL 的默认设置分别为 ADC\_CHANNEL11 和 ADC\_CHANNEL12。如果硬件需要，您可以更改默认设置，通过 MC\_pwm\_ics\_prm.h 文件“电流读数参数”编辑就可以修改。

举例来说，为了把 X 项（X=A, B）的电流反馈传给 ADC 通道 0，按下面方式编辑相关参数：

```
#define PHASE_X_ADC_CHANNEL ADC_Channel_0
#define PHASE_X_GPIO_PORT GPIOA
#define PHASE_X_GPIO_PIN GPIO_Pin_0
```

## 3.11 使用编码器时的系统设置

读取电动机转子的位置广泛使用正交增量编码器的方法。

正如本节标题一样，增量式编码器读取相对于初始位置的角位移：如果知道相对角位移，绝对角位移也就知道了。基于这个原因，在处理编码器反馈（编码

器或 VIEW\_ENCODER\_FEEDBACK 定义，而不是在评论 stm32f10x\_MCconf.h) 过程中，必须要在故障事件或微控制器的复位之后，电机初始启动之前进行转子的预定位。

增量式编码器有两个输出信号（图 29 中 TI1 和 TI2）。有了这些，再加上接口模式的 STM32F103xx 标准定时器，就可能获取旋转方向的信息。

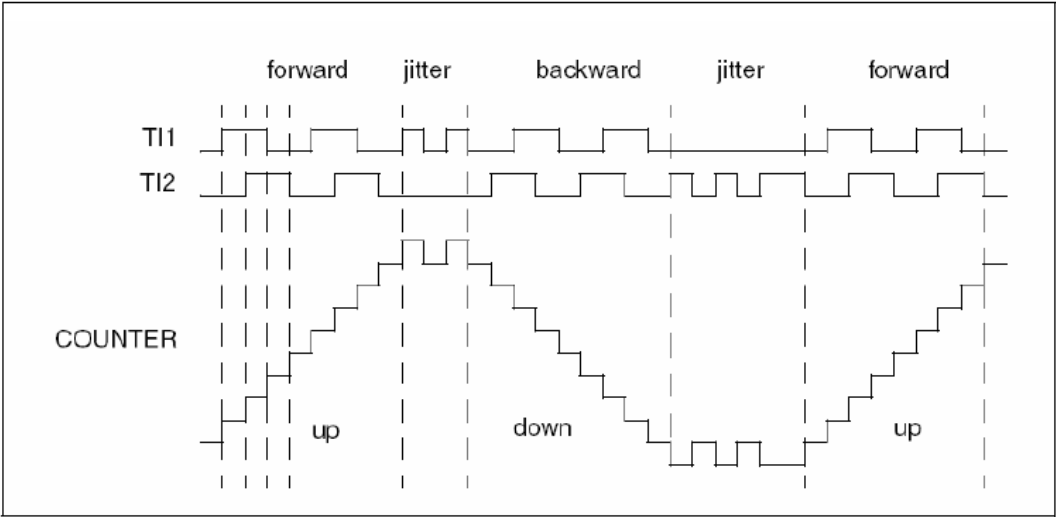


图 29 编码器输出信号：计数器操作

此外，转子角位移对时间进行微分就可以很容易地得到角速度可。

为了设置永磁同步电机磁场定向控制软件和库增量编码器一起使用，只需分别按照 4.1 和 4.3 节给出的指示简单地修改 stm32f10x\_MCconf.h 和 MC\_encoder\_param.h 头文件。

然而，需要额外注意的是什么是正向旋转：这个软件库假定三相电流正向供给的电机的旋转方向是正向。

由于这一点以及编码器输出信号与微控制器的输入引脚连接方式，真正的旋转方向和读取的方向可能会存在差异。为了避免这种误差，请按照下列步骤操作：

1. 按假定的正向手动转动转子，观察三相电机产生的 B- emf。为此，如果没有找到真正的中性点，就需要用三个电阻重构中性点。
2. 按照正相序把电机三相与硬件连接起来（例如，当使用 MB459 板时，电机的三相按正序连接到 J5 2, 1 和 3）。
3. 在有编码器配置下运行固件，按正向手工转动转子。如果液晶显示屏上显示的测量速度是正向，连接就是正确的；如果不然，则可只需简单地对编码器输出信号进行交换和重新布线。

如果还不可行，可通过软件设置修改：在 stm32f10x\_encoder.c 文件，把 164 行的代码替换掉：

把 TIM\_ICPolarity\_Rising , TIM\_ICPolarity\_Rising ) ; 换成 TIM\_ICPolarity\_Rising ,



TIM\_ICPolarity\_Falling) ;

### 3.12 使用霍尔传感器时的系统设置

霍尔传感器是可以识别转子磁场极性的传感装置；霍尔传感器根据其面对的磁极，输出逻辑电平 0 或 1，因此霍尔传感器必须按装在转子上。

通常在三相永磁电机中安装三个霍尔传感器，用来反馈转子位置信息。它们通常间隔  $120^\circ$  或者  $60^\circ$ ，本手册提到的固件库这两种情况都支持。为了使永磁同步电机磁场定向控制软件库能和三个霍尔传感器一起使用，只需要分别按照 4.1 和 4.4 节提示修改 `stm32f10x_MCconf.h` 和 `MC_hall_param.h` 头文件。

图 30 是间隔  $60^\circ$  和  $120^\circ$  安装的霍尔传感器输出的典型波形。需要注意的是，图 30 是一个周期的（即一对磁极电机转动一个周期）。

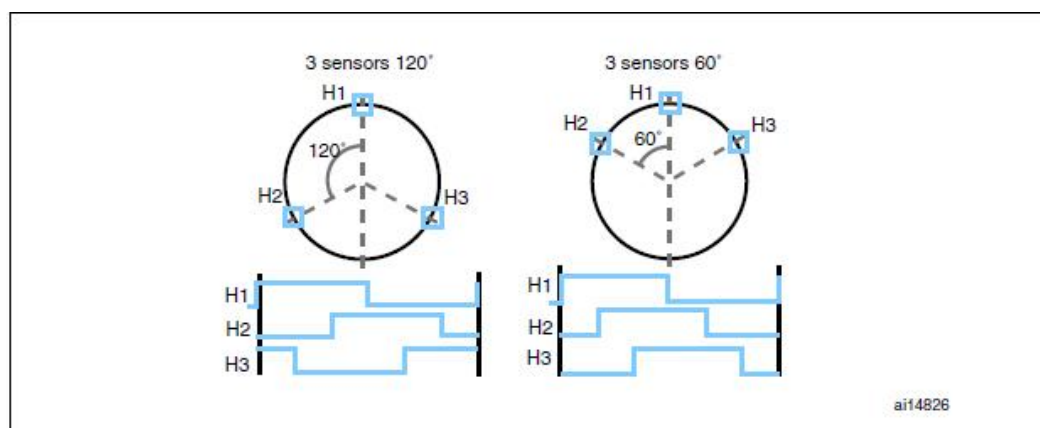


图 30 间隔  $60^\circ$  和  $120^\circ$  安装的霍尔传感器输出波形

由于霍尔传感器测得的转子位置是绝对的，不需要对转子预定位。然而，特别要注意是当传感器连接到微控制器输入时，需要进行预定位。

事实上如第 3.11 节指出，这个软件库假定的条件是正向转动是正序三相供给电机的时候的转动方向。在这种情况下为了保证正常工作，软件库要求间隔  $60^\circ$  和  $120^\circ$  间隔安装的霍尔传感器以图 30 所示顺序进行信号传输。

基于这些原因，当安装有霍尔传感器的永磁电机与你的板子相连时采取以下建议：

1. 按假定的方向手动转动转子，并观察引起的三相电动机的 B-emf。为此，如果真正中性点不可用，可通过三个电阻器的方法重新确立中性点。
2. 按照正序将电机连接到硬件。让“A相“,”B相“和”C相“成为由 TIM1\_CH1, TIM1\_CH2 和 TIM1\_CH3 各自驱动的电机的（例如当使用 MB459 板时，连接到 J5 2, 1 和 3 才是正电机相序）。
3. 按假定的方向手动转动转子，观察三个霍尔传感器输出（H1, H2 和 H3），并通过通道 1, 2 和 3 分别与对应的定时器相连，确保参照了图 30 所示的顺序。



序。

4. 测量在 A 相产生的最大 B-emf 和 H1 的第一个上升沿之间的电角度延迟。写到 MC\_hall\_param.h 头文件中 (HALL\_PHASE\_SHIFT)。为了方便, 图 31 所示与 HALL\_PHASE\_SHIFT 等于  $-90^\circ$  的例子。

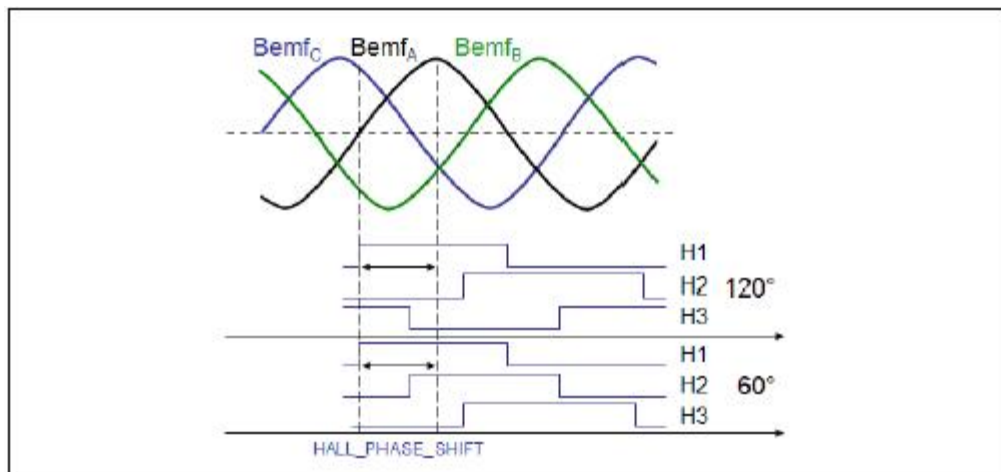


图 31 霍尔电相移的测定

### 3.13 先进无传感器系统开发

为了简化无传感器的开发, 在设计现有的固件库时定义了开发程序。这个过程有四个步骤, 使用户在很短的时间内在没有任何速度或位置反馈传感器的情况下实现电机运行。

定义开发过程是以在设计的阶段为前提, 用户可以利用从霍尔传感器或编码器送来的速度/位置信息。即使在应用中不可能或者不方便, 还是足以将电机轴机械连接到安装在电机上的一个传感器 (如在 STM3210B- MCKIT 中提供的直流无刷电机)。如果处于对比无传感器算法观察到的转子位置与实际值的目的, 反馈信息确实还是必要的。这样一来, 无传感器测算法可以可以得到很好的校正。

如上所述, 下面描述的就是 4 个步骤:

1. 在纯传感器配置中运行电机:

为了达到这个目的, 你应该:

- 根据附录 A.4 和 A.5 所给指示调整  $I_q$  和  $I_d$  电流回路调节器增益。
- 注释 FLUX\_TORQUE\_PIDs\_TUNING, 控制电机运行速度和调节速度 PI (D) 增益。在 stm32f10x\_MCconf.h 中通过取消注释 / 注释 DIFFERENTIAL\_TERM\_ENABLED 这行的方法来激活或者取消微分。

2. 在传感器模式下运行电机, 调节观测器增益

Clark 和 Parke 转换模块与速度调节器会用到传感器读取的转子的位置和速度信息。无传感器算法会并联运行。为了达到这个目的, 必须要:

- 根据附录中 A.6 提示预先计算观察器增益
  - 在 `stm32f10x_MCconf.h` 中取消注释 `OBSERVER_GAIN_TUNING`。
  - 填写 `MC_PMSM_motor_param.h` (见 4.6) 和 `MC_State_Observer_param.h` (启动部分不要求这一步, 见 4.5 节)。
  - 对比 (通过 DAC 功能) 所观察到的和实际的  $I_\alpha$  和  $I_\beta$  (误差必须为空), 并确保该观察到的 B-emf 波形清晰, 接近正弦, 实时调节观测器增益。不要求实时调节 PLL 增益, 但是可以按比例增加  $K_p$  和  $K_i$ , 扩大 PLL 带宽 (加快速度), 反之亦然。
  - 一旦得到 4 倍增益, 把他们写在 `MC_State_Observer_param.h` 文件中。
3. 在无传感器情况下电机, 处理传感器反馈模式:
- 转换模块和速度调节器可以利用来自无传感器算法 (“观察值”) 得出的转子位置和速度信息。传感器测量的信息可以与观测值进行比较。
- 本着这一目的, 必须:
- 取消对 `NO_SPEED_SENSORS` 的注释, 取消对 `VIEW_HALL_FEEDBACK` 和 `VIEW_ENCODER_FEEDBACK` 之间定义的注释 (取决于使用的传感器)
  - 填写 `MC_State_Observer_param.h` 中的启动部分, 如果电机不能在不同的电流和频率的启动参数下启动, 释放初始静态参数 (“速度反馈错误” 或 “启动失败”)

注意: 速度分辨率/精度不同, 高速 PID 需要不同的设置。

- 调节启动参数实验不同的斜坡启动, 如果需要, 进一步调节观测器和 PLL 增益
4. 恭喜! 现在电机能在无传感器纯模式下运行了:
- 根据所使用的调试功能要求, 对预期的代码大小和 CPU 的负载, 你可以:
- 在 `stm32f10x_MCconf.h` 中注释 `VIEW_HALL_FEEDBACK` 或 `VIEW_ENCODER_FEEDBACK` (取决于没有注释哪个就注释哪个)
  - 在 `stm32f10x_MCconf.h` 中对 `OBSERVER_GAIN_TUNING` 注释
  - 在 `stm32f10x_MCconf.h` 中中对 `DAC_FUNCTIONALITY` 注释。

### 3.14 使用制动电阻时的系统设置

由于实体结构的原因, 永磁同步电动机可以将动能转化成电能, 就像发电机一样。在有限条件下, 永磁同步电动机性质必须给予考虑, 以免可能损坏硬件系统。例如, 下列情况发生时可能会出现危险:

- I 六逆变器开关, 电机的运行速度超过了正常速度。在这种情况下, 线与线产生的 B-emf 电压幅值已经超过了正常的总线电压。

## I 控制系统试图制动，从负载到电路板出现能量转移

除非使用的电源系统有再生功能，要不然在上述两种情况下变频器的大电容就会充电。此外，根据转子速度（主要指第一种情况），或者转换能量大小（主要指第二种情况），电容器两端的电压会大到具有破坏性。

因此，以某种方式消耗掉产生的电能是十分必要的。为了达到此目的，可以实施不同的方法，但其中一个特别的方法就是利用制动电阻，本用户手册提供的库支持这种方法。

注意：如果电机运转超过额定速度，就必须使用再生电源变流器或制动电阻，以防止过电压损坏电路板。

### 3.14.1 如何对制动电阻运行配置 FOC 软件库

为了在 STM32F103xx 永磁同步电机磁场定向控制固件库中使用制动电阻，只需在 `stm32f10x_MCconf.h` 头文件中取消对 `BRAKE_RESISTOR` 注释。

STM32F103xx 的模拟看门狗功能会在在总线电压超过 `MC_Control_Param.h` 中声明的 `OVERVOLTAGE_THRESHOLD_V` 参数时产生中断，并导致 `BRAKE_GPIO_PORT` 端口的 `BRAKE_GPIO_PIN` 管脚变成高电平（都在 `MC_MotorControl_Layer.c` 中有定义，与 STM3210B- MCKIT 兼容的默认值是 `GPIOD` 和 `GPIO_Pin_13`）。从这时起，对总线电压要进行延滞环控制。在 `stm32f10x_it.c` 中编辑 `BRAKE_HYSTERESIS` 参数就可以进行迟滞操作。

### 3.14.2 如何对制动电阻运行修改 MB459 板

为了使 MB459 板适合制动电阻器运行，有必要在其外包装上连接一些其他组件。图 32 给出了硬件实现制动电路图。

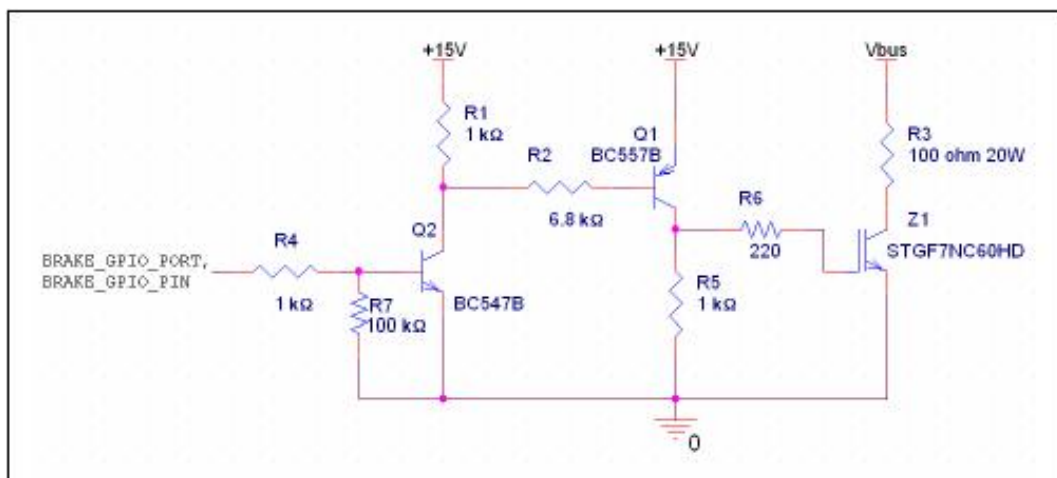


图 32 制动电阻电路

注意，电阻应仔细标上电阻值大小和持续功率。

当永磁同步电机磁场定向控制库与 STM3210B- MCKIT 一起使用时，注意 MC 连接器（J7 放在）23 引脚，它传输制动信号，离外包装领域比较近。

### 3.15 调试工具的注意事项

在使用于 MB535 JTAG 光电隔离板应用中不要使用第三方 JTAG 接口；调试工具可以保护 JTAG 接口和连接到上面的 PC 机。

注意：在使用 JTAG 接口开发固件断点时，应该禁用电机控制电子时钟电路；如果启用了，因为电机的 PWM 输出作用，直流电流就会流入电机，就会导致功率级/电机永久性损坏。在 DBGMCU\_SR 寄存器有一个专门位， DBG\_TIM1\_STOP 位，必须置为 1（见图 33）。

如果 DBG\_TIM1\_STOP 位被设置了（安全模式），定时器和 PWM 输出就会被关闭。这是逆变器的安全状态。定时器仍可以从停止的地方重新启动。

如果 DBG\_TIM1\_STOP 位被重置了（正常模式），定时器将继续正常运行，这时因为逆频器（中断服务不可用）的工作周期是恒定的，就可能引起危险。

在 main.c 模块中的 DBGMCU\_Config（DBGMCU\_TIM1\_STOP，启用）函数执行上述描述的任务。

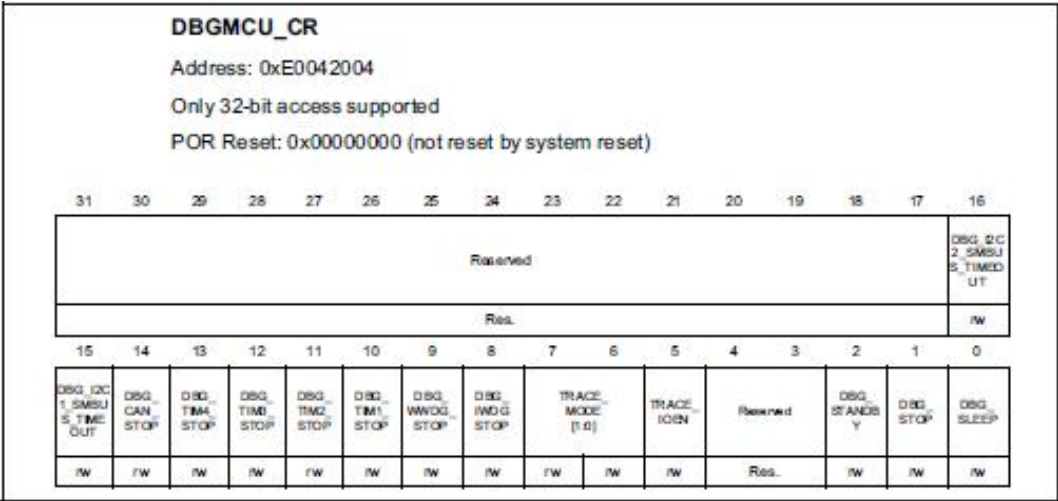


图 33 TIM1 控制寄存器的 DBG\_TIM1\_STOP 位（摘自的 STM32 参考手册）

## 4 库的入门

建立包含 STM3210B- MCKIT（具有 STM32F103xx 微控制器，运行软件）和永磁电机驱动系统操作平台很容易。

本节介绍如何快速配置系统，如有必要，还可根据需要定制相应的库。

按照以下步骤来完成：

1. 收集所有与使用的硬件有关的信息（电机参数，功率器件性能，速度/位置传感器参数，电流传感器跨导）；
2. 使用 IDE 编辑头文件 `stm32f10x_MCconf.h`（如 4.1 节详细阐述的一样）和下面的头文件参数，
  - `MC_Control_Param.h`（见 4.2 节），
  - `MC_encoder_param.h`（见 4.3 节）或 `MC_hall_prm.h`（见 4.4 节），
  - `MC_State_Observer_param.h`（见 4.5），
  - `MC_PMSM_motor_param.h`（见 4.6）；
3. 重新生成项目并下载到 STM32F103xx 微控制器上。

请注意，即使最终目的是无传感器驱动也应遵循此过程。事实上，从任何种类传感器都能接收速度/位置信息，这就能实现更加精确的开发，修改无传感器转子位置重构利用的算法。

### 4.1 库的配置文件：stm32f10x\_MCconf.h

这个文件的目的是声明具有编译条件的密钥，也在整个库编译过程使用：

- l 选择在实际上使用的电流测量技术（根据情况是在单间分流、三相分流或 ICS 传感器之间选择）。
- l 选择实际使用得速度/位置传感器（根据情况是在霍尔传感器、增量正交编码器或无速度传感器之间选择）。在无传感器模式下，启动电机之前可以设置驱动来定位转子。
- l 嵌入式永磁电机（IPMSMs）启用或禁用优化驱动器。
- l 启用或禁用可以实现电机超过额定速度运转的弱磁功能。
- l 启用或禁用前馈电流调节功能。
- l 启用或禁用与预期性能和代码大小相符的 PID 控制器微分酌。
- l 根据对制动性能和最高速度的要求启用或禁用制动电阻功能。
- l 启用或禁用用于软件变量中最重要的实时跟踪功能的 DAC 虚拟通道 2。为了获得最佳的调试，用户应该保持启用此功能。
- l 启用或禁用用于 PID 调节的特定软件的运行。

- I 启用或禁用状态观测器和 PLL 增益的调节。
- I 在“无速度传感器”情况下，也可以获得来自霍尔传感器或编码器的信号，并用 DAC 功能来评估无传感器算法工作是否正常。这些信号不会用于磁场定向控制算法；可以霍尔传感器，正交增量编码器选择，也可以不用。

如果头文件编辑不恰当（没有选择或未定义选择），在新建工程时，会弹出错误信息。注意如果在头文件定义的配置与实际使用硬件不匹配，或接线错误，将不会有错误信息提示。

具体来说：

- I `#define ICS_SENSORS`  
当使用隔离电流传感器完成电流采样时，要取消注释。
- I `#define THREE_SHUNT`  
当通过三相分流电阻进行完成电流采样时（默认情况），要取消注释。
- I `#define SINGLE_SHUNT`  
当通过单相分流电阻进行完成电流采样时（这需要按照 3.9 节所述设置 MB459 功率电路板：在单相扩扑结构中设置系统，按照 34 页使用 MB459 功率电路板），要取消注释。
- I `#define ENCODER`  
当增量编码器连接到位置测量启动位时取消注释；并填写 `MC_encoder_param.h`（如 4.3 节所述）。
- I `#define HALL_SENSORS`  
当三个霍尔传感器（60° 或 120° 放置）用于检测转子速度时取消注释；并填写 `MC_hall_prm.h`（如 4.4 节所述）。
- I `#define NO_SPEED_SENSORS`  
取消注释在 FOC 算法内使用转子位置信息（状态观测器提供转子位置信息）（默认设置）。在这种情况下，用户应并填写 `MC_State_Observer_param.h` 和 `MC_PMSM_motor_param.h`（如 4.5 和 4.6 节分别说明的一样）。
- I `#define VIEW_HALL_FEEDBACK`  
只有在与 `NO_SPEED_SENSORS` 结合时要取消注释。这是用来处理来自三个霍尔传感器的信息，通过 DAC 功能显示（并与来自无传感器转子位置重建算法的信息相比）。
- I `#define VIEW_ENCODER_FEEDBACK`  
只有在与 `NO_SPEED_SENSORS` 结合时要取消注释。这是用来处理来自增量式编码器的信息，将通过 DAC 的功能显示（并与来自无传感器转子位置重建算法的信息相比）（默认设置）。
- I `#define NO_SPEED_SENSORS_ALIGNMENT`

只有在与 NO\_SPEED\_SENSORS 结合时要取消注释。在电机启动前都要使驱动器对转子定位。在这种情况下，用户也完成 MC\_State\_Observer\_param.h 部分（如 4.5 节所述）

**I #define IPMSM\_MTPA**

取消注释，利用优化的 MTPA（最大转矩每安培）驱动来设计嵌入式永磁同步电动机（IPMSMs）。在这种情况下，用户不得不完成 MC\_PMSM\_Motor\_param.h 中的一部分（见 2.1.4 节和 4.6 节）。注释这个#define（默认），演示程序会优化永磁同步电动机需要的矢量（如 2.1.3 节所述）

**I #define FLUX\_WEAKENING**

如果用户需求使用电机转速超过额定速度（默认），取消注释。并填写 MC\_PMSM\_Motor\_param.h（如 4.6 节说明）；使用弱磁功能时预防措施见 3.13 节。

**I #define FEED\_FORWARD\_CURRENT\_REGULATION**

为了备份具有前馈计算功能的标准 PID 电流调节，要取消注释。并填写 MC\_PMSM\_Motor\_param.h 中相应的章节（如 4.6 节所述）

**I #define BRAKE\_RESISTOR**

取消注释，激活电阻制动软件运行（更多应用 MB459 板的硬件修改信息参见 3.14.2）。

注意：为了避免任何功率级的伤害，在超过额定速度运行时必须强制性使用制动电阻器。

**I #define DIFFERENTIAL\_TERM\_ENABLED**

取消注释，激活 MC\_PID\_regulators 库模块中的 PID 调节功能不同项（见 5.9 节）。

**I #define FLUX\_TORQUE\_PIDs\_TUNING**

当利用转子位置传感器取消注释。生成专门用于转矩和磁通 PID 增益调节的软件。并填写 MC\_Control\_Param.h。

**I #define OBSERVER\_GAIN\_TUNING**

如果取消注释，就会激活在液晶屏幕上生成用于观测器和 PLL 增益调节的可视界面。

**I #define DAC\_FUNCTIONALITY**

取消注释，激活 DAC 功能。更多关于此功能的详细信息见 3.6 节。

一旦完成这些设置，就只要求将模块与项目连接；这意味着你不需要将 C 文件排除在编译外。

注意：当使用分流电阻测量电流，确保 REP\_RATE 参数（在 MC\_Control\_Param.h）设

置正确（见4.2节和A.2：对单相或三相分流电阻配置是基于PWM频率选择更新重复率，详细信息见136页）。

## 4.2 驱动控制参数：MC\_Control\_Param.h

MC\_Control\_Param.h 头文件相关参数：

- I 第 46 页的功率器件参数
- I 第 46 页的电流调节参数
- I 第 46 页的电源板保护阈值
- I 第 47 页的速度循环采样时间
- I 第 48 页的转速 PID 控制器初始化值
- I 第 49 页正交电流 PID 控制器初始化值
- I 第 49 页根据机械速度线性变化的 PID 常数

### 功率器件参数

- I #define PWM\_FREQ  
定义开关频率以赫兹为单位；并取消对与 PWM 频率选择对应的最大值允许的调制指数定义注释（MAX\_MODULATION\_XX\_PER\_CENT）。  
注意如果使用 ICS，可以忽略所选的 PWM 频率，允许 100% 调制指数。
- I #define DEADTIME\_NS  
定义失效时间以 ns 为单位，以避免直通调件。

### 电流调节参数

- I #define REP\_RATE  
根据下式定义定子电流采样频率，磁链和转矩 PID 调节器采样频率：

$$\text{磁链和转矩PID采样频率} = \frac{2 \cdot \text{PWM\_FREQ}}{\text{REP\_RATE} + 1}$$

事实上，没有理由没更新定子电流值就运行的FOC算法，也没有理由不运行FOC算法就改变定子电流，在建议定子电流采样频率和FOC算法运行频率相一致。

注意：如果通过分流电阻测量电流，REP\_RATE必须是单数（见A.2节：基于单相或三相分流电阻配置PWM频率重复率的选择，详见136页）；其值是8位长；

### 电源板保护阈值

- I #define NTC\_THRESHOLD\_C
- I #define NTC\_HYSTERIS\_C  
当软件库与 MB459 板一起使用时，这两个阈值（以 °C 表示）是用来设置功率器件工作温度范围（散热片上测量的温度）。特别是如果测量温度超过



NTC\_THRESHOLD\_C, 就会产生错误提示, 直到测得温度低于 NTC\_THRESHOLD\_C -NTC\_HYSTERESIS\_C 才会消除。

I #define OVERVOLTAGE\_THRESHOLD\_V

I #define UNDERVOLTAGE\_THRESHOLD\_V

当软件库和MB459板一起使用时, 这两个阈值(以伏表示)规定的是总线直流电压最低值和最高值。如果总线电压超过OVERVOLTAGE\_THRESHOLD\_V或者低与UNDERVOLTAGE\_THRESHOLD\_V, 就会产生错误信息, 直到总线电压保持在允许范围内。

I #define BUS\_ADC\_CONV\_RATIO

定义ADC输入电压与总线直流电压之比。

### 速度回路的采样时间

#define PID\_SPEED\_SAMPLING\_TIME

通过设定下面一个定义来选择速度调节循环频率:

#define PID\_SPEED\_SAMPLING\_500µs 0 //min 500µs

#define PID\_SPEED\_SAMPLING\_1ms 1

#define PID\_SPEED\_SAMPLING\_2ms 3 //(4-1)\*500µs=2ms

#define PID\_SPEED\_SAMPLING\_5ms 9

#define PID\_SPEED\_SAMPLING\_10ms 19

#define PID\_SPEED\_SAMPLING\_20ms 39

#define PID\_SPEED\_SAMPLING\_127ms 255 //max(255-1)\*500µs=127ms

注: 请注意演示程序中的速度调节循环频率要与弱磁和前馈功能(如果已启用)控制的频率相一致。

### 转速PID控制器初始化值

I #define PID\_SPEED\_REFERENCE\_RPM

在闭环模式下启动转子时转速定位值以rpm为单位定义;

I #define PID\_SPEED\_KP\_DEFAULT

速度循环调节比例值(16位, 从0到32767可调);

I #define PID\_SPEED\_KI\_DEFAULT

速度调节循环积分值(16位, 从0到32767可调);

I #define PID\_SPEED\_KD\_DEFAULT

速度调节循环微分值((16位, 从0到32767可调);

I #define SP\_KP\_DIV

速度调节循环比例增益的系数(无符号的16位双功率值)。

I #define SP\_KI\_DIV

速度调节循环积分增益系数（无符号的16位双功率值）。

l #define SP\_KDDIV

速度调节循环微分增益系数（无符号的16位双功率值）。

### 正交电流PID控制器的初始化值

见附录A. 4，从电机参数和要求的控制带宽的开始，说明了计算初始PI常数的方法。

l #define PID\_TORQUE\_REFERENCE

转矩控制模式下启动时，转矩（ $I_q$ ）的参考值，（有符号16位值）；

l #define PID\_TORQUE\_KP\_DEFAULT

转矩循环调节的比例值（有符号的16位，从0到32767）；

l #define PID\_TORQUE\_KI\_DEFAULT

转矩循环调节的积分值（有符号的16位，从0到32767）；

l #define PID\_TORQUE\_KD\_DEFAULT

转矩循环调节的微分值（有符号的16位，从0到32767）；

l #define PID\_FLUX\_REFERENCE

在转矩控制模式下启动时，磁通量（ $I_d$ ）的参考值（有符号16位值）；默认值是0；

l #define PID\_FLUX\_KP\_DEFAULT

磁通循环调节的比例值（有符号的16位，从0到32767）；

l #define PID\_FLUX\_KI\_DEFAULT

磁通循环调节的积分值（有符号的16位，从0到32767）；

l #define PID\_FLUX\_KD\_DEFAULT

磁通循环调节的微分值（有符号的16位，从0到32767）；

l #define SQUARE\_WAVE\_PERIOD

当在stm32f10x\_MCconf.h中取消对FLUX\_TORQUE\_PIDs\_TUNING注释时产生的参考转矩的方波（以ms为单位）；

l #define TF\_KP\_DIV

在电流循环调节中使用的比例增益系数（无符号的16位双功率值）。这个比例对应附录A. 4中的KpDIV参数；

l #define TF\_KI\_DIV

在电流循环调节中使用的积分增益系数（无符号16位双功率值）。这个比例对应附录A. 4中的KiDIV参数；

l #define TF\_KD\_DIV

在电流循环调节中使用的微分增益系数（无符号16位双功率值）。

根据机械速度线性变化的PID常数。

参照5.9.4节：对应123页的电机频率调节速度调节循环中的Ki, Kp和Kd。

### 4.3 增量编码器参数： MC\_encoder\_param.h

如果通过正交、方波、旋转编码器测量位置/速度传感器（在stm32f10x\_MCconf.h头文件中定义的ENCODER或VIEW\_ENCODER\_FEEDBACK），则需要完成MC\_encoder\_param.h头文件。

**I** #define TIMER2\_HANDLES\_ENCODER

如果两路传感器信号与TIMER2输入引脚连接（默认：要求使用STM3210B-MCKIT），取消注释；

**I** #define TIMER3\_HANDLES\_ENCODER

如果两路传感器信号与TIMER3输入引脚连接，取消注释；

**I** #define TIMER4\_HANDLES\_ENCODER

如果两路传感器信号与TIMER4输入引脚连接，取消注释；

**I** #define ENCODER\_PPR

定义编码器每转一周单通道产生的脉冲数(实际分辨率是4的倍数)；

**I** #define MINIMUM\_MECHANICAL\_SPEED\_RPM

定义转子最小机械转速(转数/分)，当小于它时，速度测量在应用中不具备可行性和安全性；测量速度低于设定值错误计数器就会累积。为了制止这种累计并控制电机速度到零，用户可以将参数设置为零：在编译时会出现一条警告信息，“无符号整数与零对比无意义”。

**I** #define MAXIMUM\_MECHANICAL\_SPEED\_RPM

定义转子最小机械转速(转数/分)，当小于它时，速度测量在应用中不具备可行性和安全性；测速高于设定值时，错误计数就会累积。

**I** #define MAXIMUM\_ERROR\_NUMBER

在产生错误讯息之前，定义速度测量连续检测出速度错误次数(stm32f10x\_Timebase.c中的SPEED\_MEAS\_TIMEBASE in指定检测率)。

**I** #define SPEED\_BUFFER\_SIZE

定义检测转速使用的缓冲区大小。2的倍数便于计算。

#### 校准设置：

正交编码一个相对位置传感器。考虑磁场定向控制要求绝对位置信息，必需以某种方式建立一个零角度位置。这可以通过校准相位完成，并在电机第一个启动之前进行，也在错误事件后进行。基本上包括以在固定方向线性增长的零参考转矩(Iq)和参考磁通(Id)。

如果在相位最后配置正确，定子就会被锁定在一个已知的位置，编码器定时器计数器也会相应的初始化。

依据电机的惯性和负载情况下面的参数用于制定校准相位：

- I T\_ALIGNMENT（单位：ms）定义相位校准时间；
- I ALIGNMENT\_ANGLE 规定向量方向（下图中  $\theta_0$  角）；

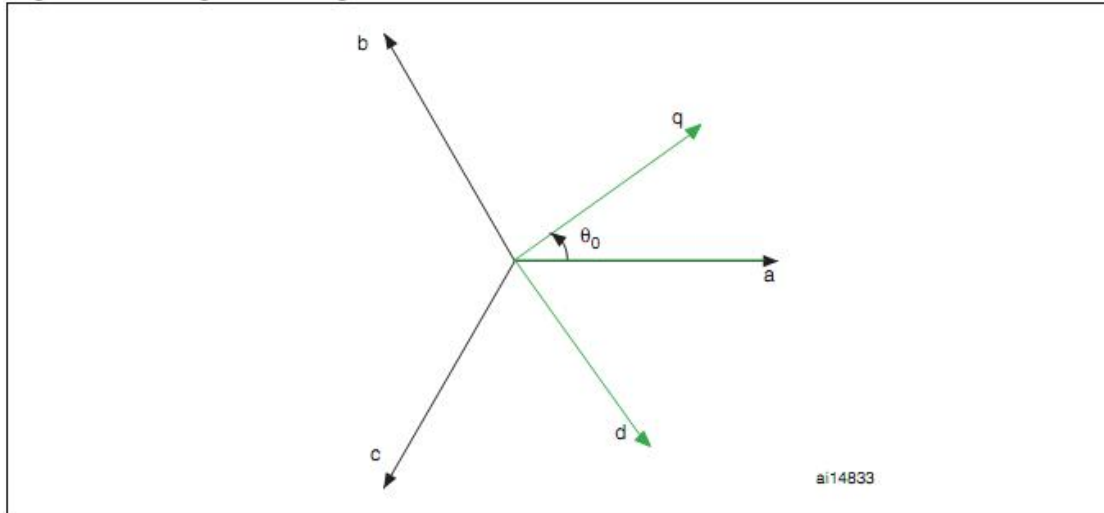


图 34 校准角度

- I I\_ALIGNMENT（数字化的）定义参考 Id 幅值的最终值。通过 STM3210B-MCKIT 和 ALIGNMENT\_ANGLE 设置为 90 度（默认），可以通过下式计算最后 A 相电流值：

$$\text{校准 A 相最终电流} = (I\_ALIGNMENT * 0.64) / (32767 * R_{shunt})$$

#### 4.4 霍尔传感器参数：MC\_hall\_prm.h

如果使用三路霍尔传感器测量位置/速度，就要完成 MC\_hall\_prm.h 头文件（在 stm32f10x\_MCconf.h 中定义 HALL\_SENSORS 或 VIEW\_HALL\_FEEDBACK）。

- I #define TIMER2\_HANDLES\_HALL  
如果三路传感器信号与 TIMER2 输入引脚相连，取消注释（默认设置；使用 STM3210B-MCKIT 的要求）；
- I #define TIMER3\_HANDLES\_HALL  
如果三路传感器信号与 TIMER3 输入引脚相连，取消注释；
- I #define TIMER4\_HANDLES\_HALL  
如果三路传感器信号与 TIMER4 输入引脚相连，取消注释
- I #define HALL\_SENSORS\_PLACEMENT  
以角度位单位定义霍尔传感器之间电位移（物理位移×磁极对数）。在 60°（DEGREES\_60）或 120°（DEGREES\_120）之间选择。

**I #define HALL\_PHASE\_SHIFT**

以角度定义 0° 与 TIMx\_CH1(STM32 MC-KIT 上的 H1 信号)第一个上升沿之间的电角度偏移。

如何确定这个参数，详情见 3.12 章节。

**I #define HALL\_MAX\_SPEED\_FDBK\_RPM**

定义转子最大机械转速（转数/分），当大于它时，速度反馈不可用：用于区别故障。

**I #define HALL\_MAX\_SPEED**

如果检测到速度大于 HALL\_MAX\_SPEED\_FDBK\_RPM 时，函数 HALL\_GetSpeed(单位：0.1Hz) 返回这个值。默认值是 500Hz，但可以通过上层软件设定成 0-FFFF。

**I #define HALL\_MAX\_PSEUDO\_SPEED**

如果检测到速度大于 HALL\_MAX\_SPEED\_FDBK\_RPM 时，函数 HALL\_GetRotorFreq 返回这个值。单位：dpp。从赫兹转化到伪频率详情见附录 A.7。

**I #define HALL\_MIN\_SPEED\_FDBK\_RPM**

定义转子最小机械转速(rpm)，当小于它时，速度反馈不可用。

**I #define HALL\_MAX\_RATIO**

-定义可以测量的最低速度（当计数=0xFFFF）

-当电机停止时，防止时钟分频器减小过度。（每个捕获中断以优化时钟分辨率时预分频器会自动调整）

**I #define HALL\_MAX\_OVERFLOW**

这是考虑到的定时器连续溢出的最大次数。默认设置为 2：当定时器出现两次以上溢出（意味霍尔传感器在连续有效边缘周期至少增大两倍），溢出数目将不再累加。这通常表明数据已经丢失（霍尔传感器超时）或速度的急剧下降。对应超时延迟根据选择定时器预分频器不同而不同，是一个变量；预分频器频率越高（低速），定时器超时时间越长（见 5.7 节）。

**I #define HALL\_SPEED\_FIFO\_SIZE**

这时存储最新测量的速度的软件 FIFO 长度。在连续数据块堆栈之间有必要计算滚动平均值。

## 4.5 状态观测器参数：MC\_State\_Observer\_param.h

如果用户想利用或评价无传感器算法检测转子位置/速度，就要完成 MC\_State\_Observer\_param.h。MC\_PMSM\_motor\_param.h 头文件需要相应的配置完成（见 4.6 节）。

同样见 3.13 节，指导用户逐步设置无传感系统并且启动运行。

涉及以下参数：

- I 状态观察器
- I 启动
- I 测量统计和可靠性

#### 4.5.1 状态观测器参数

##### I #define MAX\_CURRENT

定义等于 ADC 转化的电流值位 32767（最大 16 位），单位安培。如果电流是

通过分流电阻测得，则： $MAX\_CURRENT = \frac{3.3}{2 R_{shunt} A_n}$

其中：

- $A_v$  是运算放大器的增益(在 MB459B 工具箱板中  $A_v = 2.57$ )
- $R_{shunt}$  是分流电阻值（欧姆）

##### I #define K1

K1（32 位值）是状态观测器矢量增益参数（如章节 2.2 描述）。必须使用章节 A.5 中的公式中：“先验”决定 K1。当电机运行时，可通过 LCD 接口调节初始值，并评估结果。这样，K1 在 LCD 上显示的值小了 10 倍。

##### I #define K2

K2（32 位值）是状态观测器矢量增益参数（如章节 2.2 描述）。必须使用章节 A.5 中的公式中：“先验”决定 K2。当电机运行时，可通过 LCD 接口调节初始值，并评估结果。这样，K2 在 LCD 上显示的值小了 100 倍。

##### I #define PLL\_KP\_GAIN

默认公式提供 PLL 相位检测增益“先验”决定。然而，当电机运行时其值（16 位）可以调节的，通过 LCD 接口就可以调节，还可以进行评估。如果有必要，默认值可以修改位适当的值。

##### I #define PLL\_KI\_GAIN

默认公式提供 PLL 循环过滤增益“先验”决定。然而，当电机运行时其值（16 位）可以调节的，通过 LCD 接口就可以调节，还可以进行评估。如果有必要，默认值可以修改位适当的值。

##### I #define F1, #define F2

这些系数（16 位）可以放大观测器方程系数，以使电机绕组和电感产生作用。

最大值是  $2^{15}=32768$

注意：

根据多个系统参数（电机参数，状态观测器参数，采样频率，电流和电压转换参数），编译错误 [Pe068] 或 [Pe069] 可能出现在 MC\_State\_Observer\_Interface.c 源文件。这时，用户可遵循程序下面解决问题：

- 1 跳转到出错行，查看哪个 Cx 变量（x=1..5）引起错误。
- 2 转向 Cx 变量的定义（MC\_State\_Observer\_param.h）和评估是否涉及 F1 或 F2。
- 3 相应减小 F1 或 F2，选择 2 的倍数。

#### 4.5.2 启动参数

基于对转子运行时的反电动势的观测可重构转子位置。因此，当电机从静止状态启动时，为了使电机自旋需要制定启动步骤：三相对称电流在定子上产生旋转磁通，从而引起转子转动。

根据章节 4.5.3 参数（主要状态从启动到运行）当观测到的反电动势值变得可靠，就可以认为启动程序启动成功，否则产生定时器输出（这样主要状态从启动转为故障）。

通过设定三相电流系统幅值和频率曲线，这章节描述的参数可将启动用于实际应用（如图 35）

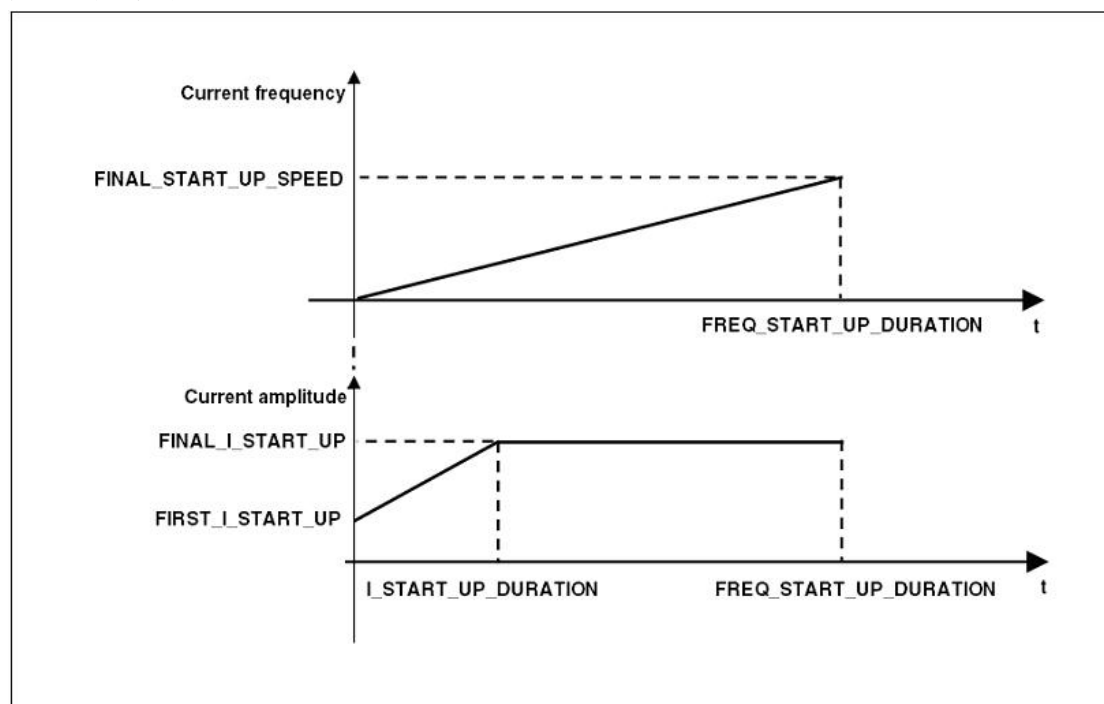


图 35 启动电流系统频率和幅值曲线

- I #define FREQ\_START\_UP\_DURATION  
定义启动允许的总时间，单位毫秒。
- I #define FINAL\_START\_UP\_SPEED  
定义旋转磁通的速度，启动的时间的最后的的速度，单位转数/分（这个参数

设定的是频率线性斜坡的斜率)。

I #define FIRST\_I\_STARTUP

数字量，按照以下计算公式定义三相电流系统的初始幅值：

$$I(\text{digit}) = \frac{I(\text{Amps}) \cdot R_{\text{shunt}} \cdot A_v \cdot 65536}{3.3}$$

I #define FINAL\_I\_STARTUP

数字量，按照上述计算公式定义三相电流系统的最终幅值。这幅值的选择应该使产生的磁场力矩与实际应用负载相匹配。

I #define I\_STARTUP\_DURATION

定义从幅值的初始到最后，线性电流幅值增长允许时间，单位毫秒。

I #define MINIMUM\_SPEED\_RPM

定义启动程序终止时的正常运行最小速度以（当转子速度/位置是可靠的），单位转数/分。

根据用户的实际应用要求，在每个电机启动前对转子校准或预定位。通过适当配置 stm32f10x\_MCconf.h，这些额外的功能是可作为代选项（参考章节 4.1）。

这样，下面的参数的选择以实现预期的功能：

- #define SLESS\_T\_ALIGNMENT

定义校准相位需要的持续时间，单位毫秒。

- #define SLESS\_ALIGNMENT\_ANGLE

指定矢量的方向（如图 34 角度  $\theta$ ）

- #define SLESS\_I\_ALIGNMENT

数字量，定义参考  $I_d$  的最终值大小。通过将 SLESS\_ALIGNMENT\_ANGLE 设置成  $90^\circ$ （默认）以及 STM3210B-MCKIT，A 相位的最终电流值通过公式计算：

对齐 A 相最后电流 =  $(\text{SLESS\_I\_ALIGNMENT} \times 0.64) / (32767 \times R_{\text{shunt}})$

### 4.5.3 统计参数

状态观测器输出实时监测统计评估转子速度/位置反馈数据的可靠性。由于这项技术用于系统的故障检测，这章节中参数的描述用于设定需求安全级别。

I #define VARIANCE\_THRESHOLD

参数设定速度测量方差的门槛。根据公式：

$s^2 \geq m^2 \text{ VARIANCE\_THRESHOLD}$ ，当方差观测速度方差大于平均值的百分比，就认为无传感算法是不可， $\sigma$  和  $\mu$  分别是方差和观测速度平均值（方差 VARIANCE\_THRESHOLD 是 0.0625 即平均值的百分比为 25%）。

VARIANCE\_THRESHOLD 的参数越小，故障检测算法就越严格（即安全级别越



高), 反之亦然。

l #define RELIABILITY\_HYSTERESIS

参数(8 位)定义在转子速度/位置检测方法声明不可靠前, 速度检测方差比 VARIANCE\_THRESHOLD 阈值高的连续次数。在这情况下, 主要状态从运转化到故障(故障信息见章节 3.8 关于, 这个方法使用的函数见章节 5.8)。

l #define NB\_CONSECUTIVE\_TESTS

参数(8 位)定义在成功启动之前, 速度检测方差比 VARIANCE\_THRESHOLD 阈值低的连续次数。在这情况下, 主要状态从启动转化到运行(故障信息见章节 3.8)。

## 4.6 永磁同步电机(PMSM)参数: MC\_PMSM\_motor\_param.h

MC\_PMSM\_motor\_param.h 需要添加电机参数。可以分为三个不同的部分:

- l 磁场定向控制(FOC)需要的参数
- l 无传感转子速度/位置检测需要的参数
- l 弱磁功能需要的参数
- l 优化驱动内置永磁同步电机 IPMSM (MTPA) 需要的参数
- l 电流前馈调节需要的参数

### 4.6.1 基本电机参数

采用 FOC 需要的最小电机参数:

l #define POLE\_PAIR\_NUM

定义磁极对的数目。

l #define NOMINAL\_CURRENT

数字量, 根据下式公式定义电机额定电流(0-最大值):

$$I(\text{digit}) = \frac{I(\text{Amps}) \cdot R_{\text{shunt}} \cdot A_v \cdot 65536}{3.3}$$

当然, 这些数据与逆变器电流额定值相匹配。

l #define MOTOR\_MAX\_SPEED\_RPM

定义根据实际用电机的最大速度(转数/分)。参数的设置依据负载大小和电机机械结构以及具体应用。

如果机械负荷比额定值小, 通过弱磁功能可获得比额定值更高的速度(参考章节 2.3, 2.4 和 4.1)。

l #define ID\_DEMAG

以数字量（负 16 位值）定义电机磁体未被磁化时最大参考电流  $i_d^*$ 。参数最大等于 `-NOMINAL_CURRENT`（默认）

注意如果未启用 IPMSM MTPA 和弱磁功能，不需要具体化 `#define ID_DEMAG`（让它保持默认值不变）。

#### 4.6.2 无传感磁场定向控制（FOC）电机参数

状态观测器算法用这些参数检测转子速度和位置（参考章节 4.5）：

- `#define RS`

定义电机绕组（相）电阻值，单位欧姆。

- `#define LS`

定义电机绕组（相）电感值，单位亨利。

注意当电机时在 IPMSM 中，则  $L_d \neq L_q$ ，同步电感  $L_s$  可以设置为等于  $L_q$ （参考 A.9 中 [5]）

- `#define MOTOR_VOLTAGE_CONSTANT`

定义电机恒定电压  $K_e$ （相相 V/krpm RMS），单位伏特。

#### 4.6.3 弱磁运行的额外参数

根据具体应用，如果驱动电机运行超过额定速度就要完成这部分（额定速度是电机能达到最大转矩情况下的最高速度）。

任何情况下电机速度都不能超过机械限制。然而，当电机运行超过额定速度，就会强制再生制动势或者电阻制动（参考章节 3.14）。

弱磁控制策略见章节 2.3。

- I `#define FW_VOLTAGE_REF`

定义在弱磁控制中定子电压保持恒定的幅值（最大有效电压的 10%）。参考等级。参数设定越高（接近 1000），直流总线电压效率越大。

- I `#define FW_KP_GAIN`

定义（正数）弱磁模块中运行的 PI 调节器的比例增益（参考图 13：弱磁运行方案）。

- I `#define FW_KI_GAIN`

定义（正数）弱磁模块中运行的 PI 调节器的积分增益（参考图 13：弱磁运行方案）。

#### 4.6.4 IPMSM 驱动优化 (MTPA) 的额外参数

编辑与 MTPA 相关的参数，用下面信息填写附属电子表格：

- I 电机额定电流，0 — 峰值，单位 Amperes，单元 B2。这些数据必需与逆变器额定电流相匹配。
- I 电机绕阻  $L_d$  感应系数，单位 H，单元 B3
- I 电机绕阻  $L_q$  感应系数，单位 H，单元 B4
- I 电机恒定电压  $K_e$ ，单位伏特 RMS/krpm，相相电压，单元 B5
- I 电机绕阻电阻，单位 Ohms，单元 B6
- I 磁极对数，单元 B7
- I 分流电阻，单位 Ohms，单元 B9
- I 运算放大器增益 (读取分流电阻)，单元 B10

作为数据处理结果，可以从电子表格中获取以下信息：

- I `#define IQMAX`  
用户可以从单元 B19 中复制
- I `#define SEGDIV`  
用户可以从单元 B20 中复制
- I `#define ANGC`  
用户可以从单元 B21 中复制
- I `#define OFST`  
用户可以从单元 B22 中复制

#### 4.6.5 用于前馈、高性能电流控制的额外参数

为了编辑与前馈相关参数，用下面信息填写附属电子表格：

- I 电机绕阻  $L_d$  感应系数，单位 H，单元 B3
- I 电机绕阻  $L_q$  感应系数，单位 H，单元 B4  
注意当电机应用在 SM-PMSM，则  $L_d$  等于  $L_q$ 。
- I 电机恒定电压  $K_e$ ，单位伏特 RMS/krpm，相相电压，单元 B5
- I 电阻，单位 Ohms，单元 B9
- I 运算放大器增益 (读取分离电流)，单元 B10
- I SAMPLING\_FREQ (是根据下式公式，从 REP\_RATE 和 PWM\_FREQ 开始的预编译计算的 FOC 采样频率：

$$\text{磁通、转矩PID采样频率} = \frac{2 \text{ PWM\_FREQ}}{\text{REP\_RATE} + 1}, \text{ kHz, 单元 B11}$$

- I 直流总线电压到 **ADC** 转换因子；定义的是 **ADC** 输入电压和对应直流总线电压之比，单元 **B12**

通过数据处理，可以从电子表格中获取以下信息：

- I `#define CONSTANT1_Q`  
用户可以从单元 **B25** 中复制
- I `#define CONSTANT1_D`  
用户可以从单元 **B26** 中复制
- I `#define CONSTANT2`  
用户可以从单元 **B27** 中复制

## 5 函数库

函数描述用下列格式：

梗概 列出类型声明。

描述 简明解释函数如何运作来描述函数功能。

输入 设置格式和单位

返回值 函数返回值，包括一个输入值超出范围或一个错误代码时的返回值。

提示 指出函数的限制或明确在运行程序前应考虑的具体要求

注意 指明防止硬件故障的重要信息

函数调用 列出调用函数。有效的防止同时使用资源发生冲突。

代号实例 表面恰当使用函数的方式，指出前提条件（中断使能等）。

如果有些不可用就不包括在内（如：没有参数或明确用法）。

### 5.1 三相电阻拓扑电流采集和空间矢量 PWM 产生： stm32f10x\_svpwm\_3 shunt 模块

stm32f10x\_svpwm\_3shunt 模块主要完成两项重要任务：

- I 空间矢量脉冲宽度调制 (SVPWM)
- I 三相电阻拓扑电流采集

在一定精度下为了通过三相电阻将流向三相负载电流进行重组，必须使 PWM 信号产生和 A/D 转换同时进行。这就是为什么这两项任务要包含于单个软件单元。

#### 5.1.1 可用函数列表

下面是 stm32f10x\_svpwm\_3shunt.h 头文件中可用的函数列表：

- I SVPWM\_3ShuntInit 在 62 页
- I SVPWM\_3ShuntCurrentReadingCalibration on page 在 63
- I SVPWM\_3ShuntGetPhaseCurrentValues 在 63 页
- I SVPWMEOCEvent 在 64 页
- I SVPWMUpdateEvent 在 64 页
- I SVPWM\_3ShuntCalcDutyCycles 在 64 页
- I SVPWM\_3ShuntAdvCurrentReading 在 65 页

SVPWM\_3ShuntInit

梗概 void SVPWM\_3ShuntInit(void);

**描述** 函数用于配置微处理器外设来读取三相电阻拓扑电流和产生 PWM。  
函数初始化 NVIC、ADC、GPIO、TIM1 外设。  
特别，ADC 和 TIM1 外设配置成每 PWM 转换周期同时运行两路 A/D 转换。  
详情见 5.1.2。

**输入** 无。

**返回值** 无。

**提示** 必须在主函数中调用。

**函数调用** 标准库：

```
RCC_ADCCLKConfig, RCC_AHBPeriphClockCmd,  
RCC_APB2PeriphClockCmd, GPIO_StructInit, GPIO_Init,  
GPIO_PinLockConfig, GPIO_PinRemapConfig, TIM1_DeInit,  
TIM1_TimeBaseStructInit, TIM1_TimeBaseInit, TIM1_OCStructInit,  
TIM1_OC1Init, TIM1_OC2Init, TIM1_OC3Init, TIM1_OC4Init,  
TIM1_OC1PreloadConfig, TIM1_OC2PreloadConfig,  
TIM1_OC3PreloadConfig, TIM1_OC4PreloadConfig,  
TIM1_BDTRConfig, TIM1_SelectOutputTrigger,  
TIM1_ClearITPendingBit, TIM1_ITConfig,  
TIM1_Cmd, TIM1_GenerateEvent, TIM1_ClearFlag, ADC_DeInit,  
ADC_Cmd, ADC_StructInit, ADC_Init, ADC_StartCalibration,  
ADC_GetCalibrationStatus, ADC_RegularChannelConfig,  
ADC_InjectedSequencerLengthConfig,  
ADC_InjectedChannelConfig,  
NVIC_PriorityGroupConfig, NVIC_StructInit, NVIC_Init.
```

电机控制库：

```
SVPWM_3ShuntCurrentReadingCalibration
```

```
SVPWM_3ShuntCurrentReadingCalibration
```

**梗概** void SVPWM\_3ShuntCurrentReadingCalibration(void);

**描述** 函数功能用于存储对应零电流的三路模拟电压，用于补偿运放产生的零漂。

**输入** 无

**返回值** 无

**提示** 函数读取用于电流采集的 ADC 通道模拟电压值。因此必须在 PWM 输出前调用，以便通过逆变器电流为零。这些值存储在 Phase\_x\_Offset 变量中。

函数调用 标准库:

ADC\_ITConfig, ADC\_ExternalTriglInjectedConvConfig,  
ADC\_ExternalTriglInjectedConvCmd,  
ADC\_InjectedSequencerLengthConfig,  
ADC\_InjectedChannelConfig, ADC\_ClearFlag,  
ADC\_SoftwareStartInjectedConvCmd, ADC\_GetFlagStatus,  
ADC\_GetInjectedConversionValue,  
ADC\_SoftwareStartInjectedConvCmd

电机控制库:

SVPWM\_InjectedConvConfig

SVPWM\_3ShuntGetPhaseCurrentValues

梗概 Curr\_ComponentsSVPWM\_3ShuntGetPhaseCurrentValues(void);

描述 函数从 A/D 转换外设获得的电流值开始计算 q1.15 格式中 A 相和 B 相电流值。

输入 无

返回值 Curr\_Components 类型变量。

提示 为了取得 q1.15 格式的电流值，当读取相电流 A/D 转换值时，必须减去与漂移对应的数字量。因此，函数必须在 SVPWM\_3ShuntCurrentReadingCalibration 之后调用。

函数调用 无

SVPWME0CEvent

梗概 void SVPWME0CEvent();

描述 在 ISR 转换完成之内执行程序。计算总线电压和温度传感器采集和关闭 ADC 外部触发使能。

输入 无

返回值 总是真值

提示 无

函数调用 ADC\_GetInjectedConversionValue.

SVPWMUpdateEvent

梗概 void SVPWMUpdateEvent(void);

描述 更新 ISR 之内执行程序。重置 ADC 外部使能。

输入 无

返回值 无

提示 无

函数调用 `ADC_ClearFlag`.

`SVPWM_3ShuntCalcDutyCycles`  $V_\alpha$  和  $V_\beta$

梗概 `void SVPWM_3ShuntCalcDutyCycles (Volt_Components Stat_Volt_Input);`

描述 在 FOC 算法运算之后，计算定子新电压  $V_\alpha$  和  $V_\beta$ 。函数功能是从这些电压值开始的逆变器支路准确计算三个周期。并且，一旦下一次 PWM 周期三个周期已知，函数设置通道 4 输出与用于设置下一次电流读取的采样点的寄存器对比。特殊情况下，依据周期值，计算采样点（参考章节 5.1.3）。SVPWM 理论方法详见 5.1.2 节。

输入  $V_\alpha$  和  $V_\beta$

返回值 无

提示 无

函数调用 无

`SVPWM_3ShuntAdvCurrentReading`

梗概 `void SVPWM_3ShuntAdvCurrentReading(FunctionalState cmd);`

描述 用于超前电流读取的启动和关闭。如果禁止电流超前读取则电流读取与事件更新同时执行。

输入 `cmd` (ENABLE or DISABLE)

返回值 无

提示 无

函数调用 `TIM1_ClearFlag`, `TIM1_ITConfig`

## 5.1.2 空间矢量脉冲宽度调制（SVPWM）运行

图 36 显示定子电压  $V_\alpha$  和  $V_\beta$ ，图 37 阐明每同步六相空间矢量区域对应的 PWM。

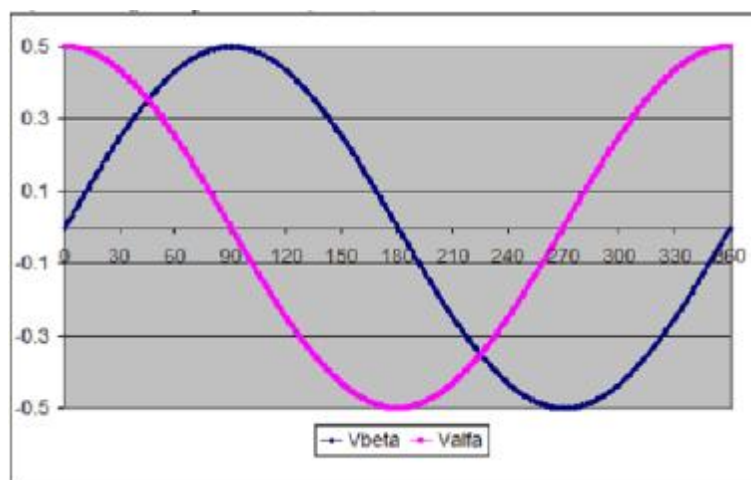


图 36  $V_\alpha$  和  $V_\beta$  定子电压



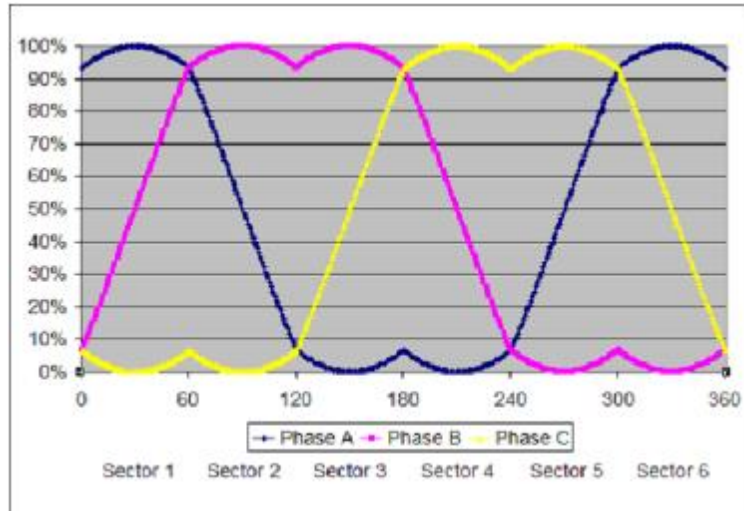


图 37 SVPWM 相电压波形

定义如下：  $U_a = \sqrt{3} \times T \times V_a$ ,  $U_b = -T \times V_b$  和  $X = U_b$ ,  $Y = \frac{U_a + U_b}{2}$ ,  $Z = \frac{U_b - U_a}{2}$

资料说明表 1 分别状态显示空间矢量区域。

表 1 区别域

	Y < 0			Y ≥ 0		
	Z < 0	Z ≥ 0		Z < 0		Z ≥ 0
		X ≤ 0	X > 0	X ≤ 0	X > 0	
Sector	V	IV	III	VI	I	II

通过下式关系分别计算 PWM 应用于 A、B、C 三相的正相脉冲宽度持续的时间：

$$\text{Sector I, IV: } t_A = \frac{T+X-Z}{2}, t_B = t_A + Z, t_C = t_B - X$$

$$\text{Sector II, V: } t_A = \frac{T+Y-Z}{2}, t_B = t_A + Z, t_C = t_A - Y$$

$$\text{Sector III, VI: } t_A = \frac{T-X+Y}{2}, t_B = t_C + X, t_C = t_A - Y, T \text{ 是 PWM 的周期。}$$

现在考虑 PWM 波形是中心对称，因此相电压必须集中到周期的 50%，根据 PWM 输出带负载的值，各自的参考电阻如下：

$$\text{Sector I, IV: TimePhA} = \frac{T}{4} + \frac{T/2 + X - Z}{2}, \text{TimePhB} = \text{TimePhA} + Z, \text{TimePhC} = \text{TimePhB} - X$$

$$\text{Sector II, V: TimePhA} = \frac{T}{4} + \frac{T/2 + Y - Z}{2}, \text{TimePhB} = \text{TimePhA} + Z, \text{TimePhC} = \text{TimePhA} - Y$$

$$\text{Sector III, VI: TimePhA} = \frac{T}{4} + \frac{T/2 + Y - X}{2}, \text{TimePhB} = \text{TimePhC} + X, \text{TimePhC} = \text{TimePhA} - Y$$

### 5.1.3 三相分流拓扑电流采集和通用 A/D 转换

流过三相系统的三相电流  $I_1$ 、 $I_2$  和  $I_3$  有下式数学关系：

$$I_1 + I_2 + I_3 = 0$$

因此，为了重现通过一般的三相负载电流，采集三相电流中的两项就可以通过上式关系推算第三项电流值。

STM32F103xx A/D 的灵活性可以同时采样两路 A/D 转换电流，从而计算出流过电机电流。使用外围设备的触发，PWM 输出同时 ADC 也可以用于电流采样。因此，电流转换可以在任何给定 PWM 周期内完成。为此，控制算法使用 TIM1 第四个 PWM 同时就启动转换。

鉴于有规律的转换转换功能为用户保留，上述描述的转换用于电流的读取。一旦电流读取 A/D 转换完成，双通道 A/D 同时转换总线电压和温度。

如图 38 所示 TIM1 PWM 输出和 ADC 同步进行。配置 A/D 转换外设以便于再 TIM1\_CH4 上升沿触发。

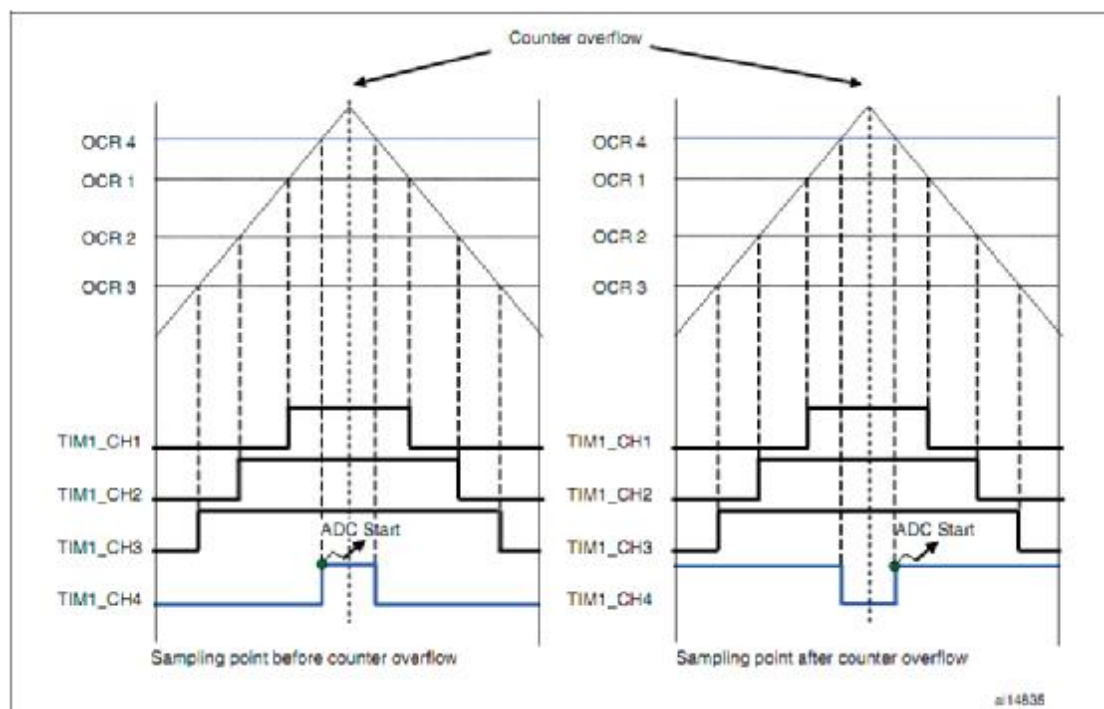


图 38 PWM 和 ADC 转换同步

用这种方式，假定必须在计数器溢出前设定采集点，即，当 TIM1 计数值与 OCR4 寄存器上升计数值相匹配，启动电流采集 A/D 转换。如果必须在计数器溢出后设定采集点，通过修改 TIM1\_CCER 寄存器 CC4P 位使 PWM4 输出反相。这样，当 TIM1 计数方式计数值与 OCR4 寄存器向下计数值相匹配，启动电流采集 A/D 转换。

前两次同时转换完成后就会启动下两次同时转换，一是总线电压另一个是测

量的温度。在第二次转换结束时，三相负载电流已经完成更新，在 A/D 转换中断服务末尾(JE0C ISR) 执行 FOC 算法。

在 FOC 算法执行后，下载到 OCR4 寄存器的值为下一次 PWM 周期设置采集点，并且设置 A/D 转换采集正确通道。

通用转换留给用户待用，必须手动配置（见标准固件库用户手册 UM0427）。

#### 5.1.4 三相分流电阻拓扑结构中调节延时参数和定子电流采集

图 39 所示为具有相关分流电阻的三相逆变器一个桥臂：

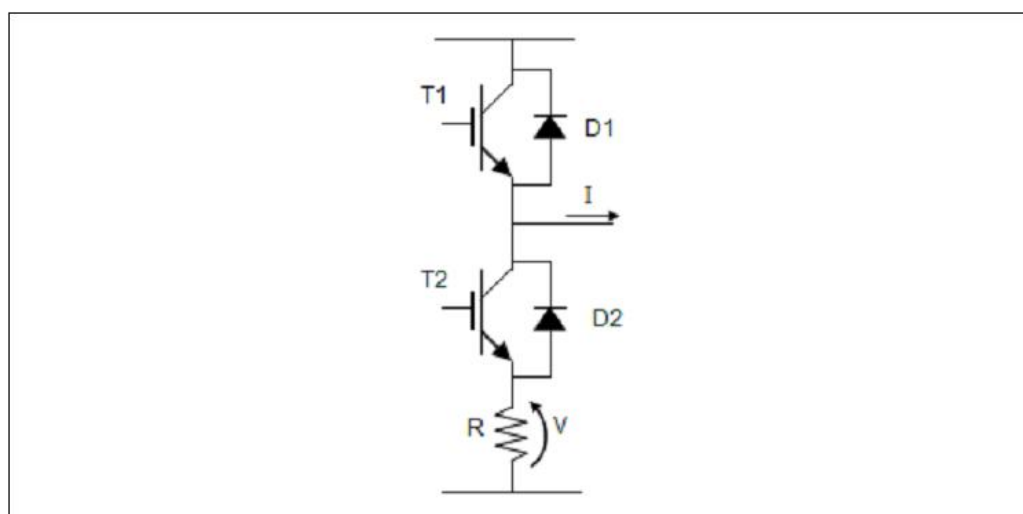


图 39 逆变器桥臂和分流电阻位置

为了间接测量 I 相电流，需要读取电流流过电阻 R 条件下电压值 V。

可以论证，无论 I 电流方向如何，当晶体管 T2 导通并且 T1 关断时电流始终流过电阻 R。这表明为了正确重现流过某一相逆变桥电流，必需正确同时触发转换器和产生 PWM 信号。这也意味当下面晶体管未开通或非常短时不能读取电流。

幸运地是，如章节 5.1.3 讨论的一样，重现流过三相负载电流，同时采集三相电流中的两相就可以，第三相电流可以通过章节 5.1.3 关系式计算。这样，依靠空间矢量区域，A/D 转换电压 V 仅通过下侧开关周期最高的那两相就可以完成。特殊地，观察图 37，去除区域 1 和 6，不使用 A 相分流电阻；去除区域 2 和 3，不使用 B 相分流电阻；去除区域 4 和 5，不使用 C 相分流电阻。

此外，为了适当同时读取两项电流 A/D 转换值，必需根据 PWM 频率和实际应用周期区分不同情况。

注释：下述说明指的是空间矢量区域 4。同样的方法应用于其他区域。

Case 1: A 相下波周期比  $DT + T_N$  大

其中：

I DT 是失效时间。

- I  $T_N$  是由于其它相开关导致的在某一相分流电阻电压产生的噪声持续时间。
- I  $T_s$  是 STM32F103xx A/D 转换采集时间（下列阐述假定  $T_s < DT + T_N$ ）。更多信息参考 STM32F103xx 参考手册。

当产生的 SVPWM 调制指数比较小时(<60%)上述就会发生（见图 40）。调制指数应用于相电压占最大相电压的百分比（周期从 0%到 100%）。

图 41 展现的时应用于 A 和 B 相下波的 PWM 信号，还有 B 和 C 相 STM32F103xx A/D 转换器引脚测量的模拟电压（时间基准小于 PWM 周期）。

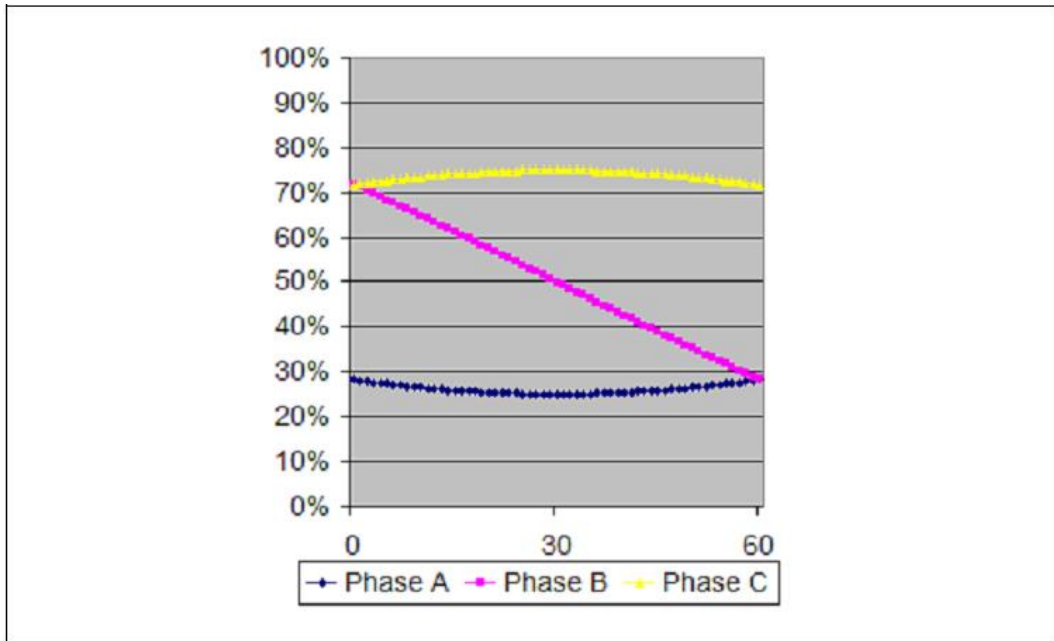


图 40 下波门信号（低调制指数）

注意图 41 中电流反馈位常数，因为图中假设 B 和 C 相在可视化时间窗口超时。

此外，可以观察到在这事件下当计数器溢出时，两个定子电流采集转换就会运行，如图 41 所示。

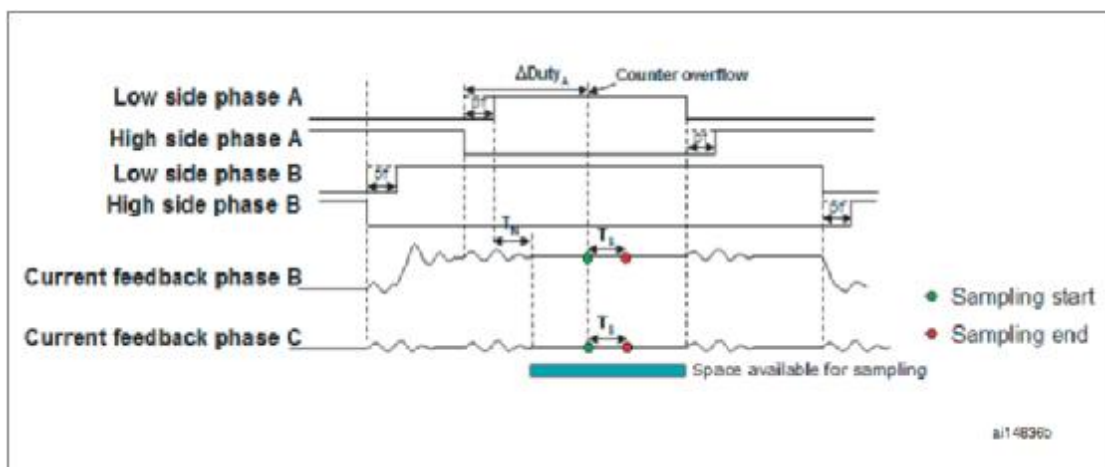


图 41 A 低波周期> DT+TN

Case 2:  $(DT+T_N+T_S)/2 < \Delta \text{Duty}_A < DT+T_N$  和  $\Delta \text{Duty}_{AB} < D_T+T_R+T_S$

随着调制指数的增加， $\Delta \text{Duty}_A$  的值小于  $DT+T_N$ 。将不可能在计数器溢出时采集。

这在这种情况下，仍在 A 相两低波之间转换采集两路电流，但只有当计数器溢出之后才行。

考虑到为了避免 A 相开关转换对 B 相电流的反馈产生的噪声，要求等待干扰结束 ( $T_N$ )。见图 42。

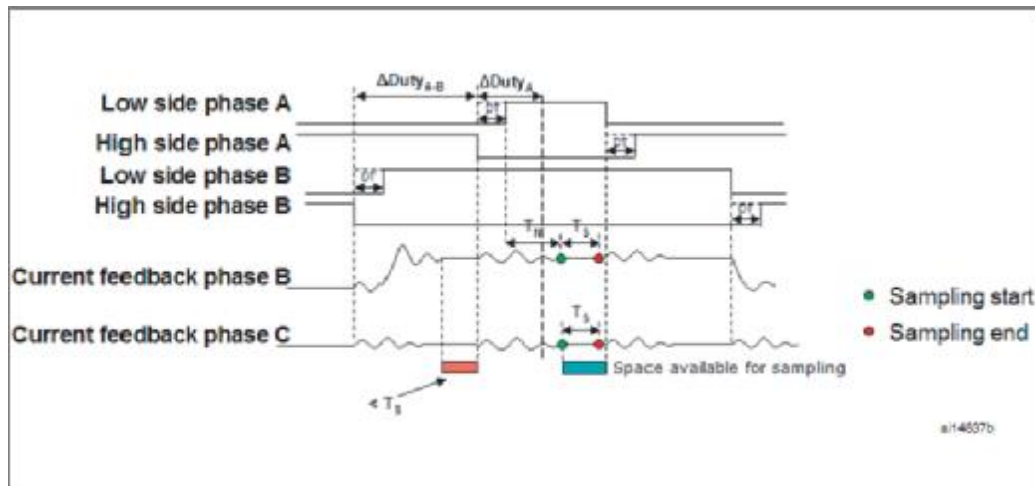


图 42  $(DT+T_N+T_S)/2 < \Delta \text{Duty}_A < DT+T_N$  和  $\Delta \text{Duty}_{AB} < D_T+T_R+T_S$

Case 3:  $\Delta \text{Duty}_A < (DT+T_N+T_S)/2$  和  $\Delta \text{Duty}_{A-B} > D_T+T_R+T_S$

在这种情况下，不可能在 A 相低波段采集电流。无论如何，可以在 B 相低波关断和 A 相高波开始采集电流。因此必须选择在 A 相高波关断之前采集电流  $T_S$  us (如图 43)。

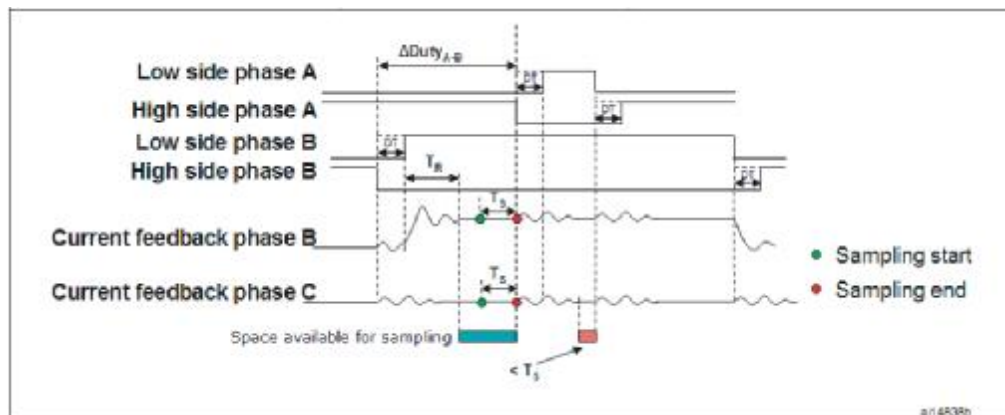


图 43  $\Delta \text{Duty}_A < (DT+T_N+T_S)/2$  和  $\Delta \text{Duty}_{A-B} > D_T+T_R+T_S$

Case 4:  $\Delta \text{Duty}_A < (DT+T_N+T_S)/2$  和  $\Delta \text{Duty}_{A-B} < D_T+T_R+T_S$

在这种情况下，A 相周期太短，无法在两个低波间采集电流。

此外如果 B 相和 A 相周期之间的不同，不足以在 B 相低波开通和 A 相高波截止之间运行 A/D 转换，也不可能采样电流 (如图 44)。

为了避免这种情况，必需减小最大的调制指数或降低 PWM 频率。

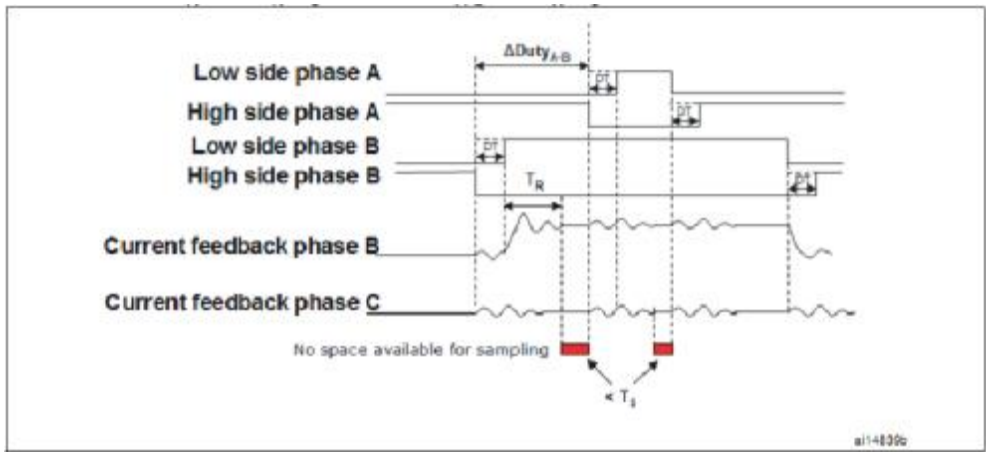


图 44  $\Delta Duty_A < (DT + T_N + T_S)/2$  和  $\Delta Duty_{A-B} < DT + T_R + T_S$

固件中默认设置下述参数。这些指 MB459 开发板：

- I  $DT = 0.8 \mu s$
- I  $T_N = 2.55 \mu s$
- I  $T_S = 0.7 \mu s$
- I  $T_R = 2.55 \mu s$

表 2 列出作为 PWM 频率函数的最大运行周期。

表 2 的值用 MB459 开发板测量的。这评估平台用于支持多种电机驱动拓扑结构（PMSM 和交流感应）和电流读取策略（单相和三相分流电阻）。因此，表 2 中数据应该理解为一个起始点而非最优数据。

表 2 PWM 频率与最大周期之比与三相分流拓扑结构之间的关系

PWM frequency	Max duty cycle	Max modulation Index
Up to 11.4 kHz	100%	100%
12.2 kHz	99%	98%
12.9 kHz	98.5%	97%
13.7 kHz	98%	96%
14.4 kHz	98%	96%
15.2 kHz	97%	94%
16 kHz	96.5%	93%
16.7 kHz	96.5%	93%
17.5 kHz	95.5%	91%

依据硬件的设计可以调整干扰参数，通过编辑 MC\_pwm\_3shunt\_prm.h 头文件中下述定义：

```
#define SAMPLING_TIME_NS 700 //0.7usec
#define TNOISE_NS 2550 //2.55usec
#define TRISE_NS 2550 //2.55usec
```



改变干扰参数、采集时间和失效时间可以改变表 2 提供的值。

## 5.2 单相电阻拓扑电流读取和空间矢量 PWM 产生： stm32f10x\_svpwm\_1shunt module

stm32f10x\_svpwm\_1shunt module 的两项主要任务：

- I 空间矢量脉冲宽度调制 (SVPWM)
- I 单相电阻拓扑电流读取

为了按一定精确用一个分流电阻重现流过三相负载电流，就需要同时产生 PWM 信号和启动同步 A/D 转换。这也是为什么这两项任务包含在同一个软件模块中。

### 5.2.1 可用函数列表

下面在 stm32f10x\_svpwm\_1shunt.h 头文件可用函数的列表：

- I SVPWM\_1ShuntInit 在 74 页
- I SVPWM\_1ShuntCurrentReadingCalibration 在 75 页
- I SVPWM\_1ShuntGetPhaseCurrentValues 在 75 页
- I SVPWM\_1ShuntCalcDutyCycles 在 76 页
- I SVPWM\_1ShuntAdvCurrentReading 在 77 页
- I SVPWMEOCEvent 在 77 页
- I SVPWMUpdateEvent 在 77 页

#### SVPWM\_1ShuntInit

梗概 void SVPWM\_1ShuntInit(void);

描述 函数作用是设置微控制器外设，以便于读取单分流电阻拓扑电流，产生居中对齐 PWM 信号。函数初始化 NVIC、ADC、GPIO、TIM1 和 DMA 外设。尤其，ADC 和 TIM1 外设需配置成每 PWM 周期运行两次 A/D 转换。更多的信息见 5.2.3 节。

输入 无

返回值 无

提示 必需在主函数中调用

函数调用 标准库：

RCC\_ADCClockConfig, RCC\_AHBPeriphClockCmd,  
RCC\_APB2PeriphClockCmd, GPIO\_StructInit, GPIO\_Init,  
GPIO\_PinLockConfig, GPIO\_PinRemapConfig, TIM1\_DeInit,  
TIM1\_TimeBaseStructInit, TIM1\_TimeBaseInit, TIM1\_OCStructInit,

TIM1\_OC1Init, TIM1\_OC2Init, TIM1\_OC3Init, TIM1\_OC4Init,  
TIM1\_OC1PreloadConfig, TIM1\_OC2PreloadConfig,  
TIM1\_OC3PreloadConfig, TIM1\_OC4PreloadConfig,  
TIM1\_BDTRConfig, TIM1\_SelectOutputTrigger,  
TIM1\_ClearITPendingBit, TIM1\_ITConfig,  
TIM1\_Cmd, TIM1\_GenerateEvent, TIM1\_ClearFlag, TIM1\_DMACmd,  
TIM1\_DMAConfig, ADC\_DeInit, ADC\_Cmd, ADC\_StructInit,  
ADC\_Init, ADC\_StartCalibration, ADC\_GetCalibrationStatus,  
ADC-RegularChannelConfig, ADC\_InjectedSequencerLengthConfig,  
ADC\_InjectedChannelConfig, NVIC\_PriorityGroupConfig,  
NVIC\_StructInit, NVIC\_Init. DMA\_DeInit, DMA\_Init, DMA\_Cmd  
电机控制库: SVPWM\_1ShuntCurrentReadingCalibration

SVPWM\_1ShuntCurrentReadingCalibration

梗概 void SVPWM\_1ShuntCurrentReadingCalibration(void);

描述 函数作用是用于存储对应零电流的模拟电压, 补偿运放电路产生的漂移。

输入 无

返回值 无

提示 函数读取 ADC 通道的模拟电压, 用于电流读取。因此, 必需在 PWM 输出启动前调用, 以便保证流过逆变器的电流为零。这些值存储在 hPhaseOffset 变量中。

函数调用 标准库: ADC\_ITConfig,  
ADC\_ExternalTrigInjectedConvConfig,  
ADC\_ExternalTrigInjectedConvCmd,  
ADC\_InjectedSequencerLengthConfig,  
ADC\_InjectedChannelConfig,  
ADC\_ClearFlag, ADC\_SoftwareStartInjectedConvCmd,  
ADC\_GetFlagStatus, ADC\_GetInjectedConversionValue,  
ADC\_SoftwareStartInjectedConvCmd  
电机控制库:  
SVPWM\_InjectedConvConfig

SVPWM\_1ShuntGetPhaseCurrentValues

梗概 Curr\_Components SVPWM\_1ShuntGetPhaseCurrentValues(void);

描述 函数从 A/D 转换外设获得值开始以 q1.15 格式计算 A 相和 B 相电流值。  
如图 45 所示。



输入 无

返回值 Curr\_Components 变量类型。

提示 为了让电流值为 q1.15 格式，在读取 A/D 转换值电流值时，必须减去对应的偏移量。因此，函数必需在 SVPWM\_1ShuntCurrentReadingCalibration 之后调用。

函数调用 无

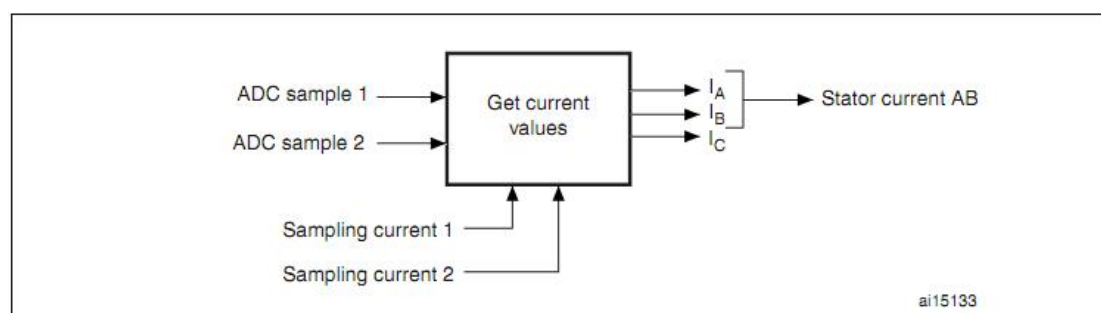


图 45 GetPhaseCurrentValues 结构框图

### SVPWM\_1ShuntCalcdutyCycles

梗概 void SVPWM\_1ShuntCalcdutyCycles (Volt\_Components Stat\_Volt\_Input);

描述 FOC 算法执行之后，计算定子电压新量  $V_\alpha$  和  $V_\beta$ 。函数作用就是从这些电压量开始准确计算逆变桥三个周期。此外，一旦下一个 PWM 周期三个工作周期已知，函数执行下述任务：

- 获取定子磁通位置（连续或边界层 1、2 或 3）
- 计算 PWM 通道，必需改变或重新计算周期寄存器值
- 计算采集点和相关采集相位
- 设置 PWM 模式通道 1、2、3、4 的预加负载。

如图 46，单相分流电流读取的理论方法参考章节 5.2.2

输入  $V_\alpha$  and  $V_\beta$ .

返回值 无

提示 无

函数调用 无

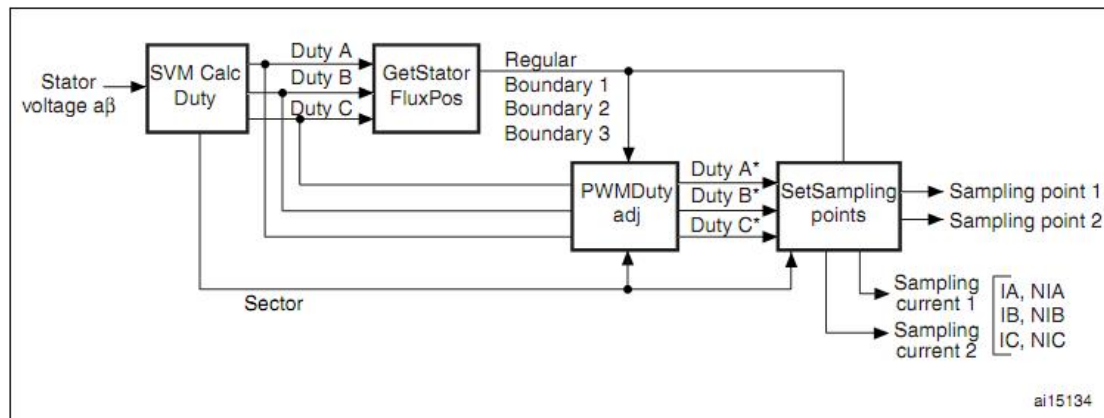


图 46 CalcDutyCycles 结构框图

### SVPWM\_1ShuntAdvCurrentReading

梗概 void SVPWM\_1ShuntAdvCurrentReading(Functional State cmd);

描述 用于启动或关闭电流的读取功能。如果关闭了电流读取，就和更新一样一直在采集总线电压和温度。

输入 cmd (ENABLE or DISABLE)

返回值 无

提示 无

函数调用 TIM1\_ClearFlag, TIM1\_ITConfig

### SVPWMEOCEvent

梗概 void SVPWMEOCEvent();

描述 ISR 转换程序末尾运行。对于单相分流电流读取，调用两次。第一次存储第一次采集的值。返回“假”。第二次返回“真”表明 FOC 周期执行。计算采集的总线电压和温度，关闭外部 ADC 触发。

输入 无

返回值 第一次返回“假”，第二次返回“真”

提示 无

函数调用 ADC\_GetInjectedConversionValue, ADC\_ITConfig.

### SVPWMUpdateEvent

梗概 void SVPWMUpdateEvent(void);

描述 ISR 更新时运行。设置四通道的 PWM 输出模式 (Toggle 或 PWM)，启动或关闭每个通道 DMA 事件，更新 DMA 缓冲和 DMA 长度，最后，重新启动外部 ADC 触发。

输入 无

返回值 无

提示 无

函数调用    `ADC_ClearFlag, ADC_ITConfig`

5.2.2 单相分流拓扑电流采集

图 47 说明单相分流硬件结构。

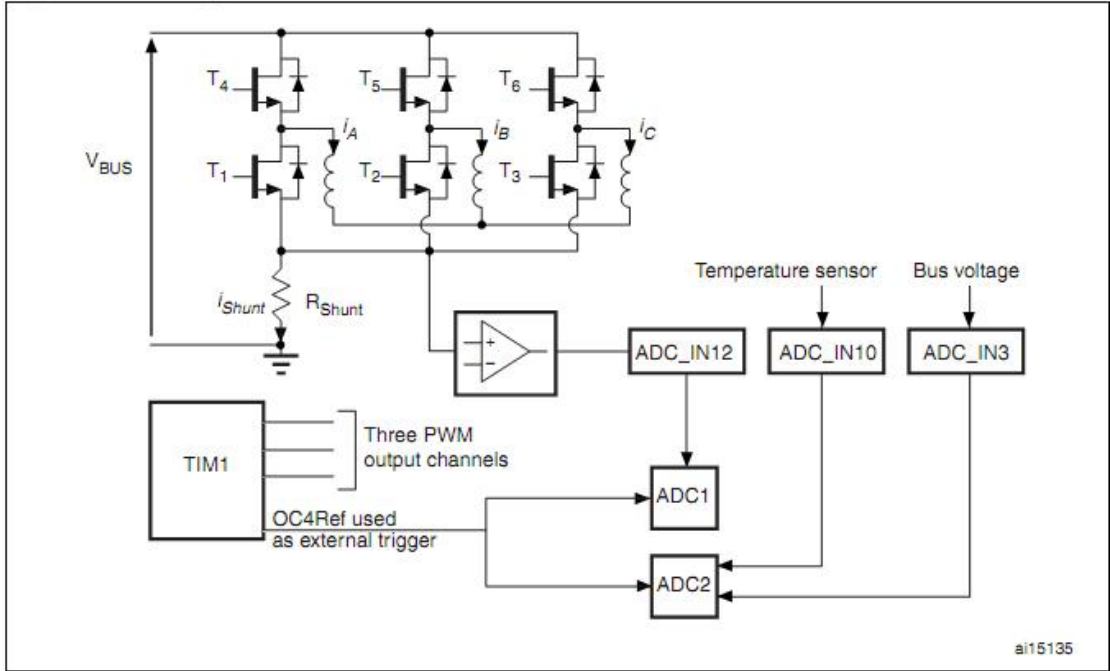


图 47 单相分流硬件结构

可以证明低波的每个配置，表 3 中都给出了流过分流电阻的电流。T<sub>4</sub>、T<sub>5</sub> 和 T<sub>6</sub> 分别为 T<sub>1</sub>、T<sub>2</sub> 和 T<sub>3</sub> 的互补值。

在表 3 中，值“0”表示开关打开而值“1”表示开关闭合。

表 3 通过分流电阻的电流

T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	<i>i<sub>Shunt</sub></i>
0	0	0	0
0	1	1	<i>i<sub>A</sub></i>
0	0	1	- <i>i<sub>C</sub></i>
1	0	1	<i>i<sub>B</sub></i>
1	0	0	- <i>i<sub>A</sub></i>
1	1	0	<i>i<sub>C</sub></i>
0	1	0	- <i>i<sub>B</sub></i>
1	1	1	0

使用中心居中形式，每个 PWM 周期细分成 7 个周期（如图 48）。在三个小周期（I, IV, VII）中，通过分流电阻的电流为零。其他周期时，通过电阻的电流在 PWM 中心对称。

如图 48 所示的状况，存在两类：

- I 周期 II 和 VI,  $i_{Shunt}$  等于  $-i_c$
- I 周期 III 和 V,  $i_{Shunt}$  等于  $i_A$

在这些情况下，可以通过采集值重现流过电机的三相电流：

- I 在周期 III 或 V 中,  $i_A$  是  $i_{Shunt}$  测量值
- I 在周期 II 或 VI 中,  $i_c$  是  $-i_{Shunt}$  测量值
- I  $i_B = -i_A - i_c$

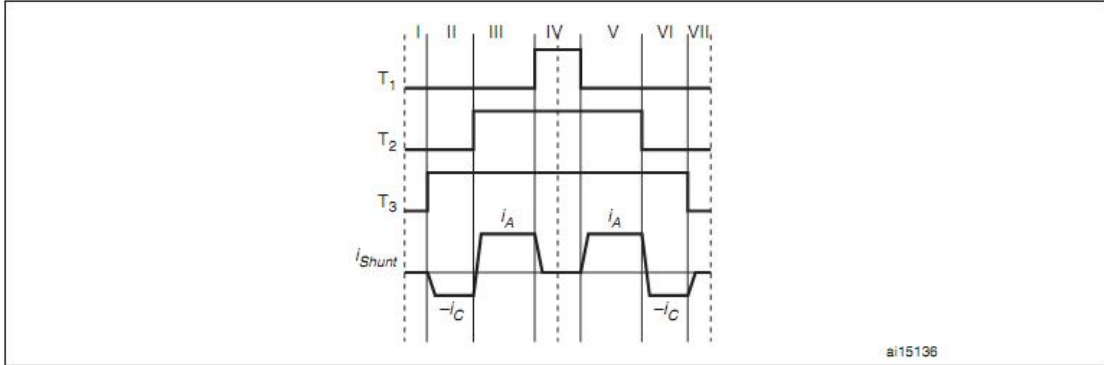


图 48 单相分流电流读取

但是如果定子电压要求矢量坐标位于两矢量区域之间的分界范围，三个占空比中两个会接近相同的值。这样七个周期段就减少为五个。

在这些条件下，只能采集单相电流，其他两相电流不能被重现。这意味着当外部强加电压要求矢量位于图 49 中图表显示的灰色矢量区域中时，在同一个 PWM 周期内不可能同时测量两相电流，。

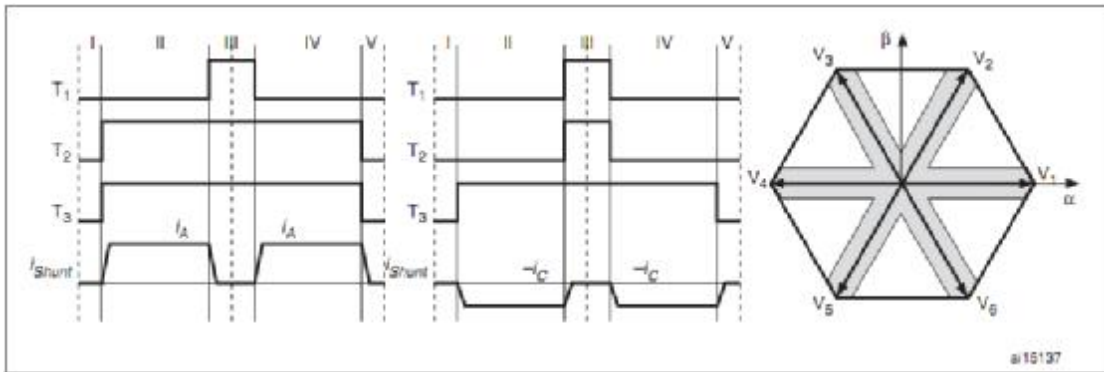


图 49 两矢量区域的边界

类似地，在调制指数较低时，三占空比都接近相同的值。这样，七个周期段减少为三个。这三个周期段中，流过分流电阻的电流为零。这意味着当外部强加电压矢量位于图 50 中图表显示的灰色矢量区域时，不可能测量任何电流中。

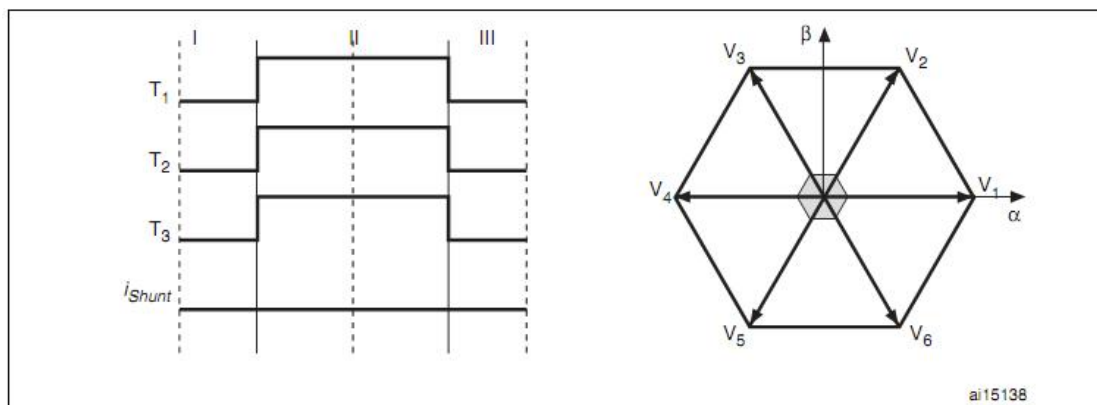


图 50 低调制指数

### 5.2.3 定义干扰参数和边界区域

$T_{Rise}$  是在电源驱动开关打开或关闭后 ADC 通道数据稳定所需时间。

ADC 采集持续的时间称为采集周期。

$T_{MIN}$  是执行采集要求最小的时间，即

$$T_{MIN} = T_{Rise} + \text{采集时间} + \text{迟滞时间}$$

$D_{MIN}$  是  $T_{MIN}$  占空比的表示。它与 PWM 频率有关，关系如下：

$$D_{MIN} = (T_{MIN} \times F_{PWM}) \times 100$$

通过编辑 MC\_pwm\_1shunt\_prm.h 头文件中下述定义，可以依据硬件调节干扰参数：

- I #define SAMPLING\_TIME\_NS 700 //0.7usec
- I #define TRISE\_NS 2550 //2.55usec

改变干扰参数，采集周期和迟滞时间将影响表 4 中的值。

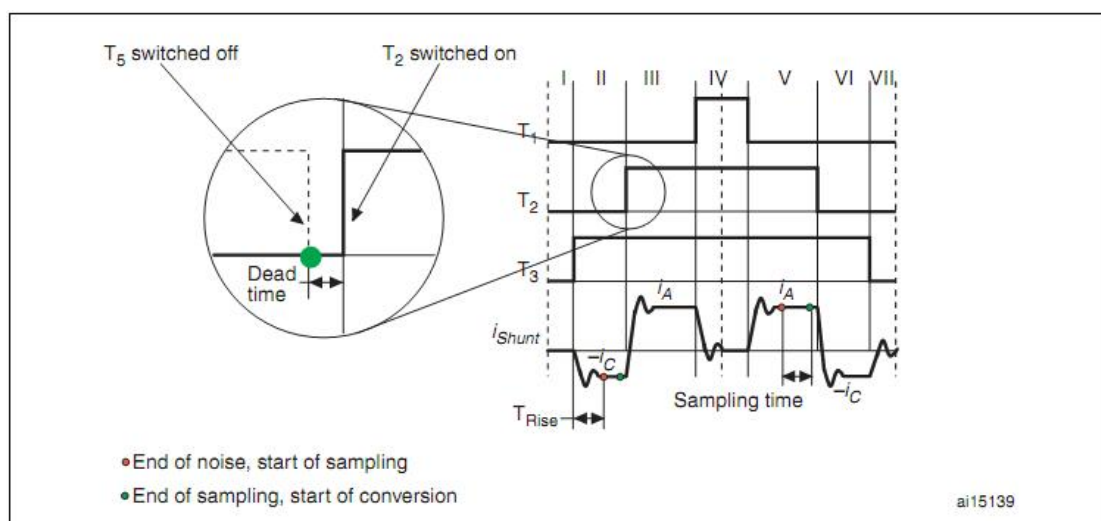


图 51 干扰参数的定义

当 PWM 形式三个占空比（经过空间矢量调制计算）差异大于  $D_{MIN}$  时，电压要

求矢量坐标所在的区域叫规则区域，如图 52 所示。

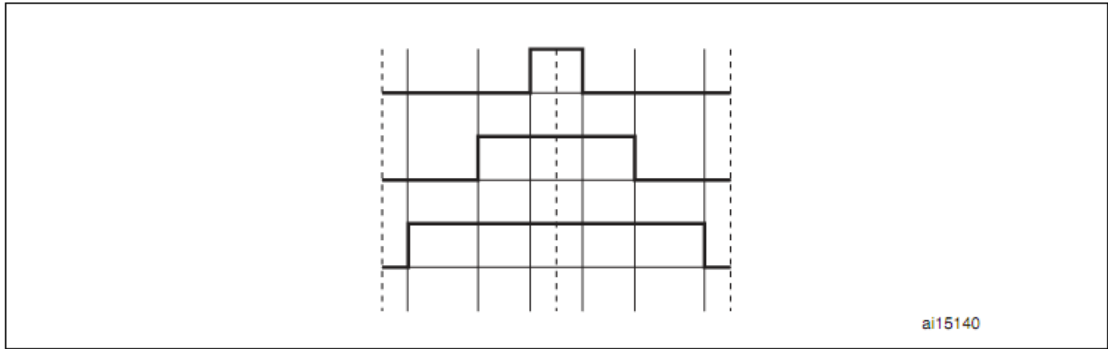


图 52 固有区域

当两个占空比与其它之间的差异小于  $D_{\min}$  时，电压需求矢量坐标区域叫边界 1，第三个占空比比其它大，之间的差异大于  $D_{\min}$ 。如图 53 所示。

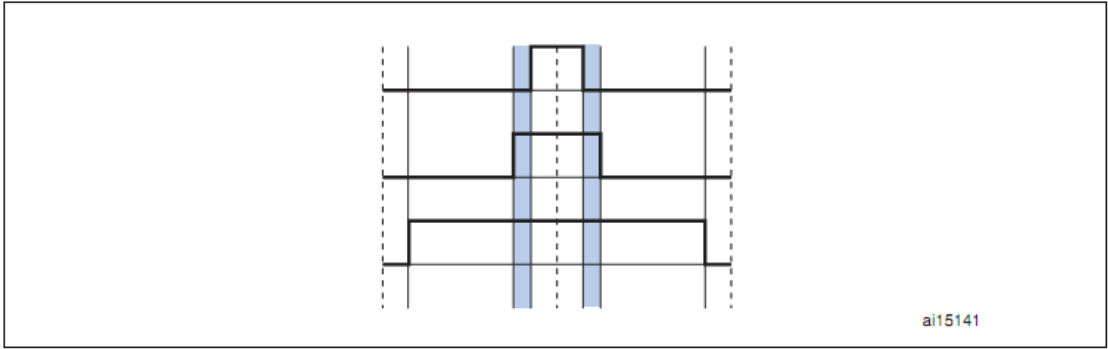


图 53 边界 1

当两个占空比与其它之间的差异小于  $D_{\min}$  时，电压需求矢量坐标区域叫边界 2，第三个占空比比其它小，之间的差异大于  $D_{\min}$ 。如图 54 所示。

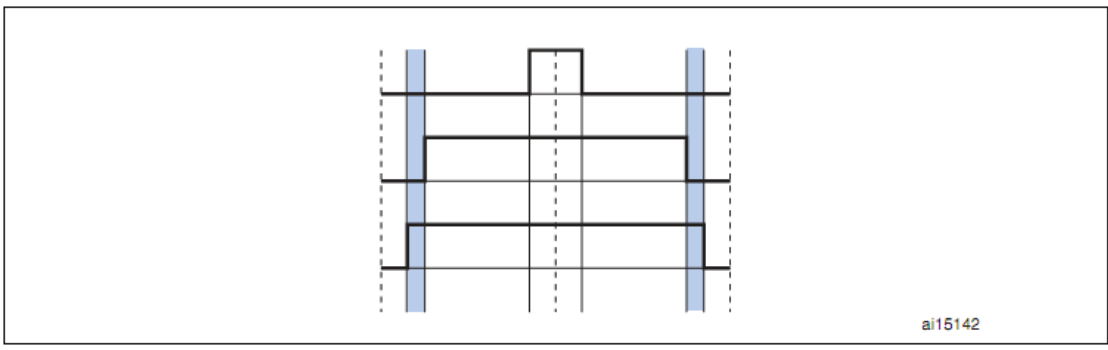


图 54 边界 2

当三个 PWM 信号互不相同，而且差异小于  $D_{\min}$  时，电压需求矢量坐标区域叫边界 3。如图 55 所示。

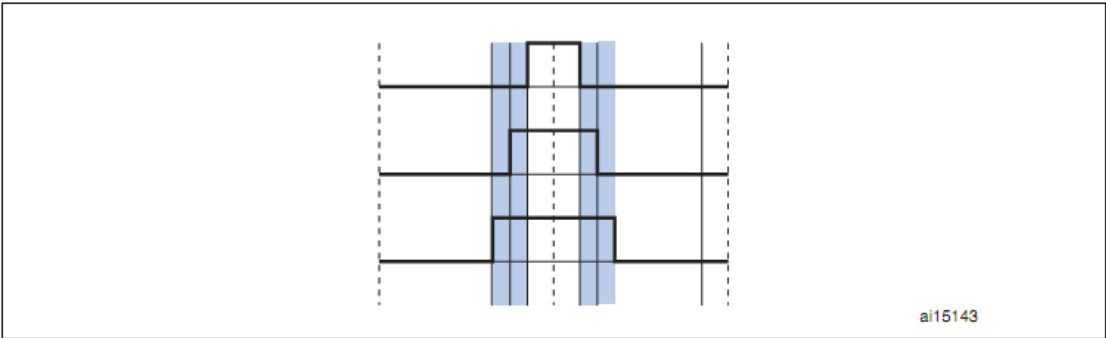


图 55 边界 3

如果电压要求矢量坐标位于边界 1 或边界 2 区域，必须在 PWM 信号相位中引入形变以便采集电机相电流。

固件在边界区域电流采集采用 ST 专利技术。联系最近的 ST 销售点或支持团队，获取更多技术信息。

注释：电机高速转动导致的高电感下，默认设置采用的电流补偿技术会导致电流采集不理想。这样，注释下述 MC\_pwm\_1shunt\_prm.h 头文件的定义，排除电流补偿技术：

```
#define CURRENT_COMPENSATION
```

### 5.2.4 性能

下述参数已经在固件中设置。这些参数与 MB459 模块相关：

- DT = 0.8  $\mu$ s
- T<sub>R</sub> = 1.5  $\mu$ s
- T<sub>S</sub> = 0.7  $\mu$ s

最大适应占空比作为 PWM 频率的一项功能在表 4 中列出。

表 4 单相分流拓扑 PWM 频率与最大占空比关系

PWM frequency	Max duty cycle	Max modulation index	Min REP_RATE
Up to 11.4 kHz	100%	100%	1
12.2 kHz	100%	100%	
12.9 kHz	100%	100%	
13.7 kHz	100%	100%	
14.4 kHz	100%	100%	
15.2 kHz	100%	100%	3
16 kHz	99.5%	99%	
16.7 kHz	99%	98%	
17.5 kHz	99%	98%	

## 5.3 隔离电流传感器读取和空间矢量的 PWM 产生： stm32f10x\_svpwm\_ics module

stm32f10x\_svpwm\_ics 模块执行的两项重要任务。

- I 空间矢量脉冲宽度调制 (SVPWM)。
- I 使用两个隔离电流传感器(ICS)的三相电流读取。

为了使两个 ICS 按要求精度重现流过三相负载电流，必须同时进行 A/D 转换和产生 PWM 信号。因此这两项任务包含在同一个软件模块中。

### 5.3.1 可用的函数列表和中断服务程序

下述是 stm32f10x\_svpwm\_ics.h 头文件中列出的可用函数列表：

- I SVPWM\_IcsInit 在 84 页
- I SVPWM\_IcsCurrentReadingCalibration 在 84 页
- I SVPWM\_IcsGetPhaseCurrentValues 在 85 页
- I SVPWM\_IcsCalcDutyCycles 在 85 页
- I SVPWMEOCEvent 在 86 页
- I SVPWMUpdateEvent 在 86 页

#### SVPWM\_IcsInit

梗概 void SVPWM\_IcsInit(void);

描述 函数功能是设置用于 ICS 电流读取的微控制器外设和产生 PWM。

函数初始化 NVIC、ADC、GPIO 和 TIM1 外设。尤其，每次 PWM 寄存器更新（调用 U 事件）ADC 和 TIM1 外设配置完成两对 A/D 转换。第一对转换读取电流值而第二对获取总线电压和温度传感器电压。ICS 配置中 A/D 转换触发的更多资料参考章节 5.3.2。

输入 无。

返回值 无。

提示 必须在主函数调用。

函数调用 标准库：

```
RCC_ADCCLKConfig, RCC_AHBPeriphClockCmd,  
RCC_APB2PeriphClockCmd, GPIO_StructInit, GPIO_Init,  
GPIO_PinLockConfig, GPIO_PinRemapConfig, TIM1_DeInit,  
TIM1_TimeBaseStructInit, TIM1_TimeBaseInit,  
TIM1_OCStructInit, TIM1_OC1Init, TIM1_OC2Init, TIM1_OC3Init,  
TIM1_BDTRConfig, TIM1_SelectOutputTrigger,
```



TIM1\_ClearITPendingBit, TIM1\_ITConfig,  
TIM1\_Cmd, ADC\_DeInit, ADC\_Cmd, ADC\_StructInit, ADC\_Init,  
ADC\_StartCalibration, ADC\_GetCalibrationStatus,  
ADC\_InjectedSequencerLengthConfig,  
ADC\_InjectedChannelConfig,  
ADC\_ExternalTrigInjectedConvCmd, NVIC\_PriorityGroupConfig,  
NVIC\_StructInit, NVIC\_Init.

电机库:

SVPWM\_IcsCurrentReadingCalibration

SVPWM\_IcsCurrentReadingCalibration

梗概 void SVPWM\_IcsCurrentReadingCalibration(void);

描述 函数的功能是存储对应零电流的两路模拟电压，补偿 ICS 和运放电路引起的漂移。

输入 无。

返回值 无。

提示 这个函数通过 MCL\_Init 调用，在每个电机启动时执行。通过 A/D 通道读取模拟电压值，并在 PWM 输出启动前用于读取电流，这样流过逆变器的电流为零。

函数调用 标准库:

ADC\_ITConfig, ADC\_ExternalTrigInjectedConvConfig,  
ADC\_ExternalTrigInjectedConvCmd, ADC\_InjectedChannelConfig,  
ADC\_ClearFlag, ADC\_SoftwareStartInjectedConvCmd,  
ADC\_GetFlagStatus, ADC\_GetInjectedConversionValue,  
SVPWM\_IcsInjectedConvConfig

电机库:

SVPWM\_IcsInjectedConvConfig

SVPWM\_IcsGetPhaseCurrentValues

梗概 Curr\_Components SVPWM\_IcsGetPhaseCurrentValues(void);

描述 函数用 A/D 转换获取的值以 q1.15 格式计算 A 相和 B 相电流值，

输入 无。

返回值 Curr\_Components 变量类型

提示 为了获取 q1.15 格式电流值，当读取相电流 A/D 转换值时，必须减去对应的漂移数值。这样，函数必须在 SVPWM\_IcsCurrentReadingCalibration 之后调用。

函数调用 无。

#### SVPWM\_IcsCalcDutyCycles

梗概 `void SVPWM_IcsCalcDutyCycles (Volt_Components Stat_Volt_Input);`

描述 在执行 FOC 算法之后，就可以计算定子新电压量  $V_\alpha$  和  $V_\beta$ 。函数的功能从这些电压量开始，准确计算三个逆变器桥臂三个占空比。

SVPWM 的理论和运行方法参考章节 5.1.2。

输入  $V_\alpha$  和  $V_\beta$

返回值 无。

提示 无。

函数调用 无。

#### SVPWMEOCEvent

梗概 `void SVPWMEOCEvent();`

描述 在 ISR 转换最后执行程序。用于计算总线电压和温度传感器采集。

输入 无。

返回值 总是真。

提示 无。

函数调用 `ADC_GetInjectedConversionValue`

#### SVPWMUpdateEvent

梗概 `void SVPWMUpdateEvent(void);`

描述 在更新事件 ISR 中执行程序。执行为空。

输入 无。

返回值 无。

提示 无。

函数调用 无。

### 5.3.2 隔离电流传感器拓扑和通用 A/D 转换的整合中电流采集

三相系统中电流  $I_1$ 、 $I_2$ 、 $I_3$ 有如下的数学关系：

$$I_1 + I_2 + I_3 = 0$$

因此，为了重现流过三相负载，只要采集三相电流中的两相就足够，而第三路可以通过上述的公式计算出。

STM32F103xx A/D 转换的灵活触发可以使为重现流过电机的定子电流两路 A/D 转换与 PWM 重载寄存器更新同时进行。调节重复计数器可以更新速度。这很重要，正如图 56 所示，在计数器溢出和下溢时平均电流值等于采集电流很精确。参阅 STM32F103xx 参考手册了解更多关于 A/D 转换触发和重复计数器。

注释：为用户保留常规转换，但是必须手动配置（参考固件标准库用户手册 UM0427）。

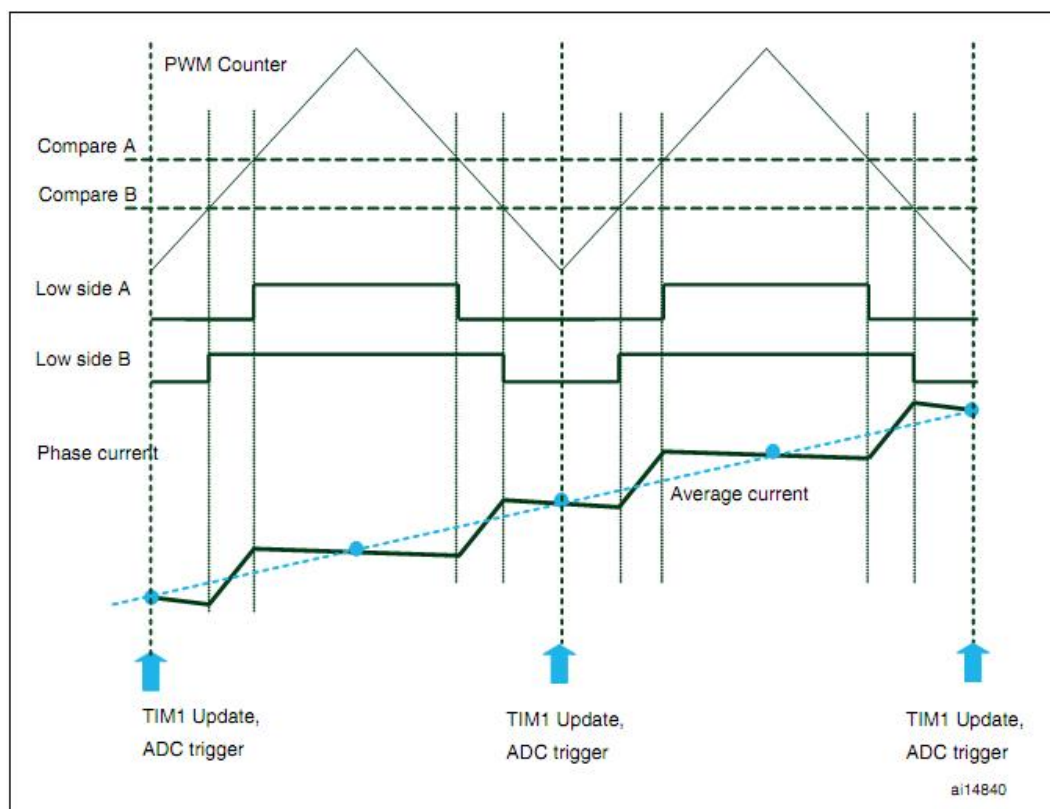


图 56 ICS 配置中定子电流采集(Rep\_Rate=1)

## 5.4 PMSM (SM-PMSM / IPMSM) 磁场定向控制：MC\_FOC\_Drive and MC\_FOC\_Methods 模块

MC\_FOC\_Drive 和 MC\_FOC\_Methods 单元，设计用于外置或内置永磁同步电机，提供电磁力矩 ( $T_e$ ) 控制，有时还会具有弱磁能力。另外，通过 PID 反馈控制还可以调节速度。

运行起来不要求调节所有可选择电流或速度传感配置（与 stm32f10x\_MCconf.h 文件的设置相一致）：

- l 隔离电流传感器 (ICS)
- l 三相分流电阻电流传感器
- l DC 环单相分流电阻器
- l 编码器位置和速度感应
- l 霍尔传感器位置和速度感应
- l 无传感位置和速度检测

MC\_FOC\_Drive 单元控制其他单元的几个函数，与微控制器外设寄存器没有直接

接口。

### 5.4.1 可用 C 函数列表

- | FOC\_Init 在 88 页
- | FOC\_Model 在 89 页
- | FOC\_CalcFluxTorqueRef 在 90 页
- | FOC\_TorqueCtrl 在 91 页
- | FOC\_MTPA 在 92 页
- | FOC\_FluxRegulator 在 92 页
- | FOC\_FF\_CurrReg 在 93 页
- | FOC\_MTPAInterface\_Init 在 93 页
- | FOC\_MTPA\_Init 在 94 页
- | FOC\_FluxRegulatorInterface\_Init 在 94 页
- | FOC\_FluxRegulator\_Init 在 94 页
- | FOC\_FluxRegulator\_Update 在 95 页
- | FOC\_FF\_CurrReg\_Init 在 95 页

#### FOC\_Init

梗概 void FOC\_Init (void)

描述 初始化所有与磁场定向控制算法有关的变量值。每次电机启动优先调用。

输入 无。

返回值 无。

函数调用 当 FLUX\_WEAKENING 使能(库配置文件): FOC\_FluxRegulator\_Init  
当 IPMSM\_MTPA 使能: FOC\_MTPA\_Init  
当 FEED\_FORWARD\_CURRENT\_REGULATION 使能:  
FOC\_FF\_CurrReg\_Init

提示 在演示程序中，在 INIT 状态时调用。

#### FOC\_Model

梗概 void FOC\_Model (void)

描述 函数的功能是调节 PMSM 力矩和磁通，运行 FOC 算法。

电流控制  $i_{qs}^{**}$  和  $i_{ds}^{**}$ （在磁场定向条件下，分别控制转矩和磁通）定义在函数之外（在速度控制模式下，由速度和磁通调节器，通过 FOC\_CalcFluxTorqueRef 函数提供，而力矩控制模式下由用户通过设置 FOC\_TorqueCtrl 函数 LCD 菜单提供，见章节 3.1 说明）。

因此，作为需要的电流源，函数启动作为 CR-PWM 的功率转换器。介于此

目的，通过参数 REP\_RATE (与 PWM\_FREQ 一起)运行高性能同步 (d,q) 电流调节器，其运行频率已被定义，正如章节 4.2 所说明一样。

通过 ADC JE0C ISR 触发，函数获取定子电流(通过 ICS 或分流电阻读取)，进行 Clark 和 Park 转变，把它们转换成  $i_{qs}$  和  $i_{ds}$  (如图 4 所示)。

然后，这些电流与参考值  $i_{qs}^{**}$  和  $i_{ds}^{**}$  一起送到 PID 调节器。这时，如果在库配置中启动 FEED\_FORWARD\_CURRENT\_REGULATION, PID 调节器输出电压  $v_{qs}$  和  $v_{ds}$ ，就会加到前馈模块中的输出电压上。它们然后反过来作用到定子结构上 (通过 Reverse Park 转化)，最后驱动功率。

为了正确执行 Park and Reverse Park 转变，必须确定转子的位置 ( $\theta_{rel}$ ) (由于电流需要在相位定向和与转子磁通正交)。为了完成这项任务，根据配置 (stm32f10x\_MCconf.h 中设置)，函数能够通过编码器，霍尔传感器，或提供的无传感算法读取转子角度。

输入 无。

返回值 无。

函数调用

```
Clarke, Park, RevPark_Circle_Limitation;
PID_Regulator, Rev_Park;
如果与编码器一起工作:
ENC_Get_Electrical_Angle;
如果与霍尔传感器一起工作:
HALL_GetElectricalAngle;
如果与无传感模式一起工作:
ST0_Get_Electrical_Angle;
如果与 ‘ICS’ 一起工作:
SVPWM_IcsGetPhaseCurrentValues, SVPWM_IcsCalcDutyCycles;
如果与 ‘三相分流’ 一起工作:
SVPWM_3ShuntGetPhaseCurrentValues,
SVPWM_3ShuntCalcDutyCycles;
如果与 ‘单相分流’ 一起工作:
SVPWM_1ShuntGetPhaseCurrentValues,
SVPWM_1ShuntCalcDutyCycles.
```

FOC\_CalcFluxTorqueRef

梗概 void FOC\_CalcFluxTorqueRef (void)

描述 在速度控制模式下，函数提供电流分量  $i_{qs}^{**}$  和  $i_{ds}^{**}$  作为参考值 (函

数 FOC\_Model 会用到) (参考图 5 “速度闭环控制”)。

速度设置点和实际定子速度  $\omega_r$  在 PID 闭环中进行比较并输出  $i_{qs}^{**}$ 。然后, 如果使用的时 IPMSM 电机并且用户使用了转矩最优算法, MTPA 模块就会参考输出, 输出最合适的  $i_{ds}^*$  参考。否则,  $i_{ds}^*$  就设置为零。

在这一点上,  $i_{qs}^*$  和  $i_{ds}^*$  通过弱磁模块 (如果已经启用该项) 获取最终电流参考  $i_{qs}^{**}$  和  $i_{ds}^{**}$  的, 并且通过同步 PID 调节器和前馈模块 (如果已经启用该项) 供给电机。

输入 无。

返回值 无。

函数调用 PID\_Regulator;

如果和编码器一起工作:

ENC\_Get\_Mechanical\_Speed;

如果和霍尔传感器一起工作:

HALL\_GetSpeed;

如果和无传感模式一起工作:

ST0\_Get\_Speed\_Hz;

如果 FLUX\_WEAKENING 已经启用 (库配置文件):

FOC\_FluxRegulator;

FOC\_FluxRegulator\_Update;

如果 IPMSM\_MTPA 已经启用:

FOC\_MTPA;

如果 FEED\_FORWARD\_CURRENT\_REGULATION 已经启用:

FOC\_FF\_CurrReg.

FOC\_TorqueCtrl

梗概 void FOC\_TorqueCtrl(void)

描述 当在转矩控制模式时, 演示程序允许用户通过 LCD 菜单改变定子参考电压来改变 hTorque\_Reference 和 hFlux\_Reference 变量。函数的功能是复制这些变量值到参考电流  $i_{qs}^{**}$  和  $i_{ds}^{**}$ , 然后提供给电机用于函数 FOC\_Model。

输入 无。

输出 无。

函数调用 如果 FEED\_FORWARD\_CURRENT\_REGULATION 已经启用:

FOC\_FF\_CurrReg

提示 演示程序会以速度闭环调节频率执行函数 (参考章节 5.10.1: 可用函数

列表和中断服务程序)。

## FOC\_MTPA

梗概 `s16 FOC_MTPA(s16 hIqRef)`

描述 如章节 2.1.4 描述，函数执行 IPMSM MTPA 最优驱动，：输入  $i_q^*$ （速度控制模式下是 PI 调节器的输出）到函数，通过线性插入 MTPA 轨迹输出  $i_d^*$ 。

输入 参考电流  $i_q^*$ （有符号，16 位）

输出 参考电流  $i_d^*$ （有符号，16 位）

函数调用 无。

提示 作为初始步骤，头文件 `MC_PMSM_motor_param.h` 中的 MTPA 部分的应该根据电机的应用来建立（参考章节 4.2 详细解说）。在第一次电机启动前至少调用一次 `FOC_MTPAInterface_Init` 函数。

在演示程序中，通过库配置文件启用 MTPA 项（参考章节 4.1）

参见 图 9：19 页如结构框图所示 MTPA 控制；`FOC_MTPAInterface_Init`，`FOC_MTPA_Init` 函数。

## FOC\_FluxRegulator

梗概 `Curr_Components FOC_FluxRegulator (Curr_Components Stat_Curr_qd_ref, Volt_Components Stat_Volt_qd, s16 hVoltLevel)`

描述 如章节 2.3 所述函数执行弱磁功能。

输入 参考电流  $i_q^*$   $i_d^*$ （`Curr_Components` 结构体），参考定子电压  $v_q^*$   $v_d^*$ （`Volt_Components` 结构体），在运行时定子电压放大作为参考级（正数，16 位）。

输出 参考电流  $i_{qsat}^{**} i_d^{**}$ （`Curr_Components` 结构体）

函数调用 `PID_Regulator`

提示 作为初始步骤，`MC_PMSM_motor_param.h` 头文件的弱磁章节应该根据电机的应用建立（参考章节 4.2 详细解说）。在每次电机启动前调用 `FOC_FluxRegulatorInterface_Init` 函数。

在演示程序中，通过库配置文件启用该项（参考章节 4.1）

参见 图 13：24 页弱磁运行计划；`FOC_FluxRegulatorInterface_Init`，`FOC_FluxRegulator_Init`，`FOC_FluxRegulator_update` 函数。

## FOC\_FF\_CurrReg

梗概 `Volt_Components FOC_FF_CurrReg(Curr_Components Stat_Curr_qdref, Volt_Components Stat_Volt_qd, s16 hspeed, s16 hvbus)`

描述 函数执行前馈电流调节功能，如章节 2.1.5 所述。

输入 参考电流  $i_q^{**} i_d^{**}$  (Curr\_Components 结构体), 参考定子电压  $v_q^{*} v_d^{*}$  (Volt\_Components 结构体), 转子速度 (dpp), DC 总线电压 (正数, 16 位)

输出 参考电压  $v_q^{*} v_d^{*}$  Volt \_Components 结构体)

函数调用 无。

提示 作为初始步骤, 前 MC\_PMSM\_motor\_param.h 头文件馈章节应该根据电机的应用建立 (参考章节 4.2 详细解说)。FOC\_FF\_CurrReg\_Init t 函数必须在第一次电机前调用一次。

在演示程序中, 通过库配置启用该功能 (参考章节 4.1)

参见 图 10: 在 20 页前馈电流控制;

FOC\_FF\_CurrReg\_Init function, FOC\_MTPAInterface\_Init

梗概 void FOC\_MTPAInterface\_Init(void)

描述 根据所使用的电机和写在 MC\_PMSM\_motor\_param.h 中 MTPA 章节的参数, 初始化所有与 MTPA 轨迹发生器相关的变量, 得到适当的值 (FOC\_MTPA 函数)。在第一次电机前至少调用一次。

输入 无 (从 MC\_PMSM\_motor\_param.h 中读取参数)

输出 无

函数调用 FOC\_MTPA\_Init

提示 无

参见 MC\_type.h 结构声明; FOC\_ Init, FOC\_MTPA 函数

FOC\_MTPA\_Init

梗概 void FOC\_MTPA\_Init(MTPA\_Const MTPA\_InitStructure\_in, s16 hldDemag\_in)

描述 通过 FOC\_MTPAInterface\_Init 调用函数根据 MC\_PMSM\_motor\_param.h 文件中定义的默认参数初始化 MTPA 算法

输入 MTPA 初始化结构体 (MTPA\_Const 结构体), 最大允许参考电流  $i_d^{*}$

输出 无

函数调用 无

提示 无

参见 MC\_type.h for structure declarations; FOC\_MTPAInterface\_Init 函数

FOC\_FluxRegulatorInterface\_Init

梗概 void FOC\_FluxRegulatorInterface\_Init(void)

描述 根据所使用的电机和 MC\_PMSM\_motor\_param.h 节弱磁参数, 初始化与弱



磁运行相关的所有变量（FOC\_Flux\_Regulator 函数）为合适的值。在每次电机启动前调用。

输入 无（从 MC\_PMSM\_motor\_param.h 中读取参数）

输出 无

函数调用 FOC\_FluxRegulator\_Init

提示 无

参见 MC\_type.h 结构声明；FOC\_Init, FOC\_FluxRegulator 功能

FOC\_FluxRegulator\_Init

梗概 void FOC\_FluxRegulator\_Init(PID\_Struct\_t\*PI\_Stat\_Volt\_Init  
Structure\_in, s16 hNominalCurrent\_in)

描述 函数由 FOC\_FluxRegulatorInterface\_Init 调用，根据  
MC\_PMSM\_motor\_param.h 中定义的默认参数初始化弱磁算法

输入 指向 PID 实例结构体（PID\_Struct\_t structure），电机名义电流（正数  
16 位）

输出 无

函数调用 无

提示 无

参见 MC\_type.h 为结构声明；FOC\_FluxRegulatorInterface\_Init 函数

FOC\_FluxRegulator\_Update

梗概 s16 FOC\_FluxRegulator\_Update(s16 hKpGain, s16 hKiGain)

描述 根据用户输入，修改用于弱磁模块的 PI 调节器中的比例和积分增益（如  
图 13：第 24 页弱磁运行流程）

输入 比例增益（正数 16 位），积分增益（正数 16 位）

输出 定子电压放大（无符号 16 位）

函数调用 无

提示 无

参见 无

FOC\_FF\_CurrReg\_Init

梗概 void FOC\_FF\_CurrReg\_Init(s32 wConstant1Q, s32 wConstant1D, s32  
wConstant2)

描述 根据所使用的电机和写在 MC\_PMSM\_motor\_param.h 节写定的前馈  
参数，初始化与前馈运行相关的所有变量（FOC\_FF\_CurrReg 函数）。  
在初次电机启动前至少调用一次。

输入 来源 MC\_PMSM\_motor\_param.h 相关章节有符号 32 位参数

输出 无

函数调用 无

提示 参见 Function FOC\_Init, FOC\_FF\_CurrReg

## 5.5 参考坐标转换： MC\_Clarke\_Park 模块

这个单元用于 AC 设备（感应，同步和 PMSM），转化在不同的速度下旋转的参考坐标的电量。

依据随机参考坐标理论，这个单元提供三个函数，以两位电气机器分析先驱者 E. Clarke and R.H. Park 命名。

这些函数计算实现磁场定向控制（FOC）的三个变量：

- I Clarke 将定子电流转变为静止直角参考坐标（称  $\alpha$   $\beta$  坐标，如图 57）；
- I 然后，Park 将电流转变为按随机的速度旋转的坐标为（磁场定向控制，与转子同步）；
- I 反向 Park 变换使反电动势从旋转坐标（ $q$ ,  $d$ ）到静止坐标。

这个单元也包含三角函数（正弦和余弦），和一个校正电压矢量控制函数（即为循环限制）。

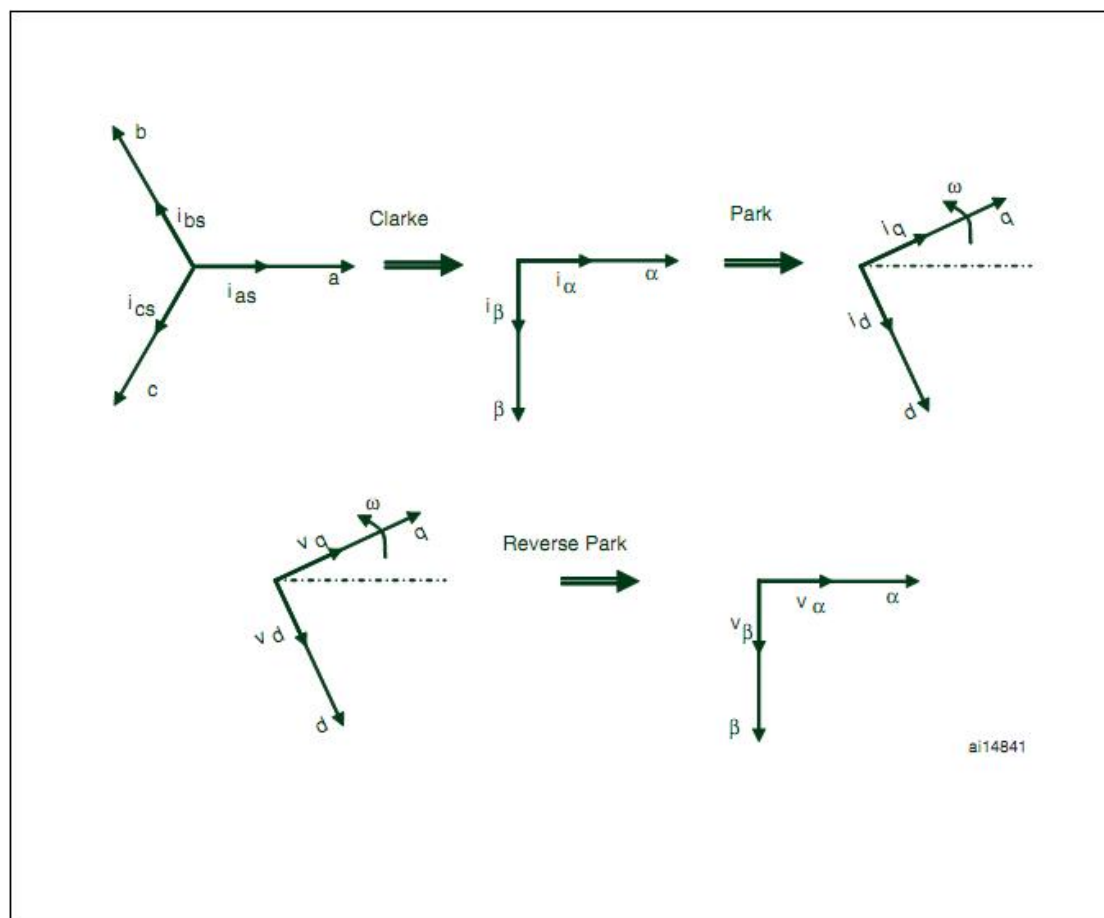


图 57 Clarke, Park 和 Park 逆变换

### 5.5.1 可用 C 函数列表

- I Clarke 第 97 页
- I Park 第 97 页
- I Rev\_Park 第 97 页
- I Rev\_Park\_Circle\_Limitation 第 98 页
- I Trig\_Functions 第 98 页

#### Clarke

梗概 Curr\_Components Clarke (Curr\_Components Curr\_Input)

描述 函数将定子电流  $i_{as}$  和  $i_{bs}$  (直接沿轴线间隔 120 度) 转化为静止参考坐标 ( $\alpha$   $\beta$ ) 中的电流  $i_\alpha$  和  $i_\beta$ ;  $\alpha$   $\beta$  轴间隔直角。详情参考章节 5.5.2。

输入 定子电流  $i_{as}$  和  $i_{bs}$  (q1.15 格式) 作为变量 Curr\_Input 成员, Curr\_Components 类型结构体。

返回值 定子电流  $i_\alpha$  和  $i_\beta$  (q1.15 格式) 作为 Curr\_Components 类型结构体成员。

函数调用 无。

#### Park

梗概 Curr\_Components Park (Curr\_Components Curr\_Input, s16 Theta)

描述 函数功能是将静止 ( $\alpha$   $\beta$ ) 参考坐标系的定子电流  $i_\alpha$  和  $i_\beta$  转化为与定子和合适定向同步的参考坐标系电流  $i_{qs}$  和  $i_{ds}$ 。参考章节 5.5.2 详解。

输入 作为 Curr\_Input 变量成员的定子电流  $i_\alpha$  和  $i_\beta$  (q1.15 格式), 是 Curr\_Components 类型结构体; 转子角度  $\theta_{rel}$  (每转 65536 个脉冲)。

返回值 定子电流  $i_{qs}$  和  $i_{ds}$  (q1.15 格式) 是 Curr\_Components 类型结构体成员。

函数调用 Trig\_Functions

#### Rev\_Park

梗概 Volt\_Components Rev\_Park (Volt\_Components Volt\_Input)

描述 函数将属于与定子同步的旋转坐标系定子电压  $V_q$  和  $V_d$  转化为到静止参考坐标系下的电压, 即  $v_\alpha$  和  $v_\beta$ 。参考章节 5.5.2 详解。

输入 定子电压  $v_{qs}$  和  $v_{ds}$  是 Volt\_Components 类型结构体成员。

返回值 定子电压  $v_\alpha$  和  $v_\beta$  (q1.15 格式) 是 Volt\_Components 类型结构体成员。

函数调用 无。

#### Rev\_Park\_Circle\_Limitation

梗概 void RevPark\_Circle\_Limitation(void)

描述 在两个新定子电压量 ( $v_d$  和  $v_q$ ) 在定子上产生磁通和转矩电流之后，通过磁通和转矩 PID 单独计算，必须在通过 Rev\_Park 函数之前使结果矢量幅值饱和等于  $\sqrt{v_d^2 + v_q^2}$ 。这些程序的功能是达到饱和。参考章节 5.5.3: 第 101 页循环限制。

输入 无。

返回值 无。

提示 定子电压矢量限制必须和 PWM 频率一致，如表 2 所示：73 页三相拓扑中 PWM 频率和最大占空比关系。

函数调用 无。

Trig\_Functions

梗概 Trig\_Components Trig\_Functions(s16 hAngle)

描述 函数根据输入角返回三角余弦和正弦函数值。

输入 一个角度，s16 格式（弧度与图 58 说明一致）

返回值 输入角为余弦和正弦，s16 格式（如图 59）作为 Trig\_Components 类型结构体成员。

函数调用 无。

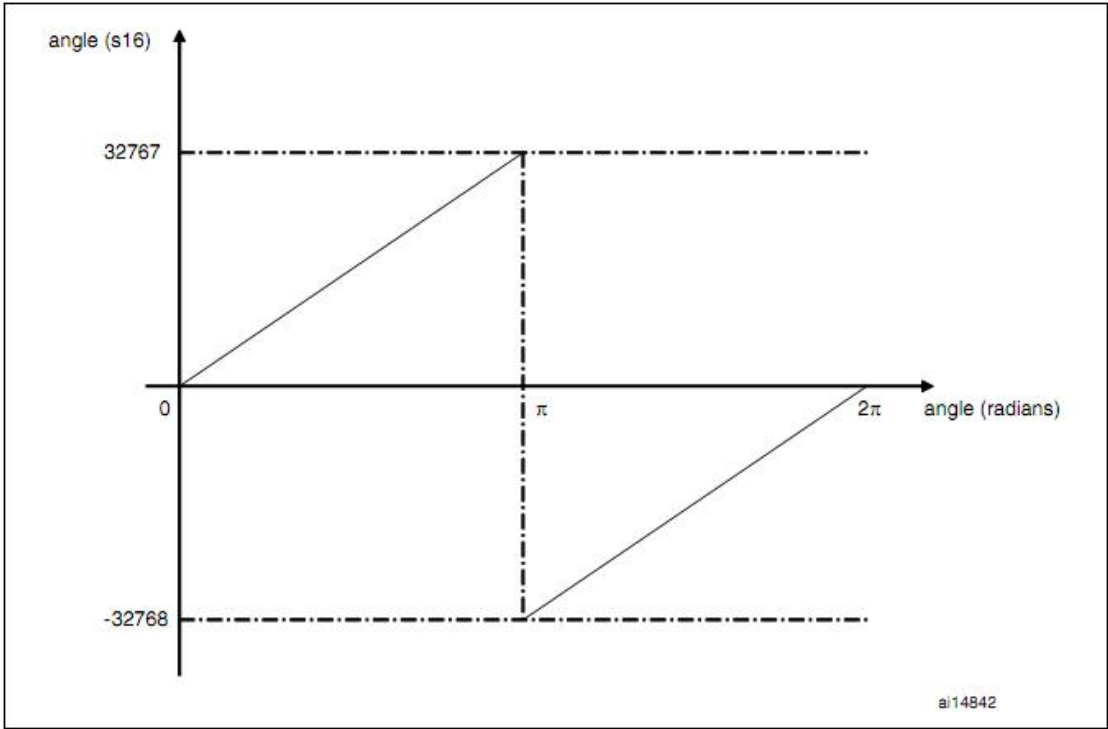


图 58 s16 与弧度

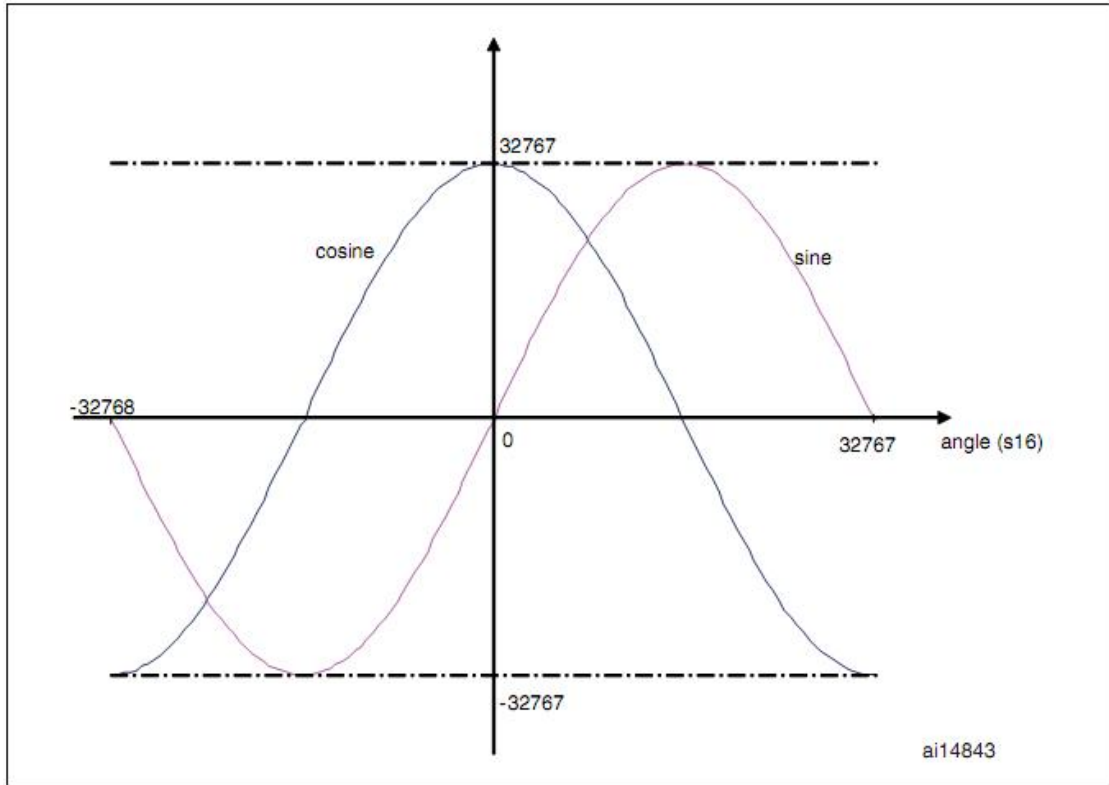


图 59 s16 的正弦和余弦

### 5.5.2 详解参考坐标系转换

PM 同步电机电压方程式非常复杂，也是随时间可变的。通过改变与转子同步旋转参考坐标系定子来量，可以简化这些方程式的复杂程度。

这种策略主要指参考坐标系理论[1]。

假定  $f_{ax}$ ,  $f_{bx}$ ,  $f_{cx}$  是三相瞬时量，沿着轴线间隔 120 度，x 可以用 s 或 r 替换以看作定子或转子量（如图 60）；假定  $f_{qx}$ ,  $f_{dx}$ ,  $f_{0x}$  是它们转换后对应的值，间隔 90 度；转换到参考坐标的方程式（以任意角速度  $\omega$  旋转）如下表达：

$$f_{qdx} = \begin{bmatrix} f_{qx} \\ f_{dx} \\ f_{0x} \end{bmatrix} = \frac{2}{3} \times \begin{bmatrix} \cos\theta & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ \sin\theta & \sin\left(\theta - \frac{2\pi}{3}\right) & \sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} f_{ax} \\ f_{bx} \\ f_{cx} \end{bmatrix}$$

$\theta$  是观察时的参考坐标系的移位角 (q, d)， $\theta_0$  在  $t=0$  时的移位（如图 60）。

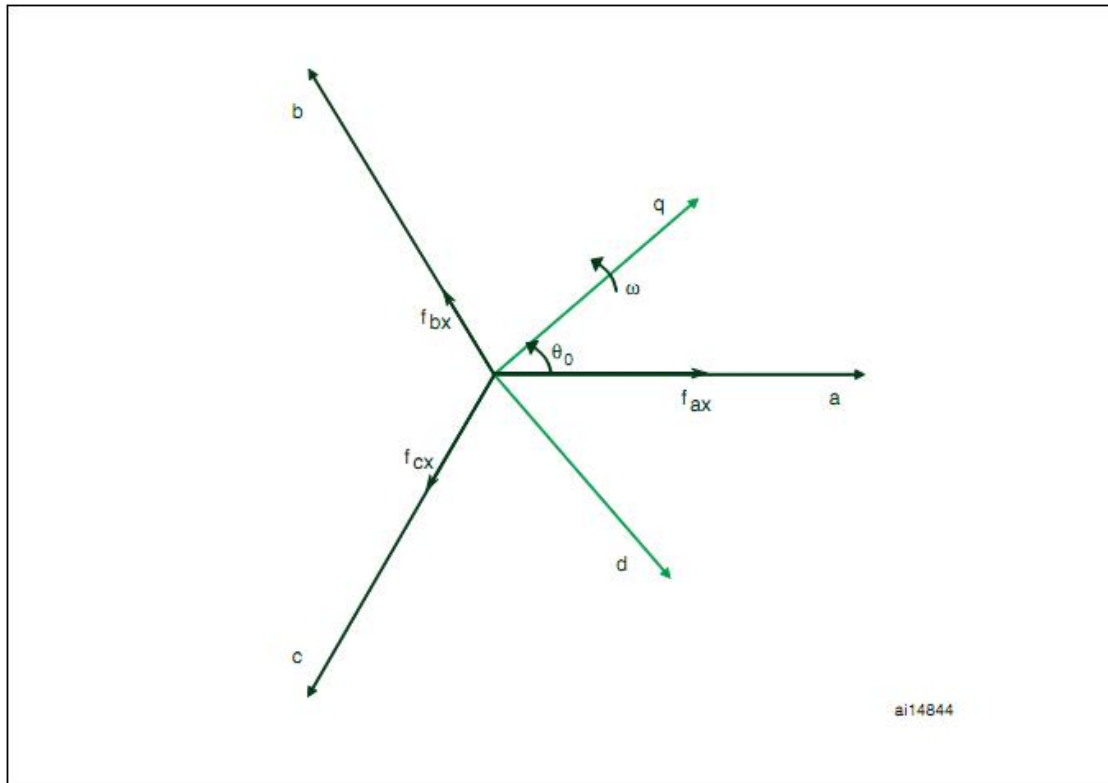


图 60 从 abc 静止坐标到旋转坐标 (q, d) 的转换。

使用 Clark 转换，可以将定子电流  $i_{as}$  和  $i_{bs}$ （直接沿着轴线每 120 度替换）转化为静止参考坐标系 ( $\alpha$   $\beta$ ) 电流  $i_\alpha$  和  $i_\beta$ 。

在一般方程式产生适当的替换：

$$i_\alpha = i_{as}$$

$$i_\beta = -\frac{i_{as} + 2i_{bs}}{\sqrt{3}}$$

定子电流变量  $i_\alpha$  和  $i_\beta$  属于静止参考坐标 ( $\alpha$   $\beta$ )，在 Park 变换中将这两个变量转化为与转子同步旋转和定向的参考坐标系，以便 d 轴与永磁磁通量对齐，以获得  $i_{qs}$  和  $i_{ds}$ 。

因此，根据参考坐标系的选择则有：

$$i_{qs} = i_\alpha \cos \theta_r - i_\beta \sin \theta_r$$

$$i_{ds} = i_\alpha \sin \theta_r + i_\beta \cos \theta_r$$

另一方面，Park 逆变换将属于旋转和转子定向的参考坐标系的定子电压  $v_q$  和  $v_d$  转换为静止参考坐标系下的  $v_\alpha$  和  $v_\beta$ ：

$$v_\alpha = v_{qs} \cos \theta_r + v_{ds} \sin \theta_r$$

$$v_\beta = -v_{qs} \sin \theta_r + v_{ds} \cos \theta_r$$

### 5.5.3 循环限制

如上论述，FOC 允许单独控制三相永磁电机的转矩和磁通。定子电压新值 ( $v_d^*$  和  $v_q^*$ ) 产生磁通和转矩定子电流，在通过磁通和转矩 PID 计算之后，在进行 Reverse Park 转换之前必须使结果矢量 ( $\vec{V}^*$ ) 饱和，最后再传到 SVPWM 模块。

饱和边界通常由值 (S16\_MAX=32767) 给出，这个值会产生最大输出电压幅值 (对应占空比从 0% 到 100%)。

然而，当使用单相或三相分流电阻配置时，根据 PWM 频率，可能需要限制最大 PWM 占空比，以保证读取定子电流。

介此原因，在单相或三相分流电阻配置下，根据 PWM 开关频率，饱和边界可以是略小于 S16\_MAX 的值。

第 73 页表 2 和第 82 页表 4 显示的是当使用 STM3210B-MCKIT 三相和单相分流拓扑 PWM 开关频率功能的最大适用调制指数。附录 A.8: MMI (最大调制指数): 自动计算解释如何给定 PWM 频率和干扰参数计算 MMI (最大调制指数)。

RevPark\_Circle\_Limitation 功能讨论定子电压元件饱和状态，如图 61 所示。

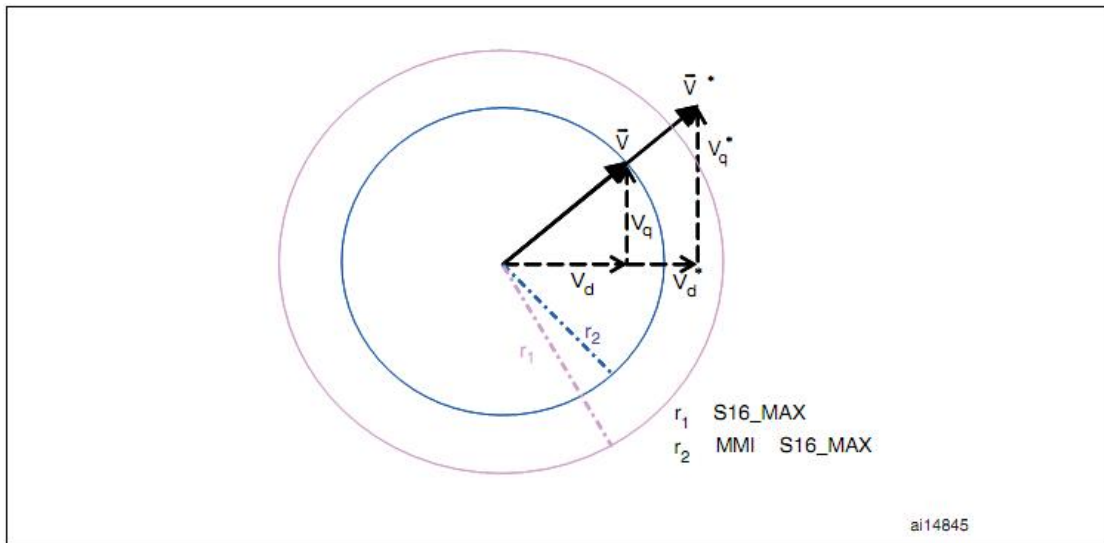


图 61 循环限制工作原理

$V_d$  和  $V_q$  描述的是传给 Reverse Park 转换功能的饱和定子电压，而  $V_d^*$  和  $V_q^*$  是 PID 电流控制的输出。从几何关系考虑，可以得到下列关系：

$$v_d = \frac{v_d^* \cdot \text{MMI} \cdot \text{S16\_MAX}}{|\vec{V}^*|}$$

$$v_q = \frac{v_q^* \cdot \text{MMI} \cdot \text{S16\_MAX}}{|\vec{V}^*|}$$

为了快速计算上述公式并保证足够的精度，值

$$\frac{\text{MMI} \cdot S16 \cdot \text{MAX}^2}{|\vec{V}|}$$

可以经过计算并对不同的值 $|\vec{V}|$ 存储在查询表格中。此外，根据选择的 PWM 频率考虑 MMI，许多查询表格存储在 MC\_Clarke\_Park.c（MMI 从 91 变化到 100）。

一旦选择了所需的 PWM 开关频率，就不能取消对与在 Max Modulation Index 列表中定义的 PWM 开关频率对应的 MC\_Control\_Param.h 注释。

```
//#define MAX_MODULATION_100_PER_CENT    //达到 11.4 kHz PWM 频率
//#define MAX_MODULATION_99_PER_CENT     //达到 11.8 kHz PWM 频率
//#define MAX_MODULATION_98_PER_CENT     //达到 12.2 kHz PWM 频率
//#define MAX_MODULATION_97_PER_CENT     //达到 12.9 kHz PWM 频率
//#define MAX_MODULATION_96_PER_CENT     //达到 14.4 kHz PWM 频率
//#define MAX_MODULATION_95_PER_CENT     //达到 4.8 kHz PWM 频率
//#define MAX_MODULATION_94_PER_CENT     //达到 15.2 kHz PWM 频率
//#define MAX_MODULATION_93_PER_CENT     //达到 16.7 kHz PWM 频率
//#define MAX_MODULATION_92_PER_CENT     //达到 17.1 kHz PWM 频率
//#define MAX_MODULATION_91_PER_CENT     //达到 17.5 kHz PWM 频率
```

关于 PWM 开关频率选择的信息，可以参考第 136 页中章节 A.2。为了计算与 PWM 开关频率对应的最大调制指数，参考第 73 页表 2 和第 82 页表 4。正如之前指明的一样，如果使用 ICS，可以允许 MMI 选择 100%，而不需要考虑选择的 PWM 频率。

## 5.6 编码器反馈处理：stm32f10x\_encoder 模块

### 5.6.1 可用函数列表和中断服务程序

以下是 stm32f10x\_encoder.h 的头文件中可用函数列表：

ENC_Init	第 103 页
ENC_Get_Electrical_Angle	第 103 页
ENC_Get_Mechanical_Angle	第 104 页
ENC_ResetEncoder	第 104 页
ENC_Clear_Speed_Buffer	第 104 页
ENC_Get_Mechanical_Speed	第 104 页
ENC_Calc_Average_Speed	第 105 页
ENC_ErrorOnFeedback	第 105 页
ENC_Start_Up	第 105 页



## ENC\_Init

梗概: `void ENC_Init(void)`

描述: 这个函数的目的是初始化编码器计时器。启动外设时钟、输入引脚和更新中断。外设时钟配置为 4 倍频模式, 这意味着该计数器在计时器 1 和计时器 2 (TIMx\_CH1 和 TIMx\_CH2 引脚) 上升沿/下降沿时计数递增/递减。

函数调用: `RCC_APB1PeriphClockCmd`, `RCC_APB2PeriphClockCmd`,  
`GPIO_StructInit`, `GPIO_Init`, `NVIC_Init`, `TIM_DeInit`,  
`TIM_TimeBaseStructInit`, `TIM_TimeBaseInit`,  
`TIM_EncoderInterfaceConfig`, `TIM_ICInit`, `TIM_ClearFlag`,  
`TIM_ITConfig`, `TIM_Cmd`

参见: **STM32F103xx 参考手册: TIMx 编码器接口模式**

## ENC\_Get\_Electrical\_Angle

梗概: `s16 ENC_Get_Electrical_Angle(void)`

描述: 这个函数以 s16 类型返回电角度。函数返回: 0 为 0 度, -32768 为 -180 度 (S16\_MIN), +32767 (S16\_MAX) 为 180 度。

输入: 无

输出: 有符号 16 位

调用: 无

## ENC\_Get\_Mechanical\_Angle

梗概: `s16 ENC_Get_Mechanical_Angle(void)`

描述: 这个函数以 s16 类型返回机械角度。函数返回: 0 为 0 度, -32768 为 -180 度 (S16\_MIN), +32767 (S16\_MAX) 为 180 度。

输入: 无

输出: 有符号 16 位

调用: 无

提示: 电气频率、机械频率、转速之间的关系:

电频率 = 磁极对数  $\times$  机械频率

RPM 的速度 =  $60 \times$  机械频率 (转速: 每分钟转速)

例如: 电频率 = 100 赫兹, 电机 8 对磁极对数:

$100\text{Hz (电频率)} / 8 = 12.5\text{Hz (机械频率)}$   
 $12.5 \times 60 = 750$  (转速)

## ENC\_ResetEncoder

梗概: `void ENC_resetEncoder(void)`

描述: 这个函数将在 `MC_encoder_param.h` 里相应校准角度的值写进编码器计时器的寄存器。这个函数在任何校准阶段结束后都将被调用。

调用：无

ENC\_Clear\_Speed\_Buffer

梗概： **void ENC\_Clear\_Speed\_Buffer(void)**

描述： 此函数为存储平均速度位缓冲区

函数调用：无

ENC\_Get\_Mechanical\_Speed

梗概： **s16 ENC\_Get\_Mechanical\_Speed(void)**

描述： 这个函数以频率为单位返回转速。返回值的分辨率为 0.1Hz，这意味着 1234 等于 123.4 赫兹

输入： 无

输出： 有符号 16 位

函数调用： 无

提示： 这个程序返回转子的频率。为了得到电速度，使用下式转换：

电频率=极对数\*频率机械

ENC\_Calc\_Average\_Speed

梗概： **void ENC\_Calc\_Average\_Speed(void)**

描述： 此函数每 SPEED\_MEAS\_TIMEBASE 毫秒必须调用；它计算最新的测量速度，如果超出 MC\_encoder\_param.h 指定的范围，那么错误计数器增加，速度值饱和。此外，如果错误计数器的值超过 MAXIMUM\_ERROR\_NUMBER，就会设置存储器错误状态布尔变量。最后，基于最新的 SPEED\_BUFFER\_SIZE 测量速度计算平均速度。

如果预编译时 MINIMUM\_MECHANICAL\_SPEED\_RPM 已被设置为零，用户可以忽视警告信息 “pointless comparison of unsigned integer with zero”。

函数调用： **ENC\_Calc\_Rot\_Speed**

输入： 无

返回： 无

ENC\_ErrorOnFeedback

梗概： **bool ENC\_ErrorOnFeedback(void)**

描述： 这个函数只是返回 ENC\_Calc\_Average\_Speed 函数中每隔 SPEED\_MEAS\_TIMEBASE 毫秒更新一次的测量速度是否错误的布尔状态变量。在运行状态下，主函数检查速度反馈是否故障时（如编码器掉电），在标准函数库中调用这个函数。

函数调用：无

输入： 无

返回：布尔值，如果出现错误返回 TRUE，否则返回 FALSE。

ENC\_Start\_Up

梗概： **void ENC\_Start\_Up (void)**

描述： 这个函数的目的是在调整永磁同步电机的相位时，调节转矩和定子磁通电流分量  $I_q$  和  $I_d$ 。该函数在调整结束时还更新主机状态（从启动到运行）

函数调用：SVPWM\_3ShuntGetPhaseCurrentValues or

SVPWM\_IcsGetPhaseCurrentValues, Clarke, Park, PID\_Regulator, RevPark\_Circle\_Limitation, SVPWM\_IcsCalcDutyCycles or SVPWM\_3ShuntCalcDutyCycles, ENC\_ResetEncoder,

ENC\_Clear\_Speed\_Buffer

输入：无

返回值：无

## 5.7 霍尔传感器反馈处理：stm32f10x\_hall 模式

### 5.7.1 可用函数列表

以下是在头文件 stm32f10x\_hall.h 中列出的可用函数列表：

I	HALL_HallTimerInit	第 106 页
I	HALL_GetRotorFreq	第 107 页
I	HALL_GetSpeed	第 107 页
I	HALL_InitHallMeasure	第 108 页
I	HALL_IsTimedOut	第 108 页
I	HALL_GetElectricalAngle	第 108 页
I	HALL_IncElectricalAngle	第 109 页
I	HALL_Init_Electrical_Angle	第 109 页
I	HALL_ClrTimeOut	第 109 页

#### HALL\_HallTimerInit

梗概 void HALL\_HallTimerInit(void)

描述 这个函数的功能是初始化霍尔传感器反馈过程中的外设。特别是，与霍尔传感器连接的 GPIO 输入引脚被初始化为浮动输入，定时器 TIMx 配置为“清晰捕获”模式并且启动 XOR 输入函数，预分频器由 HALL\_MAX\_RATIO 初始化。最后启动 TIMx 输入捕捉（XORed 下降沿信号）和溢出（更新）事件中断功能。

函数调用 RCC\_APB1PeriphClockCmd, RCC\_APB2PeriphClockCmd, GPIO\_StructInit, GPIO\_Init, TIM\_DeInit, TIM\_TimeBaseStructInit, TIM\_TimeBaseInit, TIM\_ICStructInit, TIM\_ICInit, TIM\_PrescalerConfig, TIM\_InternalClockConfig, TIM\_SelectHallSensor, TIM\_SelectInputTrigger, TIM\_SelectSlaveMode, TIM\_UpdateRequestConfig, NVIC\_Init, TIM\_ClearFlag, TIM\_ITConfig, TIM\_SetCounter, TIM\_Cmd

另见 STM32F103xx 参考手册，“霍尔传感器接口”部分

#### HALL\_GetRotorFreq

梗概 s16 HALL\_GetRotorFreq(void)

描述 此例程用差示脉冲极谱法方式计算转子电频率，从最新的测量周期存储数据开始并且根据以下计算公式：

$$w_{dpp} = \frac{CKTIM \ 2^{16}}{3 \text{ SAMPLING\_FREQ 捕获值 预分频值}}$$

其中 **CKTIM** 是定时器外设时钟, **SAMPLING\_FREQ** 为 **FOC** 算法采样频率。如果预分频器等于最大值或超时发生则速度为零。详情请参阅第 5.7.2 速度的测量原理。

函数调用 **GetAvrgHallPeriod** or **GetLastHallPeriod**

输入 无

返回 返回脉冲单位的电频率, 格式为带符号 16 位。

另见 附录 A.7 节: 有关速度脉冲单位详情。

**HALL\_GetSpeed**

梗概 **s16 HALL\_GetSpeed(void)**

描述 这个例程以 0.1 赫兹分辨率计算转子机械频率, 从最新的测量周期存储数据开始并且根据以下计算公式:

$$w_{dpp} = \frac{\text{CKTIM} \cdot 10}{3 \cdot \text{POLE\_PAIR\_NUM} \cdot \text{捕获值} \cdot \text{预分频值}}$$

其中 **CKTIM** 是定时器外设时钟, **POLE\_PAIR\_NUM** 是磁极对数。如果预分频器等于最大值或超时发生则返回值是 0, 超速 (或高频信号) 将导致返回预定义的值 (**HALL\_MAX\_SPEED**)。

函数调用 **GetAvrgHallPeriod** or **GetLastHallPeriod**.

输入 无

返回值 返回以 0.1 赫兹分辨率为单位的转子速度, 格式为带符号 16 位。

**HALL\_InitHallMeasure**

梗概 **void HALL\_InitHallMeasure(void)**

描述 当最新的速度信息“推”进时清除软件 **FIFO**。在启动电机前调用, 初始化速度测量过程。

函数调用 **HALL\_ClrCaptCounter**, **TIM\_SetCounter**, **TIM\_Cmd**, **TIM\_ITConfig**

输入 无

返回值 无

另见 A.7 节更多关于速度脉冲单位详情

**HALL\_IsTimedOut**

梗概 **bool HALL\_IsTimedOut(void)**

描述 这个函数只是返回测速超时的状态值 (以布尔变量形式), 在标准库函数中, 当主函数检查可能出现速度反馈错误时 (比如断线), 在运行状态中由 **main.c** 调用。

函数调用 无

输入 无

返回值 布尔变量, 超时返回 **TURE**, 否则返回 **FLASE**

**HALL\_GetElectricalAngle**

梗概 **s16 HALL\_GetElectricalAngle(void)**

描述 这个函数导出含有转子电角度信息的私有变量, 在本库中, 在运行 **FOC** 算法时被调用, 因为转子电角度是定子电流进行 **Park** 转换必不可少的部分。

函数调用 无

输入 无

返回值 电角度（以 S16 形式）

另见 有关参考坐标系转换见 5.5.3 节

**HALL\_IncElectricalAngle**

梗概 **void HALL\_IncElectricalAngle(void)**

描述 正如将在后面讨论到的，该软件变量包含的转子电角度信息与反馈信息同步，反馈信息来自每次电机三个霍尔传感器输出 XOR 有效转换。此外，为了提高连续两次有效转换的精度，每次执行 FOC 算法时累加最新测量的速度（速度脉冲形式）来增加转子电角度信息。**HALL\_IncElectricalAngle** 函数累加速度并且必须以比 FOC 算法快的采样频率调用。

函数调用 无

输入 无

返回值 无

**HALL\_Init\_Electrical\_Angle**

梗概 **void HALL\_Init\_Electrical\_Angle(void)**

描述 霍尔效应传感器是“绝对”的，因此只需读取其输出就可以重建转子位置。利用这个工作原理，在任何电机启动前，这个函数初始化包含当前电角度的软件变量。这个函数通过读取H1、H2和H3的信号状态（私有函数**ReadHallState**执行该任务）和初始化软件变量执行。可获得的最大精度为±30度，（即30/POLE\_PAIR\_NUM 的机械度）

输入 无

返回值 无

**HALL\_ClrTimeOut**

梗概 **HALL\_ClrTimeOut**

描述 该函数将包含超时错误标志的布尔变量设置为 **FALSE**，表面指示信息丢失，或速度锐减。

函数调用 无

输入 无

返回值 无

## 5.7.2 速度测量实施

由于 **STM32F103xx** 具有通用定时器（**TIMx**）功能，微控制器与三个霍尔传感器接口非常简单。事实上，当寄存器 **TIMx\_CR2** 的 **TI1S** 位被设定时，在 **TIMx\_CH1**，**TIMx\_CH2** 和 **TIMx\_CH3** 引脚的三个信号进行异或运算，由此产生的信号输入到逻辑运行 **TIMx** 输入捕获中。这样，速度的测量转换成一个方波的周期测量，频率比真正的电频率高出三倍。唯一的例外的是滚动方向无法从异或信号提取，必须直接与三个霍尔传感器输出反向。

**滚动方向识别**

如图 62 所示，将霍尔传感器输出与根据作为三位数二进制数的 H3-H2- H1（H3 是最重要的位）获得的状态数结合起来是可以的。

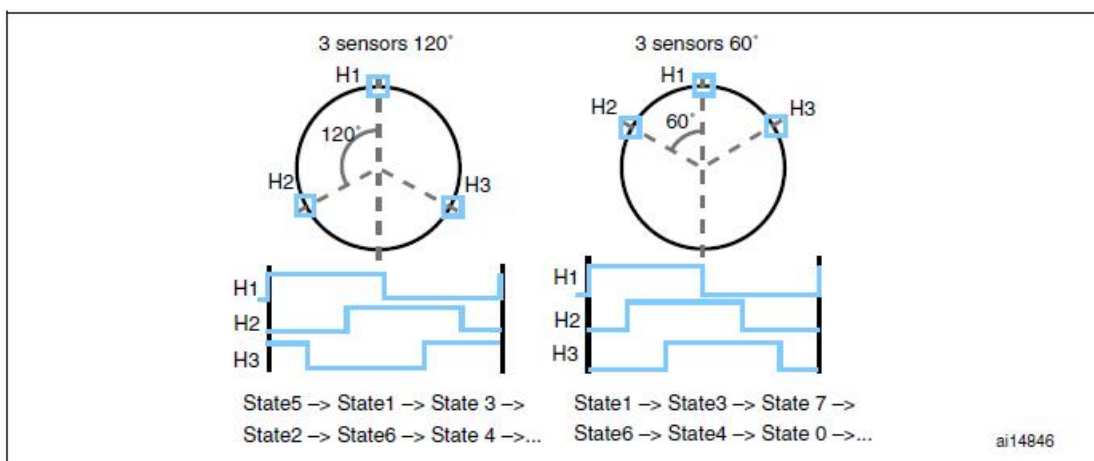


图62 霍尔传感器，对应的输出状态

因此，可以通过比较当前状态和前一个状态推断出转子转动方向，并且可以认为这是一个正方向，序列必定是图62中的一种。

### 周期测量

虽然用定时器测量周期的原理很简单，但保持最佳的分辨率很重要，特别是对于信号，例如讨论中的，比值可以轻松达到 1: 1000。为了保持最佳的分辨率，在运行时要不断调节定时器的时钟分频器。其基本原理是，如果捕获的值太低了（短周期的例子见图 63）就加快定时器；定时器在连续两个捕获之间溢出（见图 64 大周期的例子）时就放慢。

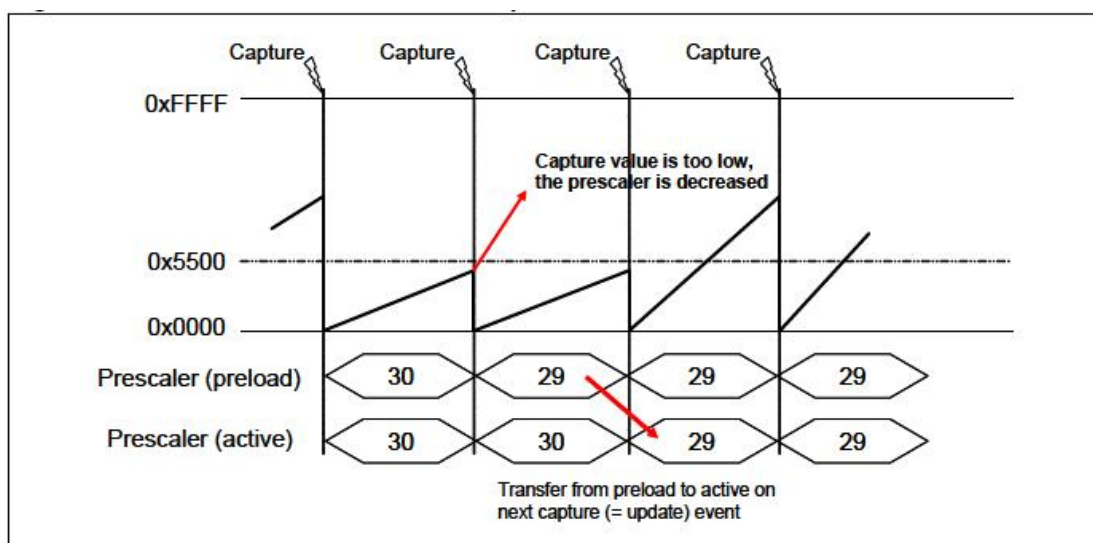


图 63 霍尔传感器定时器接口预分频器的下降

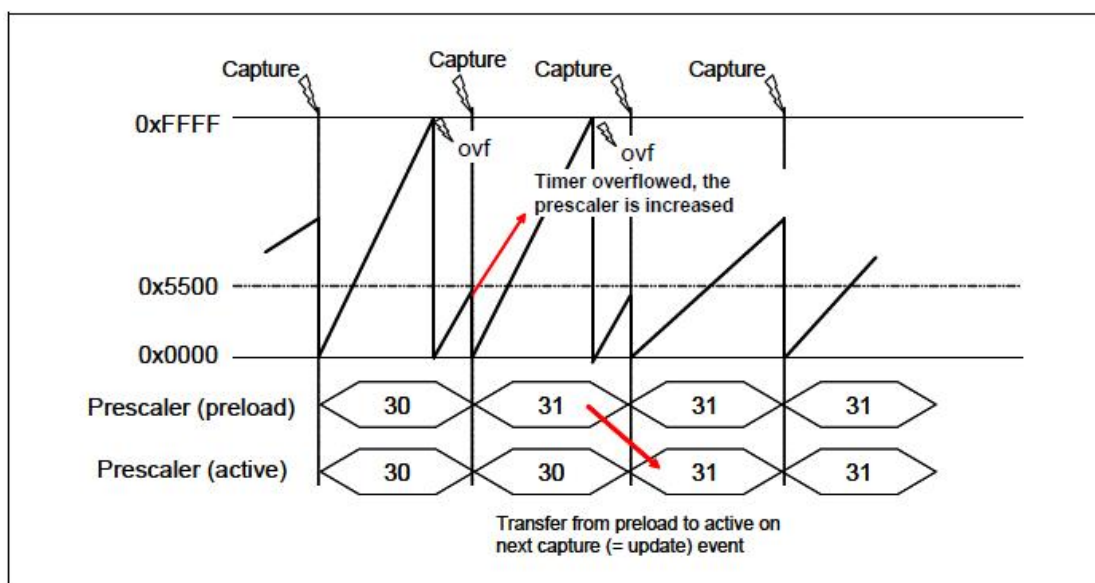


图 64 霍尔传感器定时器接口预分频器的上升

在捕捉中断中修改预分频器，利用缓存的优势：只有在下次捕捉事件中才由硬件更新新的预分频的值，而不会影响到测量。

图 65 所示的是流程图提供的进一步信息，总结了对 TIMx\_IRQHandler 采取的操作。

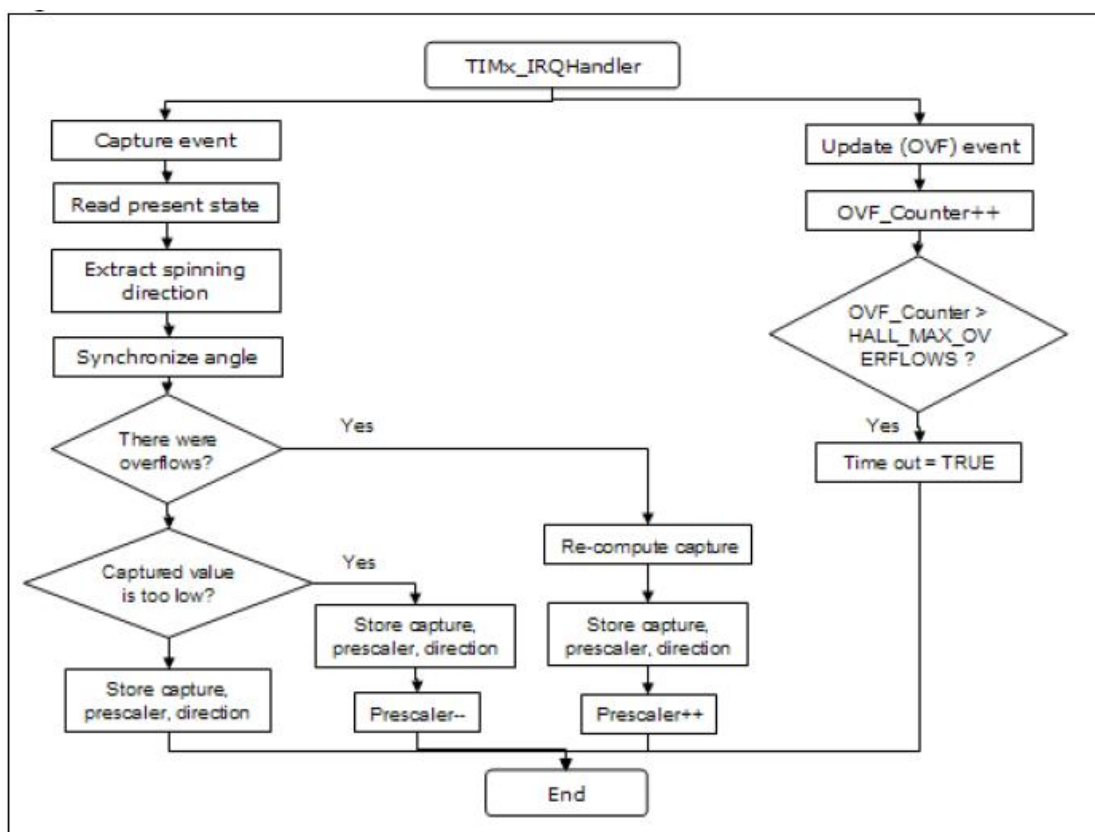


图 65 TIMx\_IRQHandler 流程图

### 5.7.3 电角度推算方法

如图 65 所示，在 TIMx\_IRQHandler 中不止完成速度测量。除了测速，异或信号的高低电平转换使得包含当前电角度的软件变量同步变得可能。

事实上，图 66 可以看到任意霍尔传感器转化都可以精确地提供转子的位置信息。

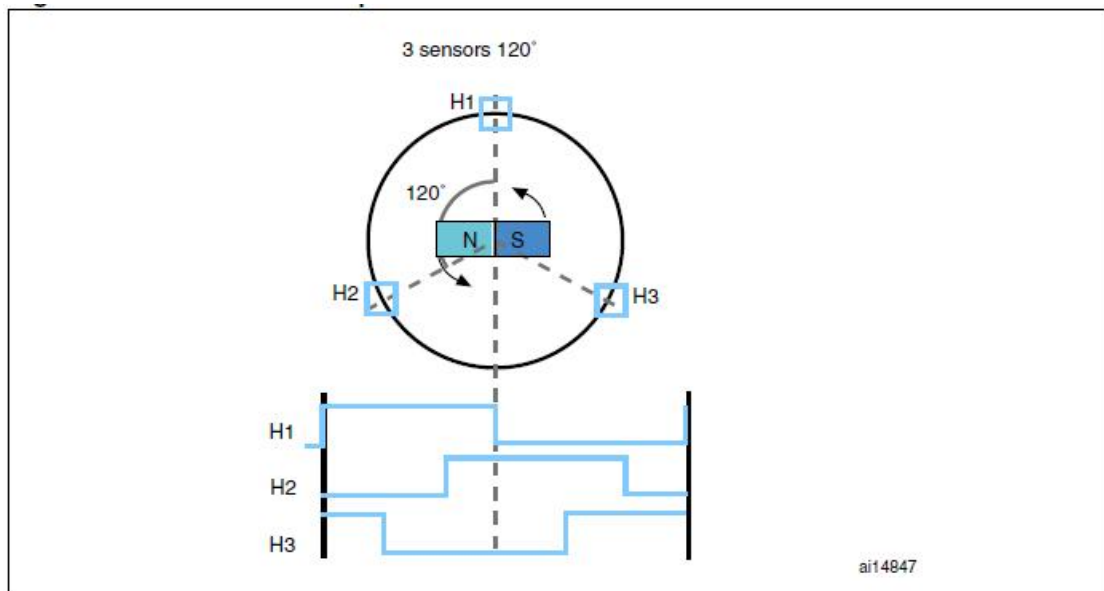


图 66 霍尔传感器输出转换

出于这个原因，在解决方案中，每一次 IC 发生电角度都与依赖霍尔输出状态的角度同步，指明方向和 PHASE\_SHIFT（如何测量见 3.12 节）。

此外，使用 FOC 算法需要及时准确获得转子位置信息，包括异或信号两个连续下降沿（每 120 电角度度发生）。所以任何插入电角度信息非常重要。为此，每次执行 FOC 算法都将最新的脉冲单位测量的速度添加到当前的电角度软件变量中。

## 5.8 无传感器速度 / 位置检测：MC\_State\_Observe 和 MC\_State\_Observer\_Interface 模块

MC\_State\_Observer 模块是为永磁同步电动机设计的，用于观测反电动势状态和锁相环（PLL）电路。它能够检测转子角位置和速度。

此外，该模块处理输出数据，通过这样做可以安全检测转子是否卡死或故障。

MC\_State\_Observer\_Interface 模块首先作为第一个的接口，提供电机参数和状态检测默认增益。

MC\_State\_Observer 模块是无传感器算法的引擎，是编译对象文件；源代码可以免费从 ST 获取：请联系您最近的 ST 销售办事处。

### 5.8.1 可用 C 函数列表

以下是 MC\_State\_Observer\_Interface.h 和 MC\_State\_Observer.h 头文件中



可用函数列表:

	STO_Init	第 114 页
	STO_StateObserverInterface_Init	第 115 页
	STO_Obs_Gains_Update	第 115 页
	STO_Calc_Rotor_Angle	第 115 页
	STO_Calc_Speed	第 116 页
	STO_InitSpeedBuffer	第 116 页
	STO_Get_Electrical_Angle	第 116 页
	STO_Get_Mechanical_Angle	第 116 页
	STO_Get_Speed	第 117 页
	STO_Get_Speed_Hz	第 117 页
	STO_IsSpeed_Reliable	第 117 页
	STO_Check_Speed_Reliability	第 117 页
	STO_Start_Up	第 118 页
	STO_Get_wlalfa_est	第 118 页
	STO_Get_wlbeta_est	第 118 页
	STO_Get_wBemf_alfa_est	第 118 页
	STO_Get_wBemf_beta_est	第 119 页
	STO_Gains_Init	第 119 页
	STO_Gains_Update	第 119 页

#### STO\_Init

梗概 **void STO\_Init(void)**

描述 它将所有有关状态检测的变量初始化为合适的值。每次电机启动前都调用一次。

输入 无

返回值 无

提示 示例程序中，在 INIT 状态期间被调用。

#### STO\_StateObserverInterface\_Init

梗概 **void STO\_StateObserverInterface\_Init(void)**

描述 这个函数根据电机参数，默认状态检测增益向量和 PLL 增益初始化无传感算法；在 MC\_State\_Observer\_param.h 和 MC\_PMSM\_motor\_param.h 头文件中重新获取数据

输入 无

返回值 无

提示 在运行期间，任何时候都可用使用 STO\_Obs\_Gains\_Update 函数覆盖初始化设置，修改检测和 PLL 增益。

#### STO\_Obs\_Gains\_Update

梗概 **void STO\_Obs\_Gains\_Update(void)**

描述 这个函数的目的是修改 STO\_StateObserverInterface\_Init 预先设置的状态检测和 PLL 的增益。

输入 无

返回值 无

提示 在示例程序中，取消 stm32f10xMCconf.h 中 OBSERVER\_GAIN\_TUNING 的注释可以调整无传感算法（通过 STO\_Obs\_Gains\_

Update)。

#### STO\_Calc\_Rotor\_Angle

**梗概** void STO\_Calc\_Rotor\_Angle(Volt\_ComponentsStat\_Volt\_alfa\_beta,  
**描述** 这是该模块的核心，实施检测状态；此函数调用频率与定子电流采样频率相同（在演示例程中，因为周期刚好和 FOC 执行速率一致，正如在 4.2 节讨论过的一样，它由 FOC 程序从内部调用）。  
它获取测量的定子电流（Stat\_Curr\_alfa\_beta），施加的电压命令（Stat\_Volt\_alfa\_beta）和测量的直流母线电压(hBusVoltage)当作第 K 步的输入；得出离散状态检测方程的第 K+1 步，从而实现电机反电动式的计算（ $e_\alpha$  的和  $e_\beta$ ）。从而通过一个数字锁相环的方法，根据反电动势计算转子速度和角度。  
检测到的反电动势、转子角度和速度被写入单元私有变量中。  
**输入** 定子电压指令  $v_\alpha$  和  $v_\beta$  (S16 格式)，测量的定子电流  $i_\alpha$  和  $i_\beta$  (S16 格式)，直流母线电压 (S16 格式)。见 MC\_type.h 结构声明。  
**提示** 参见第 2.2 节获得更多关于无传感器算法的信息。

#### STO\_Calc\_Speed

**梗概** void STO\_Calc\_Speed(void)  
**描述** 在速度闭环控制中必须调用这个函数（示例程序中由 PID\_SPEED\_SAMPLING\_TIME 参数固定的）  
函数具有两个作用：  
| 它计算缓存的测量速度平均值，将结果存储到单元私有变量中；  
| 它可以计算的缓冲区速度的总体方差：如果方差比 VARIANCE\_THRESHOLD 设定的阈值高（见第 4.5.3 节），计算的速度就会声明为“不可靠”，并且设定模块的私有标志。

**输入** 无

**返回值** 无

**提示** 无

#### STO\_InitSpeedBuffer

**梗概** void STO\_InitSpeedBuffer(void)  
**描述** 这个函数初始化缓冲区域，用于 STO\_Calc\_Speed 存储观测到的转子速度。在每次电机启动之前调用。

**输入** 无

**返回** 无

**提示** 在示例程序中，在 WAIT 和 FLASE 状态下调用。

#### STO\_Get\_Electrical\_Angle

**梗概** s16 STO\_Get\_Electrical\_Angle(void)  
**描述** 在第 K+1 步返回转子电角度，作为模块私有变量在第 K 步计算和存储的 STO\_Calc\_Rotor\_Angle 值。

**输入** 无

**返回值** 检测的转子电角度 (s16 格式)

**提示** 无

#### STO\_Get\_Mechanical\_Angle

**梗概** s16 STO\_Get\_Mechanical\_Angle(void)

**描述** 在第 K+1 步转子返回机械角度

输入 无

返回值 检测的转子机械角度（s16 格式）

提示 这个函数依赖 **STO\_Get\_Electrical\_Angle**。

**STO\_Get\_Speed**

梗概 **s16 STO\_Get\_Speed(void)**

描述 返回转子电速度，作为平均算出的缓存里的检测速度累积的 **STO\_Calc\_Speed** 值。

输入 无

返回值 检测的转子电速度（单位脉冲格式）

提示 另见 A.7 节关于速度格式

**STO\_Get\_Speed\_Hz**

梗概 **s16 STO\_Get\_Speed\_Hz(void)**

描述 返回转子的机械速度

输入 无

返回值 检测的转子机械速度（Hz\*10）

提示 此函数依赖 **STO\_Get\_Speed**。

**STO\_IsSpeed\_Reliable**

梗概 **bool STO\_IsSpeed\_Reliable(void)**

描述 这个程序会指出无传感算法提供的信息是否可靠的。为了达到此目的，函数通过 **STO\_Calc\_Speed** 检查模块的私有变量。**FALSE** 状态表明转子位置重建由于例如，检测或者 PLL 增益选择错误，或者转子被卡死而出现故障。

输入 无

返回值 布尔值，检测提供的数据可靠返回 **TRUE**

提示 无

**STO\_Check\_Speed\_Reliability**

梗概 **bool STO\_Check\_Speed\_Reliability(void)**

描述 这个程序会指出无传感算法提供的信息是否持续可靠。应与采样的频率相同的频率调用。调用 **STO\_IsSpeed\_Reliable**：如果函数返回 **RELIABILITY\_HYSTERESYS** (**MC\_State\_Observer\_param.h**) 次 **FLASE**，则转子速度/位置检测不可靠。

输入 无

返回值 布尔值，检测数据可靠返回 **TURE**

提示 这个函数依赖 **STO\_IsSpeed\_Reliable**

**STO\_Start\_Up**

梗概 **void STO\_Start\_Up(void)**

描述 这个函数是一个启动程序，实现电机由静止到旋转的启动；必须以转子电流采样频率相同的频率他调用。

因此，根据 **MC\_State\_Observer\_param.h** 中设置参数（见第 4.5.2 节），三相对称电流会产生一个旋转定子磁场，引起转子跟随转动。在这些操作中，会调用 **STO\_Calc\_Rotor\_Angle** 函数：如果观测的可靠度在 **MC\_State\_Observer\_param.h**（第 4.5.3 节）固定的范围内，主要机状态是允许切换至运行状态。

输入 无

返回值 无

提示 无

STO\_Get\_wlalfa\_est

梗概 `s16 STO_Get_wlalfa_est(void)`

描述 执行 `STO_Calc_Rotor_Angle` 第  $K$  步时这个函数返回计算的测量电流  $i$ 。

输入 无

返回值 检测电流  $i_\alpha$  (`s16` 格式)

提示 在示例中，此功能仅用于显示通过 `DAC` 的功能获得的变量（如果在 `stm32f10xMCConf.h` 启用）。

STO\_Get\_wlbeta\_est

梗概 `s16 STO_Get_wlalfa_est(void)`

描述 执行 `STO_Calc_Rotor_Angle` 第  $K$  步时这个函数返回计算的测量电流  $i$ 。

输入 无

返回值 检测电流  $i_\beta$  (`s16` 格式)

提示 在示例中，此功能仅用于显示通过 `DAC` 的功能获得的变量（如果在 `stm32f10xMCConf.h` 启用）。

STO\_Get\_wBemf\_alfa\_est

梗概 `s16 STO_Get_wBemf_alfa_est(void)`

描述 执行 `STO_Calc_Rotor_Angle` 第  $K$  步时这个函数返回计算的反电动势  $e_\alpha$ 。

输入 无

返回值 检测到的反电动势  $e_\alpha$  (`s16` 格式)

提示 在示例中，此功能仅用于显示通过 `DAC` 的功能获得的变量（如果在 `stm32f10xMCConf.h` 启用）。

STO\_Gains\_Init

梗概 `void STO_Gains_Init(StateObserver_Const*StateObserver_ConstStruct)`

描述 根据 `MC_State_Observer_param.h` 中定义的默认参数由 `STO_StateObserverInterface_Init` 调用函数初始化无传感器算法。

返回值 无

提示 无

STO\_Gains\_Update

梗概 `void STO_Gains_Update(StateObserver_GainsUpdate*  
STO_GainsUpdateStruct)`

梗概 通过 `STO_Obs_Gains_Update` 调用此函数，修改检测状态和 `PLL` 增益。

返回值 无

提示 无

## 5.9 PID 调节器：MC\_PID\_regulators 模块

`MC_PID_regulators` 模块包含所有需要的函数，包含应用要求的 `PID` 调节器实例，以此来控制电流  $I_{qs}$  和  $I_{ds}$ ，电机转速（在速度控制模块中）和定子电压（通过弱磁控制比额定值高的速度）。

注释：取消对库配置文件中 `DIFFERENTIAL_TERM_ENABLED` 的注释，计算不同的值（见 4.1 节）。

通过声明和初始化 `PID_Struct_t` type 结构体静态变量来创建 `PID` 调节器实例(结

构声明见 *MC\_type.h*)

### 5.9.1 可用函数列表

MC\_PID\_regulators 函数列表如下:

- I PID\_Init 在120页
- I PID\_Regulator 在120页
- I PID\_Speed\_Coefficients\_update 在121页

PID\_Init

梗概 void PID\_Init (PID\_Struct\_t \*PID\_Torque, PID\_Struct\_t \*PID\_Flux, PID\_Struct\_t \*PID\_Speed)

描述 函数的目的是初始化PIDs调节电流和速度。对于每一项都会载入默认值: 目标(速度或电流, 比例、积分和微分增益, 输出的高低限)。

输入 PID\_Struct\_t \*, PID\_Struct\_t \*, PID\_Struct\_t \* (结构声明见*MC\_type.h*)

函数调用: 无

提示 声明PID调节器默认值, 并可以在MC\_Control\_Param.h中修改(见46页4.2节)

PID\_Regulator

梗概 s16 PID\_Regulator(s16 hReference, s16 hPresentFeedback, PID\_Struct\_t \*PID\_Struct)

描述 函数的目的是在第K步时计算PID实例调节器输出, 总结比例、积分和微分项(如果取消了对DIFFERENTIAL\_TERM\_ENABLED注释, 函数目的是计算后者, 见4.1节)

输入 hReference (需要的设置点), hPresentFeedback (控制系统的测量输出), PID\_Struct\_t \*PID\_Struct (指向PID\_Struct\_t变量)是调节器自身实例, 保持增益, 内部状态, 内部总和界限和输出界限(结构体声明见*MC\_type.h*)。

输出 控制器输出(有符号的16位)

函数调用 无

提示 演示程序包含几个PID调节器实例; 在MC\_Control\_Param.h文件中可以修改PID调节器电流和速度默认设置(见46页4.2节)。PID\_Regulator函数通过输入的指向PID\_Struct\_t变量的指针更新PID调节器实例内部状态。

PID\_Speed\_Coefficients\_update

梗概 void PID\_Speed\_coefficients\_update(s16 motor\_speed)

描述 函数根据电机实际转速自动计算速度PID调节器的比例、积分和微分增益。基于4个设定点画出一条曲线完成计算。更多信息见123页5.9.4节。

函数调用 无

提示 MC\_Control\_Param.h文件可以声明、修改四个设定点的默认值(见46页4.2节)。

### 5.9.2 PID 调节器理论背景

调节转矩、磁通量和转速的调节器包括比例、积分和微分三部分调节(微分

项见下面注意事项)。PID 调节器理论和调节方法在技术领域有过充分的研究。这节会稍微提到这方面的理论。

PID 调节器通常根据设定的目标保持一定程度的转矩、磁通或者速度。

torque = f(rotor position)  
flux = f(rotor position) } torque and flux regulation for maximum system efficiency

torque = f(rotor speed) } torque regulation for speed regulation of the system

Where:  $Error_{sys_T}$  Error of the system observed at time  $t = T$   
 $Error_{sys_{T-1}}$  Error of the system observed at time  $t = T - T_{sampling}$

$$f(X_T) = K_p \times Error_{sys_T} + K_i \times \sum_0^T Error_{sys_i} + K_d \times (Error_{sys_T} - Error_{sys_{T-1}}) \quad (1)$$

Derivative term can be disabled

图 67 PID 通用方程

方程 1 在 PID 中比较经典，在这里；

- I  $K_p$  比例系数
- I  $K_i$  积分系数
- I  $K_d$  微分系数

注释 正如在67页提及的一样，可以禁用PID的微分项（通过编译选项实现，见文件`stm32f10x_MCconf.h`）。

### 5.9.3 调节器采样时间设定

调节规定带宽需要修改采样时间。由于算法已经采用累积项（积分项），增加循环时间会降低效果（累积会放缓，输出积分会延后）。相反，减小循环时间会增加效果（累积会加速，输出积分会增加）。这就是为什么需要优先调节这个参数设定PID调节器系数。

为了尽可能降低CPU负载，如67图式（1）一样，采样时间直接影响积分系数，因此避免了额外的乘法运算。

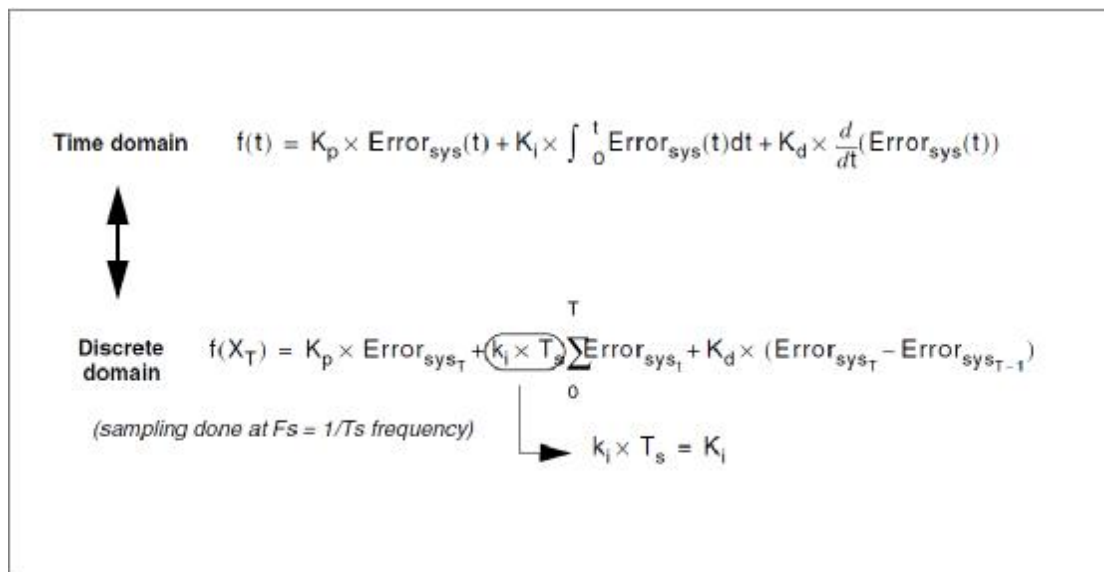


图68 时间域离散化的PID方程

理论上，采样频率越高，效果越好。实际上必须要记住：

- I 相关CPU负载根据实际增大
- I 至于速度调节，没有必要使采样频率比外围传感器反馈速度信息频率还低；这在驱动电机低速旋转同时使用霍尔传感器时会很实用。

正如在46页4.2节讨论的一样，速度循环采样频率可以通过编辑头文件MC\_Control\_Param.h中PID\_SPEED\_SAMPLING\_TIME参数来定义。磁通和转矩PID调节器采样频率由下式计算

$$\text{磁通和转矩PID频率} = \frac{2 \text{ PWM\_FREQ}}{\text{REP\_RATE} + 1}$$

注释 如果由分流电阻测量电流，REP\_RATE必须是奇数，8位长（见136页A.2）。

#### 5.9.4 电机频率与速度定义循环 Ki，Kp，Kd 的调节

根据电机频率可能会需要不同的Ki，Kp，Kd的值。这些值必须写入编码中，再送入循环算法。软件库会提供作为例子的四个

(PID\_Speed\_Coefficient\_update)之间的线性插值函数，只要系数可以线性化，这个方法就可以应用到大多数情况下。如果不可行，就需要插值点数更多，或者需要插值表。

为了写入四个设定的点，一旦采集到数据就立刻编辑文件MC\_Control\_param.h，填写Ki，Kp，Kd系数部分，如下所示。

```
//设置最小频率
#define Freq_Min 10 // 1 Hz 机械频率
#define Ki_Fmin 1000 //频率最小系数设定
#define Kp_Fmin 2000
#define Kd_Fmin 3000
//设定中间频率1
#define F_1 50 // 5 Hz机械频率
#define Ki_F_1 2000 //中间频率1系数设置
```

```

#define Kp_F_1 1000
#define Kd_F_1 2500
//设置中间频率 2
#define F_2 200 // 20 Hz机械频率
#define Ki_F_2 1000 // 中间频率 2 系数设定
#define Kd_F_2 750
#define Kd_F_2 1200
//设定最大频率
#define Freq_Max 500 // 50 Hz 机械频率
#define Ki_Fmax 500 //最大频率设定
#define Kp_Fmax 500
#define Kd_Fmax 500

```

一旦电机开始运行，整体、比例和微分系数根据在 $F_{min}$ 和 $F_1$ ， $F_1$ 和 $F_2$ ， $F_2$  和 $F_{max}$ 之间的线性曲线计算(见图69)。注意 $F_{min}$ 和 $F_1$ ， $F_1$ 和 $F_2$ ， $F_2$  和 $F_{max}$ 是机械频率，分辨率0.1Hz（例如 $F_1=1234$ 意味着  $F_1=123.4\text{Hz}$ ）。

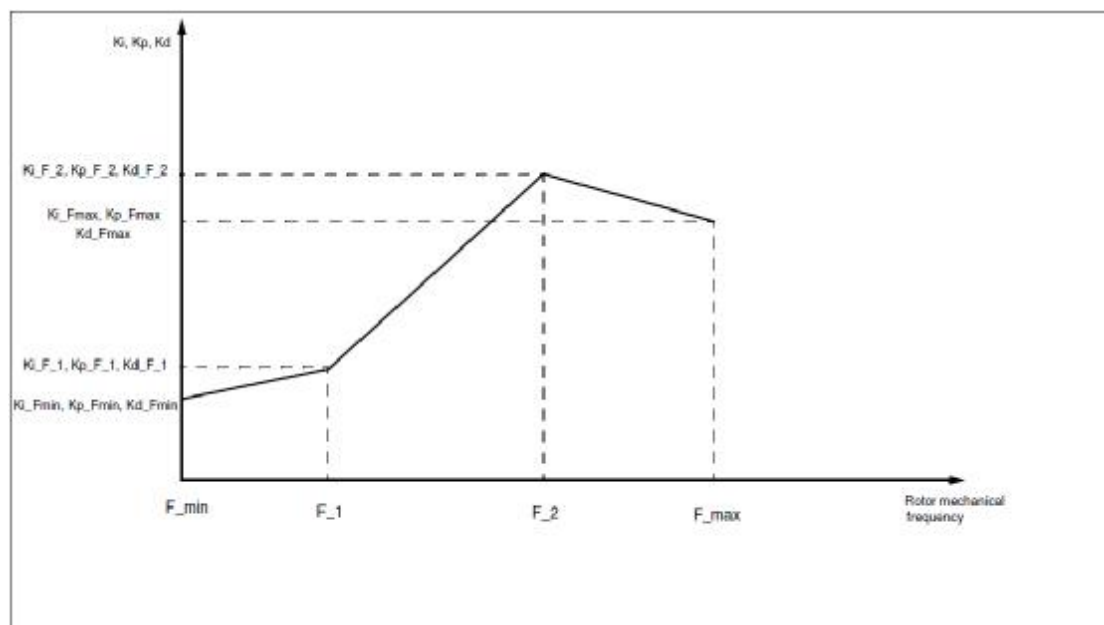


图69 计算系数的线性曲线

禁用线性曲线计算程序，stm32f10x\_Timbase模块

如果想禁用线性曲线计算，必须注释掉PID\_Speed\_Coefficients\_update(..) 程序。在这种情况下，需要使用转矩、磁通和速度调节系数 $K_i$ ， $K_p$ ， $K_d$ 默认值。见MC\_control\_Param.h文件中的PID\_TORQUE\_Kx\_DEFAULT，PID\_FLUX\_Kx\_DEFAULT，PID\_SPEED\_Kx\_DEFAULT。

为了禁用stm32f10x\_Timbase.c中的线性曲线计算程序：

```

void SysTickHandler(void)
{
    [...]
    if ((wGlobal_Flags & SPEED_CONTROL) == SPEED_CONTROL)
    {
        if (State == RUN)

```



```

    {
        //PID_Speed_Coefficients_update(XXX_Get_Speed()); //to be
        //commented
        [...]
    }
}
[...]
```

## 5.10 通用目的时间基准：stm32f10x\_Timebase 模块

stm32f10x\_Timebase单元的的目的是产生时间基准，应用到其它模块中去。

### 5.10.1 可用函数列表与中断服务程序

下面是头文件中可用函数列表

- I TB\_Init 126页
- I TB\_Wait 126页
- I TB\_StartUp\_Timeout\_IsElapsed, TB\_Delay\_IsElapsed,  
TB\_DisplayDelay\_IsElapsed, TB\_DebounceDelay\_IsElapsed 127页
- I TB\_Set\_Delay\_500us, TB\_Set\_DisplayDelay\_500us,  
TB\_Set\_StartUp\_Timeout, TB\_Set\_DebounceDelay\_500us 126页
- I SysTickHandler 127页

#### TB\_Init

梗概 void TB\_Init(void)

描述 函数的功能是初始化**STM32**标记时钟，每**500μs**产生一次中断，因此可以作为通用时间基准。

输入 无

返回 无

函数调用 SysTick\_CLKSourceConfig, SysTick\_SetReload,  
SysTick\_CounterCmd, NVIC\_SystemHandlerPriorityConfig,  
SysTick\_ITConfig

#### TB\_Wait

梗概 void TB\_Wait(u16 time)

描述 函数产生一个可编程的延时程序，延时时间的‘time’变量乘上**500μs**。

输入 无符号**16**位

返回 无

函数调用 无

提示 只有在延时程序执行之后这个程序才会存在。同时，代码的执行仍被循环冻结着。需要注意当在主程序或服务程序调用时：中断程序的调用比基准中断级别高，会冻结代码的运行。

TB\_Set\_Delay\_500us, TB\_Set\_DisplayDelay\_500us,

TB\_Set\_StartUp\_Timeout, TB\_Set\_DebounceDelay\_500us

梗概 void TB\_Set\_Delay\_500us(u16)

```

void TB_Set_DisplayDelay_500us(u16)
void TB_Set_StartUp_Timeout(u16)
void TB_Set_DebounceDelay_500us

```

**描述** 上述函数分别更新hTimebase\_500us, hTimebase\_display\_500us, hStart\_Up\_TimeBase\_500us and hKey\_debounce\_500us变量值。维持主状态在Fault状态，设置LCD刷新频率和启动超时，过滤用户密钥的跳跃。

**输入** 无符号16位

**返回** 无

**调用** 无

TB\_StartUp\_Timeout\_IsEapsed, TB\_Delay\_IsEapsed, TB\_DisplayDelay\_IsEapsed, TB\_DebounceDelay\_IsEapsed

SysTickHandler

**梗概** bool TB\_StartUp\_Timeout\_IsEapsed(void)  
bool TB\_Delay\_IsEapsed(void)  
bool TB\_DisplayDelay\_IsEapsed(void)  
bool TB\_DebounceDelay\_IsEapsed(void)

**描述** 如果延迟程序执行完了则返回TRUE，否则位FALSE

**输入** 无

**返回** 布尔值

**函数调用** 无

**SysTickHandler**

**梗概** void SysTickHandler(void)

**描述** 这是系统标记时钟中断程序。每500μs执行一次，正如由TB\_Init确定一样，用来刷新主要作为计数器的各种变量（例如，PID采样时间）。而且，如果在stm32f10xMCCConf中取消对FLUX\_TORQUE\_PIDs\_TUNING的注释，就会控制电流分流参考值产生定义周期的方波（见4.1节和附录A.5）。

**输入** 无

**返回** 无

**函数调用** 在速度控制模式下：  
FOC\_CalcFluxTorqueRef, (PID\_Speed\_Coefficients\_update)  
在转矩控制模式下：  
FOC\_TorqueCtrl  
如果使用了编码器：  
ENC\_Calc\_Average\_Speed (if using DAC, ENC\_Get\_Mechanical\_Speed)  
如果使用了霍尔传感器：  
(if using DAC, HALL\_GetSpeed)  
如果使用了霍尔传感器：  
STO\_Calc\_Speed, STO\_Check\_Speed\_Reliability, MCL\_SetFault, STO\_obs\_Gains\_Update (if using DAC, STO\_Get\_Speed)

**提示** 这是一个中断服务程序

## 5.11 功率级核查: MC\_MotorControl\_Layer 模块

### 5.11.1 函数列表

MC\_MotorControl\_Layer.h头文件函数列表如下:

- | MCL\_Init 128页
- | MCL\_ChkPowerStage 129页
- | MCL\_ClearFault 129页
- | MCL\_SetFault 129页
- | MCL\_Chk\_OverTemp 130页
- | MCL\_Chk\_BusVolt 130页
- | MCL\_Compute\_BusVolt 130页
- | MCL\_Compute\_Temp 130页
- | MCL\_Calc\_BusVolt 131页
- | MCL\_Calc\_BusVolt 131页
- | MCL\_Init\_Arrays 131页
- | MCL\_Brake\_Init 131页
- | MCL\_Set\_Brake\_On 131页
- | MCL\_Set\_Brake\_Off 131页

#### MCL\_Init

梗概 void MCL\_Init(void)

描述 这个函数在每次电机启动前初始化电机控制;函数影响PID调节器、电流读取刻度,速度传感器和高边驱动引导电容初始化。

函数调用 ENC\_Clear\_Speed\_Buffer或HALL\_InitHallMeasure和  
HALL\_Init\_Electrical\_Angle或ST0\_Init depending on the speed  
feedback configured, TB\_Set\_StartUp\_Timeout,  
TIM1\_CtrlPWMOutputs, TB\_StartUp\_Timeout\_IsElapsed,  
SVPWM\_3ShuntCurrentReadingCalibration或  
SVPWM\_IcsCurrentReadingCalibration根据电流反馈配置,  
SVPWM\_3ShuntCalcDutyCycles或SVPWM\_IcsCalcDutyCycles根据电流反馈  
配置

输入 无

返回 无

#### MCL\_ChkPowerStage

梗概 void MCL\_ChkPowerStage(void)

描述 函数检测功率等级工作条件(只针对温度和总线电压),并在需要的时候产生FAULT事件。

函数调用 MCL\_Chk\_OverTemp, MCL\_Chk\_BusVolt, MCL\_SetFault

输入 无

返回 无

#### MCL\_ClearFault

梗概 bool MCL\_ClearFault(void)

描述 函数检测产生错误事件是否已经结束。在正向,如果已经按下‘Key’键,

就会清除相关标志，返回TRUE值。否则返回FALSE值。

函数调用 TB\_Delay\_IsElapsed, MCL\_Chk\_BusVolt, MCL\_Chk\_OverTemp,  
GPIO\_ReadInputDataBit, KEYS\_ExportbKey

输入 无

返回 如果所有的标志都已经清除并且用户已经按下‘Key’键，就会返回TRUE值，否则返回FALSE值。

也可见 3.8节：FALSE信息。

MCL\_SetFault

梗概 void MCL\_SetFault (u16)

描述 发生错误事件，函数就会将电机主状态切换到Fault状态，并禁用  
Advanced Control Timer TIM1 (PWM timer)的电机控制输出。

函数调用 TB\_Set\_Delay\_500us, TIM1\_CtrlPWMOutputs,  
SVPWM\_3ShuntAdvCurrentReading 在三相分流电阻读取配置情况下

输入 在文件中MC\_const.h定义错误事件源头。

返回 无

也可以见 3.8节：Fault信息。

MCL\_Chk\_OverTemp

梗概 bool MCL\_Chk\_OverTemp(void)

描述 函数通过下式计算获取值最新温度的平均值：

$$XAV(K) = (XAV(K-1) * (T\_AV\_ARRAY\_SIZE - 1) + X(K)) / T\_AV\_ARRAY\_SIZE$$

XAV(K)是第K步的平均值，X (K)是第K步的最新测量值。

一旦计算出平均值，函数会检测获得的温度是否在允许范围内。介入阈值和滞后值可以在MC\_Control\_Param.h文件中调节（只针对MB459板）。

输入 无

返回 如果连接到ADC的通道ADC\_IN10上的热电阻的软件平均电压已经达到阈值（或在过热检测后还没返回到阈值减去滞后值）返回TRUE值。否则返回FALSE值。

MCL\_Chk\_BusVolt

梗概 BusV\_t MCL\_Chk\_BusVolt(void)

描述 这个函数检查逆变器直流端电压是否过压或欠压。插入阈值可以在文件MC\_Control\_Param.h中定义（只针对MB459板）。

函数调用 无

输入 无

返回 返回显示错误值的BusV\_t type

MCL\_Compute\_BusVolt

梗概 u16 MCL\_Compute\_BusVolt(void)

描述 这个函数以V为单位计算直流总线电压。在设定的固件库中这个函数与用户交互。

输入 无

返回 总线电压（V）

MCL\_Compute\_Temp

梗概 u8 MCL\_Compute\_Temp(void)

描述 这个函数计算功率级温度散热槽温度（摄氏度，只针对MB459板）。在

设定的固件库中这个函数与用户交互。

输入 无

返回 返回代表温度的摄氏温度整数。

**MCL\_Calc\_BusVolt**

梗概 void MCL\_Calc\_BusVolt(void)

描述 函数计算由下式得出的BUS\_AV\_ARRAY\_SIZE最新温度的平均值：

$$X_{AV} = (X_{AV}(K-1) * (BUS\_AV\_ARRAY\_SIZE-1) + X(K)) / BUS\_AV\_ARRAY\_SIZE,$$

这里 $X_{AV}(K)$ 是第K步平均值， $X(K)$ 是第K步最新测量的值。

输入 无

返回 无，平均值写入单元私有变量中。

**MCL\_Calc\_BusVolt**

梗概 s16 MCL\_Get\_BusVolt(void)

描述 这个函数只输出总线电压私有变量平均值。

输入 无

返回 数字量总线电压。

**MCL\_Init\_Arrays**

梗概 void MCL\_Init\_Arrays(void)

描述 函数初始化电压和温度平均值。在重置MCU后调用。

输入 无

返回 无

**MCL\_Brake\_Init**

梗概 void MCL\_Brake\_Init(void)

描述 只有在stm32\_MCconf.h中启用制动电阻后才定义和声明这个函数，并将驱动阻力制动的开关的GPIO引脚初始化（在MC\_MotorControl\_Layer.c文件中定义 BRAKE\_GPIO\_PORT, BRAKE\_GPIO\_PIN）。在重置MCU后调用。

输入 无

返回 无

也可见 使用制动电阻时如何设定系统见3.14节。

**MCL\_Set\_Brake\_On**

梗概 void MCL\_Set\_Brake\_On(void)

描述 只有在stm32\_MCconf.h中启用制动电阻后才定义和声明这个函数，函数通过设置BRAKE\_GPIO\_PORT端口的BRAKE\_GPIO\_PIN引脚来启用制动电阻。每次模拟看门狗程序触发时这个函数都会在ADC\_IRQHandler中调用。

输入 无

返回 无

也可见 使用电阻制动时如何设置系统见3.14节。

**MCL\_Set\_Brake\_Off**

梗概 void MCL\_Set\_Brake\_Off(void)

描述 只有在stm32\_MCconf.h中启用制动电阻后才定义和声明这个函数，函数通过设置BRAKE\_GPIO\_PORT端口的BRAKE\_GPIO\_PIN引脚为0禁用制动电阻。只有制动电阻已经启动并且总线电压低于BRAKE\_HYSTERESIS中规定的阈值时才在ADC\_IRQHandler中调用这个函数。

提示 在stm32f10x\_it.c文件中定义BRAKE\_HYSTERESIS，它的默认值是过压插入阈值的15/16。

输入 无  
返回 无  
也可见 使用电阻制动时如何设置系统见3.14节。

## 5.12 主中断服务程序: stm32f10x\_it module

stm32f10x\_it模块可以用来描述在应用中发生的意外情况。发生中断时，软件会自动根据中断向量列表启用对应的程序。使用中断请求例外，所有程序都是空的，所以你可以写入自己的意外处理和外围中断程序。

### 5.12.1 非空中断服务程序列表

上面提及的只有三个中断服务受电机控制：

- I TIM1\_BRK\_IRQHandler 133页
- I TIM1\_UP\_IRQHandler 133页
- I ADC1\_2\_IRQHandler 134

TIM1\_BRK\_IRQHandler

梗概 void TIM1\_BRK\_IRQHandler(void)

描述 这个函数的目的是在专用EAK引脚上引发中断事件。实际上，这时已经禁止TIM1输出，主电机状态进入FAULT状态。

输入 无

返回 无

函数调用 MCL\_SetFault, TIM1\_ClearPendingBit

也看见 在STM32F103xx参考手册中的提前控制时钟(TIM1)

TIM1\_UP\_IRQHandler

梗概 void TIM1\_UP\_IRQHandler(void)

描述 中断处理在TIM1计数器发生下溢时的更新事件之后进行。在处理过程中，与选择的电流采样方法（单相分流，三相分流或ICS）相关的SVPWMUpdateEven程序被调用。

输入 无

返回 无

函数调用 ADC\_ClearFlag, TIM1\_ClearFlag, SVPWMUpdateEvent

ADC1\_2\_IRQHandler

梗概 void ADC1\_2\_IRQHandler(void)

描述 函数的目的是处理ADC总体中断请求。处理两个不同程度的中断源：JEOC（转换插入末尾），AWD（模拟看门狗）。

JEOC: 如果主状态是启动，会触发电机启动程序（具体根据系统配置，见4.1节）；要不然，如果主状态是运行，会触发FOC算法执行。如果系统配置包括制动电阻（见3.14节），就会启用滞后开关（以防过电压）。如果启用了DAC功能，就会更新相关变量值。

AWD: 在过压情况下，根据是否在stm32f10x\_MCconf.h中注释BRAKE\_RESISTOR，启用制动电阻或产生错误事件(OVER\_VOLTAGE)，见4.1节和3.14节。在处理过程中，与电流采样方法（单相分流，三相分流或者ICS）相关的SVPWMEOCEvent程序会被调用。如果电流采样已经完成一个PWM周期这个

程序就会返回真值，此时也可以执行相关FOC程序。

输入 无

返回 无

函数调用 SVPWME0CEvent, FOC\_Model, ADC\_GetITStatus,  
ADC\_ClearFlag, MCL\_Calc\_BusVolt, MCL\_SetFault;  
如果使用了制动电阻:  
ADC\_GetInjectedConversionValue, MCL\_Set\_Brake\_On,  
MCL\_Set\_Brake\_Off;  
如果启用了DAC功能:  
MCDAC\_Update\_Value, MCDAC\_Update\_Output;  
如果使用了编码器:  
ENC\_Start\_Up (并且如果使用了DAC, ENC\_Get\_Electrical\_Angle);  
如果使用了霍尔传感器:  
(如果使用DAC, HALL\_IncElectricalAngle,  
HALL\_GetElectricalAngle);  
如果使用了无传感器算法:  
STO\_Start\_Up, STO\_Calc\_Rotor\_Angle, MCL\_Get\_BusVolt (并且如果  
使用了DAC, STO\_Get\_Electrical\_Angle, STO\_Get\_wlalfa\_est,  
STO\_Get\_wlbeta\_est, STO\_Get\_wBemf\_alfa\_est,  
STO\_Get\_wBemf\_beta\_est)

也可见 详情见5.1节和5.3。

## 附录 A 附加信息

### A.1 调节运行 FOC 算法相关负载

前行定时器外设在给定PWM半周期数后具有内在更新PWM寄存器功能。这个功能由可编程重复计数器完成。对调节给定PWM频率下的FOC算法相关的CPU负载非常有用（更多关于可编程重复计数器详情见STM32F103xx参考手册）。

在使用ICS时，读取电流的转换插入链直接由PWM寄存器更新事件触发。而且，由于FOC算法是在ISR转换末尾才会运行，改变重复计算器会直接影响FOC刷新频率，因此也会影响CPU使用率。

然而，在单相或三相分流扩扑电流读取时，为了确保每次PWM寄存器更新都会执行一次FOC算法，必须保持电流转换触发和PWM寄存器更新同时进行。在设定的软件库中，这会自动执行，所以只需改变重复计数器默认值（在MC\_Control\_Param.h头文件中REP\_RATE参数）就可以减小FOC算法运行频率。图70显示了当REP\_RATE设置为3时电流采样触发，以及与PWM周期相关的FOC算法运行。

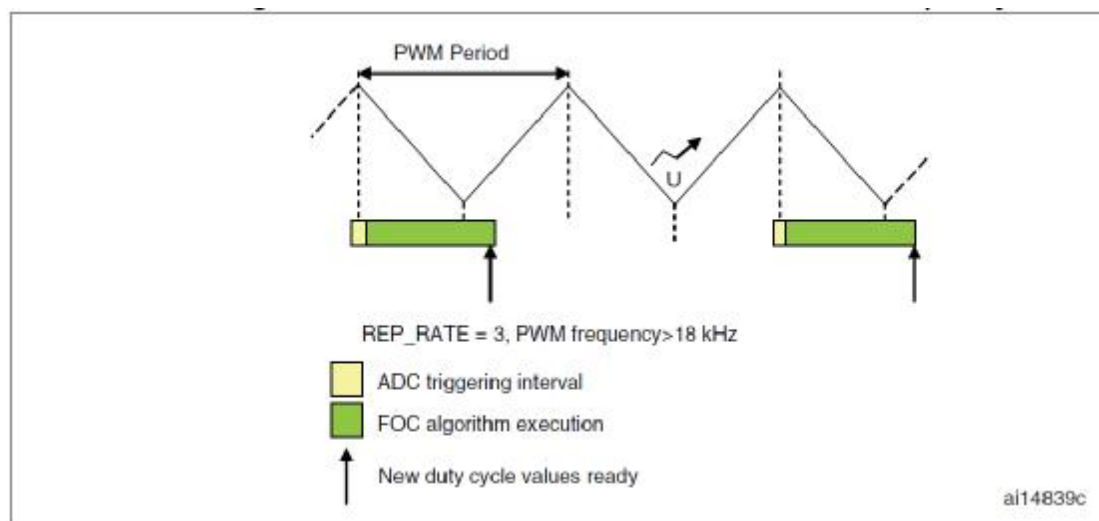


图70 当REP\_RATE=3和PWM频率>18kHz时三相分流扩扑定子电流读取和FOC算法的运行的A/D转换

提示 因为在电流读取A/D转换时要求三相分流电阻扩扑低波开关打开，所以REP\_RATE参数必须是奇数。

考虑到在无传感器和三相分流电阻扩扑电流读取配置中FOC算法运行时间大约是20 μs，相关CPU负载可根据下式计算：

$$CPU\text{负载}_{\%} = \frac{F_{PWM}}{Re\ fresh\_Rate} 20\ 10^{-6}\ 100 = \frac{F_{PWM}}{(REP\_Rate+1)/2} 20\ 10^{-6}\ 100$$



## A.2 基于单相或三相分流电阻配置的 PWM 频率选择更新重复频率

除了要了解噪声和功率耗散之间的平衡之外，还要考虑使用PMSM FOC软件库来选择PWM开关频率。正如69页5.14节和82页5.24节讨论的一样，根据PWM开关频率，如果使用单相或三相分流电阻配置读取电流，可能会限制最大应用占空比。表2：73页三相分流扩扑结构PWM频率与最大占空比之间的关系和表4：82页单相分流扩扑结构PWM频率与最大占空比之间的关系，总结了软件库和硬件一起使用使用时系统的性能。

注释 MB459板是一个评估平台；用来支持不同的电机驱动技术(PMSM 和AC感应)和电流读取策略（单相和三相分流电阻）。因此，73页表2和82页表4给出的数据应该作为启动点来理解而不是最好的数据。而且，为了保证算法恰当的工作，确保新计算占空比在下一次PWM周期应用，必须在图71下一次PWM 事件开始之前结束FOC算法的执行。

在三相分流电阻配置中，考虑5.1.4节情况，在T1M1溢出之后ADC转换触发要晚  $(DT+TN-TS)/2$ 。考虑A/D转换需要的时间，可以说FOC算法是在T1M1计数器溢出之后 $5\mu s$ 才开始执行（最坏的情况）。而且，知道了FOC算法执行时间大约 $20\mu s$ 之后，在无传感器配置中，为了在下个更新事件之前完成新占空比值，必须保证最小占空比大约  $(5+20) \times \mu s$ ，即能达到的最大FOC执行频率大约是20kHz。此时重复计数器(REP\_RATE)可以设定为1。对于高于20kHz的PWM频率，重复计数器(REP\_RATE=3)必须设定为3。如果PWM频率高于17.5kHz，计算合适的 *MC\_Control\_Param.h* MMI 值见5.53节。

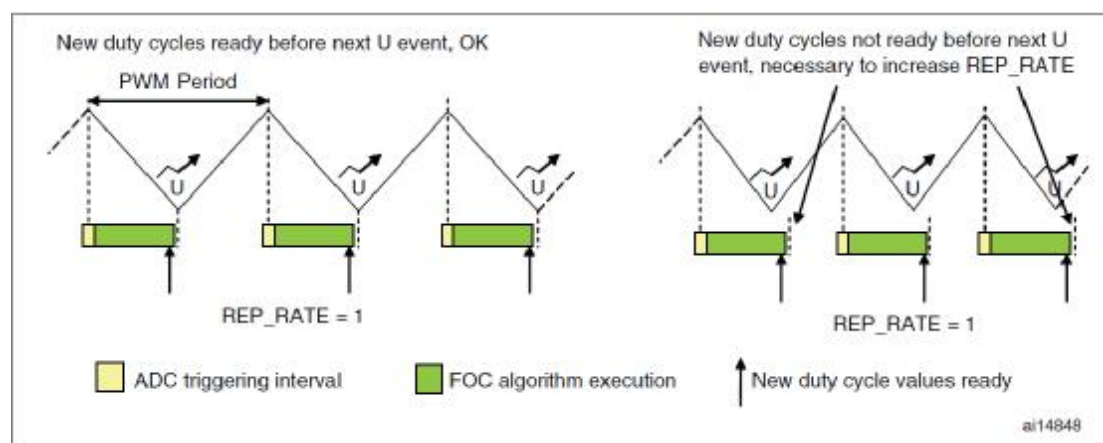


图71 三相分流扩扑定子电流读取和FOC算法执行AD转换

对于单相电流读取，见82页表4：单相扩扑中最小重复频率PWM频率与最大占空比关系（A.8最大调制指数，MMI）：自动计算解释了对于给定的PWM频率和噪声参数如何计算MMI（最大调制指数）。

## A.3 固定点数字代表

PMSM FOC软件库使用固定点代表有符号的分数值。因此，数字 $n$ 是 $n=m*f$ ， $m$ 是整数部分， $f$ 是分数部分，而且 $m$ 和 $f$ 都有固定的数字。就两个补二进制代表而

已，如果变量n需要QI 比特，代表整数部分（其中有一位需要代表符号），QF比特代表小数部分，则需要QI+QF比特来代表那个变量。因此，给定QI和QF值，变量代表有以下特征：

- I 范围:  $-2(QI-1) < n < 2(QI-1) - 2(-QF)$ ;
- I 分辨率:  $= 1 / 2^{QF}$ 。

下面的式子将浮点数量q转化为固定点量代表n:

$$n = \text{floor}(q \cdot 2^{QF})$$

通常解释这个选择就是“qQI.QF”注释。所有，如果一个变量以q3.5格式存储，就意味着3比特是幅度预留空间，5比特是为分辨率预留空间；代表的范围从-4到3.96875，分辨率是0.03125，比特权重是：

bit n.	7	6	5	4	3	2	1	0
value	-4	2	1	1/2	1/4	1/8	1/16	1/32

软件库使用PU（（“Per Unit”）系统来表达电流值。经常指一个基本数字量即最大测量电流 $I_{\max}$ ）（对于设定的硬件，大约为 $I_{\max}=0.6/R_{\text{shunt}}$ ）；所以，“每单位”电流值可以根据下式获得：

$$i_{PU} = \frac{i_{S,I}}{I_{\max}}$$

这样， $i_{PU}$ 范围从-1到1. 因此q1.15格式范围从-1到0.999969482421875，分辨率为0.000030517578125，非常合适，以至于应用很广。因此，完整从SI单位的转换为：

$$i_{q1.15} = \text{floor}\left(\frac{i_{S,I}}{I_{\max}} \cdot 2^{QF}\right)$$

#### A. 4 磁通和转矩电流 PI 增益优先决策

这节附录的目的在于提供计算转矩/磁通PI 参数（ $K_I$ 和 $K_P$ ）的标准。附录A.5讨论了调节她们的方法。

为了计算这些起始值，要求知道电机的电气特性：定子电阻 $R_s$ 和感应 $L_s$ 和硬件电气特性：分流电阻 $R_{\text{shunt}}$ ，电流感应放大倍数 $A_{op}$ 和直流总线电压 $V_{\text{BusDC}}$ 。这个方法不考虑使用控制器微分。

图72显示了调节转矩和磁通的PI 控制器模块

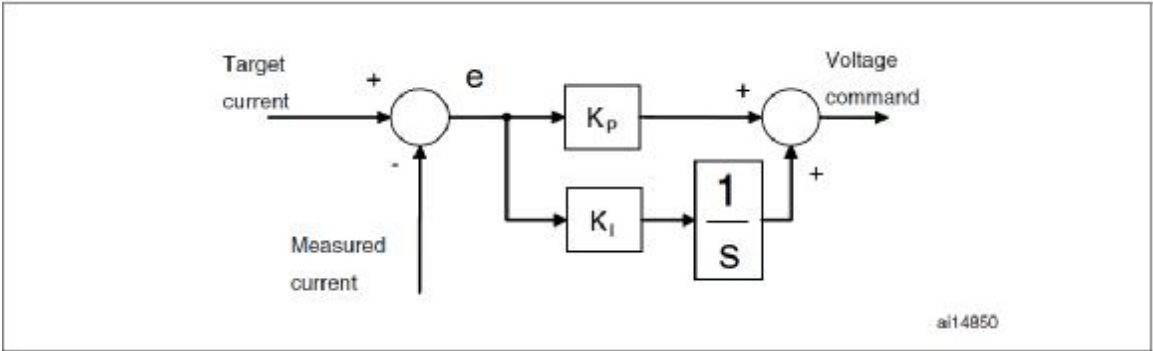


图72 PI控制器模块框图

在这个分析中，电机电气特性假设沿q轴和d轴各向同性。所以，转矩和磁通调节器具有相同起始值 $K_p$ ， $K_i$ 。

图73是感应阻抗等效回路在闭环调件下模拟电机相位的闭环系统。“A”模块存储电压指令（数字量）的软件变量和应用到电机相位上的实际电压（伏特）这两者的比例常数。同样，“B”模块是实际电流（安培）和存储相电流（数字量）的软件变量之间的比例常数。

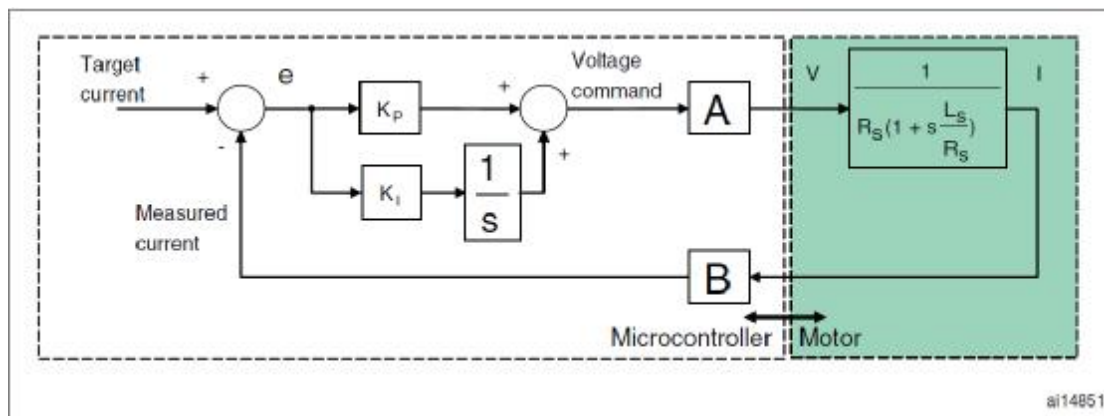


图73 闭环循环结构框图

“A”模块和“B”模块各自的转换分别由下式计算：

$$A = \frac{V_{Bus} DC}{2^{16}} \text{ 和 } B = \frac{R_{shunt} A_{op} 2^{16}}{3.3}$$

输入 $K_p/K_i = L_s/R_s$ ，可以取消图74中的极点-零点分析。

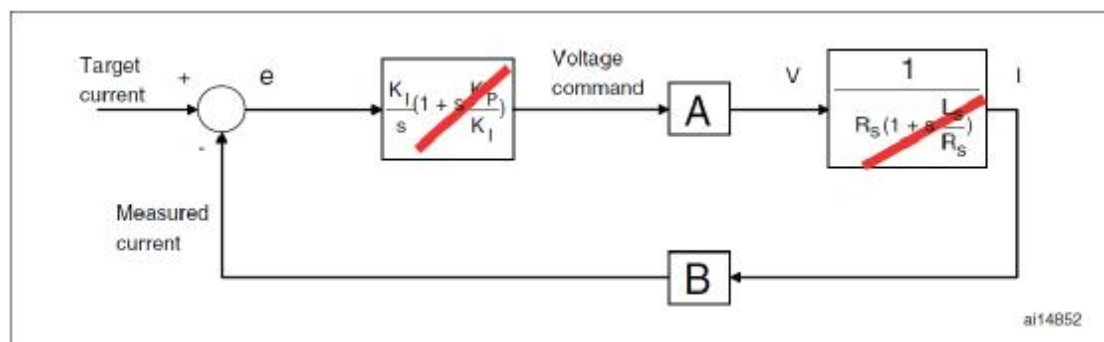


图74 极点-零点分析取消

在这种调件下，闭环系统返回到一阶系统，可以通过恰当的 $K_i$ 值来设定系统的动态特性，见图75。

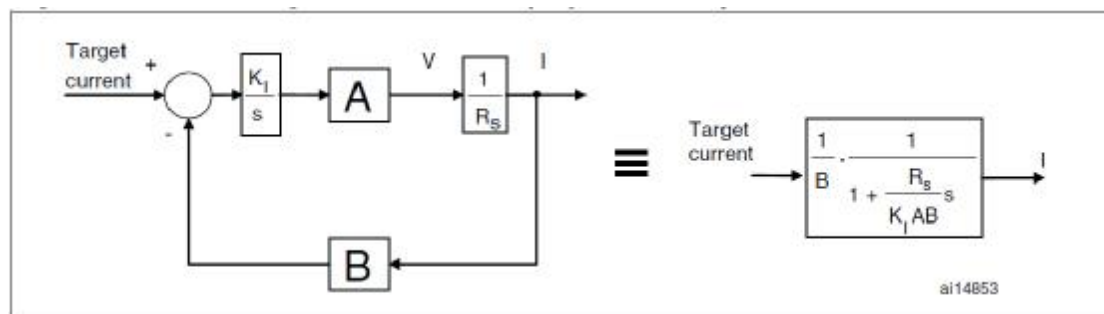


图75 极点-零点分析取消后的闭环系统结构框图

注意到在PI算法中使用的 $K_i$ 和 $K_p$ 参数可以经过恰当的除法器算出。在MC\_Control\_Param.h文件中定义 $K_iDIV$ 和 $K_pDIV$ 了除法器（类似TF\_KP\_DIV, TF\_KI\_DIV, TF\_KDDIV），所以计算出的 $K_i$ 和 $K_p$ 值必须乘上这些因子。而且PI算法不包括积分部分的PI采样时间。见下式：

$$k_i \int_0^t e(t) dt = k_i T \sum_{k=1}^n e(kT) = K_i \sum_{k=1}^n e(kT)$$

由于控制器积分部分计算连续误差之和，在 $K_i$ 计算中要有T参数。所以最后的式子可以表达为：

$$K_p = L_s \frac{w_c}{AB} K_p DIV$$

$$K_i = \frac{R_s w_c K_i DIV}{AB} T$$

$$AB = \frac{V_{Bus} DC R_{shunt} A_{op}}{3.3}$$

通常，可以将 $w_c$ 设定到1500 rad/s，从而获得动态响应和测量噪声敏感度之间良好的平衡。MB459板中测量的 $A_{op}$ 是2.57。然后就可以在知道电机参数( $R_s$ ,  $L_s$ ),  $V_{BUS}DC$ 和 $R_{shunt}$ 的情况下计算参数值

## A. 5 调节电流调节器

为了调节电流调节器，需要从计算参数（ $K_i$ 和 $K_p$ ）开始，接着按照附录A.4中指示做：转矩和磁通电流PI增益优先策略。

然后，从stm32f10x\_MCconf.h默认配置开始，按下列步骤进行：

- I 完成4.2节介绍MC\_Control\_Param.h的“功率装置参数”，“电流调节参数”，“功率板保护阈值”以及“速度循环采样时间”部分。
- I 在stm32f10x\_MCconf.h中，研发设计阶段选择传感器种类，完成4.3节和4.4相关头文件(MC\_encoder\_param.h或MC\_hall\_param.h)。无传感器配置不支持电流调节。
- I 完成MC\_PMSM\_motor\_param.h中的NOMINAL\_CURRENT和POLE\_PAIRS。
- I 取消对FLUX\_TORQUE\_PIDS\_TUNING的注释固件库产生方波形状参考转矩。

提示：当取消对注释时，必须在转矩控制模式下由固件库产生方波。方波周期和幅值可以通过编辑MC\_Control\_Param.h. 中PID\_TORQUE\_REFERENCE和SQUARE\_WAVE\_PERIOD参数设定。目的是实时调节磁通和转矩电流分流PIDs。出于这个目的，用户使用DAC功能实例查看实时的和测量的电流 $I_q$ ，为了在没有超调的情况下快速响应目标 $I_q$ 值，需要对照默认值稍微改变转矩PI(D)增益。相同PI(D)增益就可以在磁通循环中使用了。

图76和77显示了两个截取的示波结果。C1通道是PB0引脚输出，C2 通道是PB1引脚输出。DAC功能用来输出两个内部变量；这时PB0参考电流 $I_q$ ，PB1是测量电流 $I_q$ 。

注意：PB0和PB1是模拟和数字过滤的信号。从图77中可以看出测量电流 $I_q$ 与参考

电流 $I_q$ 对比已经出现超调。在这种情况下，PI 参数设置为 $K_P = 8000$ ， $K_I = 2000$ 。为了减少超调量需要在保持 $K_P$ 恒定时降低 $K_I$ 。这就是图77中如何获得 $K_P = 8000$ ， $K_I = 1000$ 的条件。从图77中可以看出电流调节的响应比图76中的慢。所以增大 $K_P$ 和 $K_I$ ，保持它们比例不变，可以增大系统响应速度。



图76  $K_P = 8000$ ， $K_I = 2000$ 时的情况

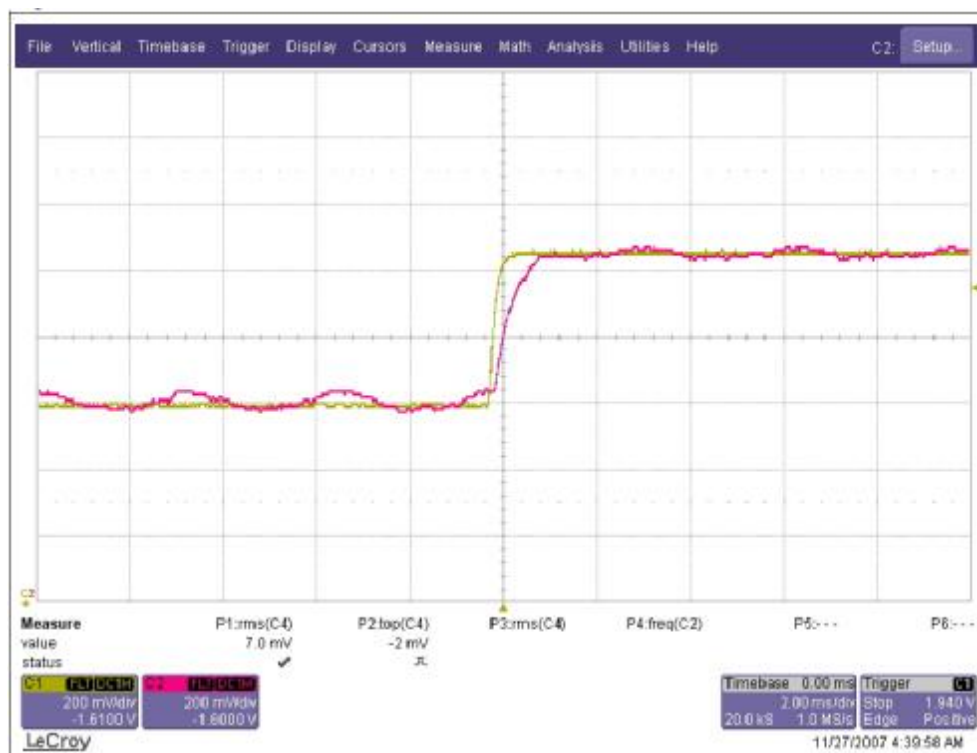


图77  $K_P = 8000$ ， $K_I = 1000$ 时的情况

## A. 6 静态观测增益优先策略

为了加速无传感器开发，用户可以根据附录中介绍的步骤计算静态观测增益，K1和K2。而且，由于3.13节介绍的先进开发系统的应用，现在能达到K1和K2最优调节。

K1和K2的初始值的计算基于静态观测本征值的布置。要求的电机参数是 $r_s$ （电机绕组电阻）， $L_s$ （电机绕组感抗）， $T$ （无传感器算法采样时间，与FOC和定子电流采样一致，见4.2节）。

电机模型本征值可以由下式计算：

$$e_1 = 1 - \frac{r_s T}{L_s}$$
$$e_2 = 1$$

观测本征值由下式计算：

$$e_{1obs} = \frac{e_1}{f}$$
$$e_{2obs} = \frac{e_2}{f}$$

典型的，根据经验法则，用户可以设定 $f=4$ ；

然后，K1和K2的初始值可以根据下式得出：

$$K_1 = \frac{e_{1obs} + e_{2obs} - 2}{T} + \frac{r_s}{L_s}$$
$$K_2 = \frac{L_s(1 - e_{1obs} - e_{2obs} + e_{1obs}e_{2obs})}{T^2}$$

最后，K1和K2就可以用来完成了MC\_State\_Observer\_param.h（见4.5.1节）

## A. 7 速度格式

在PMSM FOC固件库中通常使用两种速度格式：

- I 0.1Hz：这个格式通常由速度调节器及软件最高层（用户接口实例）使用。
- I 每PWM数字（dpp）：dpp格式表达的是速度在一个PWM周期内随电角度（s16格式）的变化。由于每次控制循环执行时（例如PWM更新中断服务程序期间）都可以累积速度信息来计算转子角度位置，这种格式单位就很方便。令 $2n=0xFFFF$ （这样不需要考虑角度翻转），根据下式0.1Hz单位的频率可以很容易转化成dpp格式：

$$w_{dpp} = w_{0.1Hz} \frac{65536}{10 \text{ SAMPLING\_FREQ}}, \text{ 这里:}$$

—— $\text{SAMPLING\_FREQ}$ 是FOC采样频率（在预编译相从REP\_RATE and PWM\_FREQ开始在自动计算）。

## A.8 MMI（最大调制指数）：自动计算

使用提供的工具可以定义对应PWM频率列表的最大调制指数。为此，打开 *MMI vs PWM Freq. exe* 文件（文件位置STM32MC-KIT\design tools\）。然后，设定需要的电流读取方法：单相或三相分流，使用图78给的参数完成要求填写的区域。MMI 然后就会基于噪声参数值（T噪声，T上升）、A/D转换设置以及迟滞时间给出对应所选的PWM频率的最大调制指数。

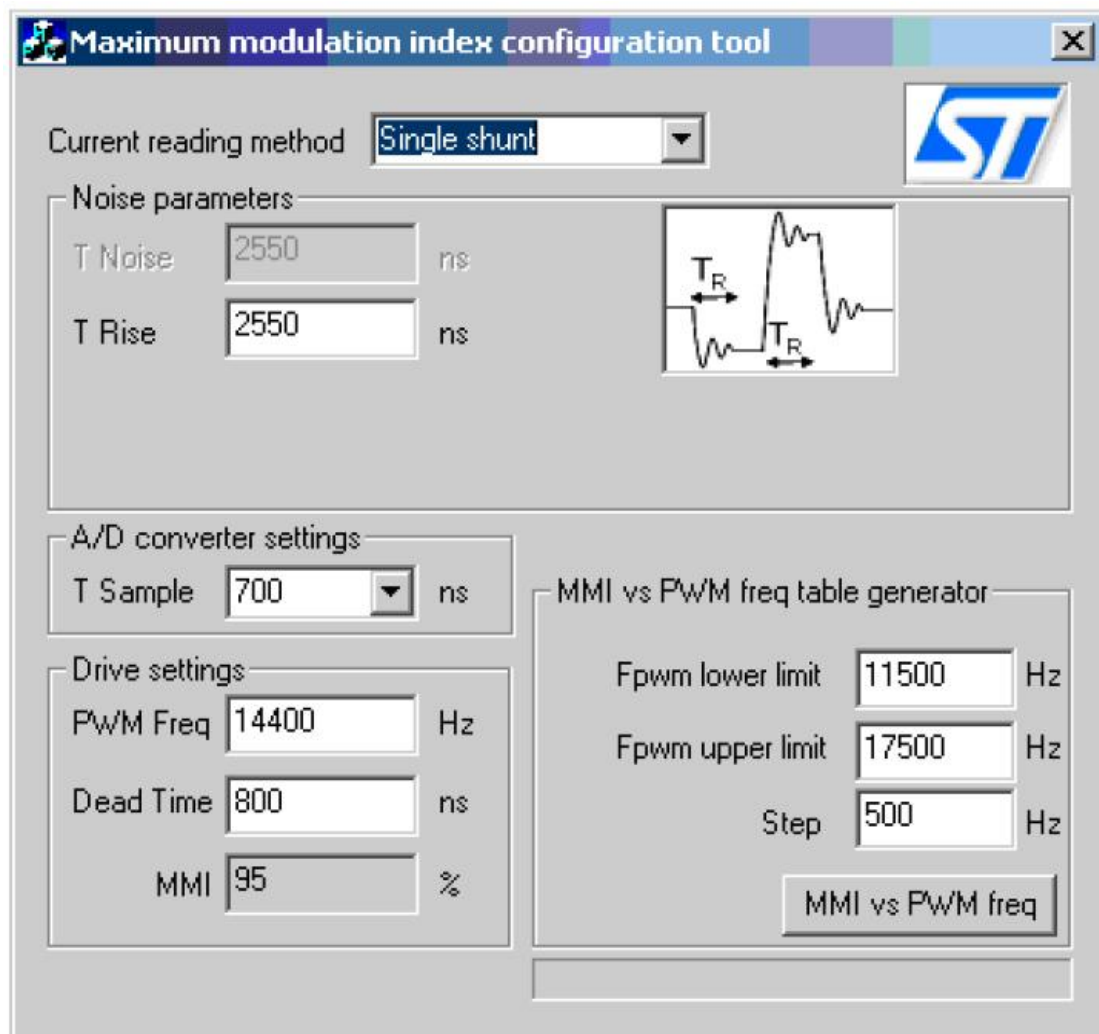


图78 配置工具的最大调制指数

然后就可以产生新MMI与PWM列表了。

为此，在配置工具的最大调制指数窗口中，设置Fpwm lower limit区域的最小PWM频率列表，设置Fpwm upper limit区域的最大PWM频率列表，设置用于计算Step区域列表的频率步进。然后按下MMI vs. PWM freq按钮，图79中的对话框就会出现。



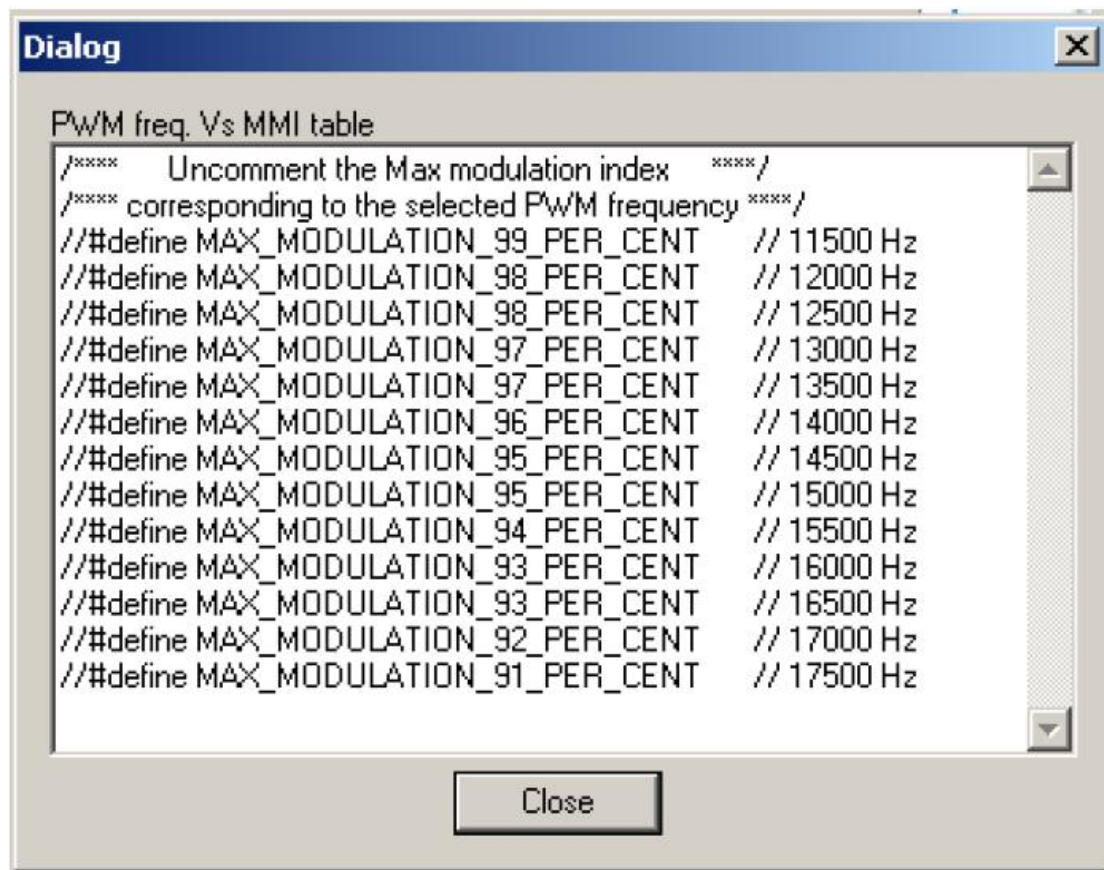


图79 MMI vs. PWM freq定义列表

产生的值可以复制黏贴到`MC_Control_Param.h`文件中。接着必须取消对选择的PWM频率的注释。



## A. 9 参考文献

- [1] P. C. Krause, O. Wasynczuk, S. D. Sudhoff, Analysis of Electric Machinery and Drive Systems, Wiley-IEEE Press, 2002.
- [2] T. A. Lipo and D. W. Novotny, Vector Control and Dynamics of AC Drives, Oxford University Press, 1996.
- [3] S. Morimoto, Y. Takeda, T. Hirasaka, K. Taniguchi, “Expansion of Operating Limits for Permanent Magnet Motor by Optimum Flux-Weakening”, Conference Record of the 1989 IEEE, pp. 51-56 (1989).
- [4] J. Kim, S. Sul, “Speed control of Interior PM Synchronous Motor Drive for the Flux-Weakening Operation”, IEEE Trans. on Industry Applications, 33, pp. 43-48 (1997).
- [5] M. Tursini, A. Scafati, A. Guerriero, R. Petrella, “Extended torque-speed regionsensor-less control of interior permanent magnet synchronous motors”, ACEMP'07, pp. 647 - 652 (2007).
- [6] M. Cacciato, G. Scarcella, G. Scelba, S.M. Billè, D. Costanzo, A. Cucuccio, “Comparison of Low-Cost-Implementation Sensorless Schemes in Vector Controlled Adjustable Speed Drives”, SPEEDAM '08, Applied Power Electronics Conference and Exposition (2008).

## 修订史

表5 文件修订史

Date	Revision	Changes
08-Jan-2008	1	Initial release
21-Aug-2008	2	<p>Permanent-magnet synchronous motor FOC software library updated to revision 2.0 (see PMSM FOC software library V2.0 features (CPU running at 72 MHz) on page 9).</p> <p>Updated sections:</p> <ul style="list-style-type: none"> <li>- Section 1: Getting started with tools</li> <li>- Section 2: Introduction to the sensorless FOC of PM motors</li> <li>- Section 3: Running the demo program</li> <li>- Power device parameters, Speed PID-controller init values, Quadrature current PID-controller init values</li> <li>- Section 4.5: State observer parameters: MC_State_Observer_param.h</li> <li>- Section 4.6: Permanent-magnet synchronous motor parameters: MC_PMSM_motor_param.h</li> <li>- Section 5.2: Current reading in single shunt resistor topology and space vector PWM generation: stm32f10x_svpwm_1shunt module</li> <li>- Section 5.4: PMSM (SM-PMSM / IPMSM) field-oriented control: MC_FOC_Drive and MC_FOC_Methods modules</li> <li>- ENC_Start_Up on page 105</li> <li>- Section 5.9.1: List of available functions on page 120</li> <li>- SysTickHandler on page 127</li> <li>- MCL_Init on page 128</li> <li>- Section 5.12: Main interrupt service routines: stm32f10x_it module</li> <li>- A.2: Selecting the update repetition rate based on the PWM frequency for single- or three-shunt</li> </ul>

		<p>resistor configuration on page 136</p> <ul style="list-style-type: none"> <li>- A.4: A priori determination of flux and torque current PI gains,</li> <li>A.9: References on page 145</li> </ul> <p>Added sections: Section 3.4: Flux-weakening PI controller tuning,</p> <p>Section 4.6.4: Additional parameters for IPMSM drive optimization(MTPA),</p> <p>Section 4.6.5: Additional parameters for feed-forward, high performance current regulation,</p> <p>Section 2.1.5: Feed-forward current regulation and A.8: MMI (maximum modulation index): automatic calculation.</p> <p>Single-shunt resistor configuration added:</p> <ul style="list-style-type: none"> <li>- Table 4.1: Library configuration file: stm32f10x_MCconf.h on page 43 updated</li> <li>- SVPWMEOCEvent and SVPWMUpdateEvent on page 64 added</li> <li>- Section 5.2: Current reading in single shunt resistor topology and space vector PWM generation: stm32f10x_svpwm_1shunt module on page 73 added</li> <li>- SVPWMEOCEvent and SVPWMUpdateEvent on page 77 added</li> <li>- SVPWMEOCEvent and SVPWMUpdateEvent on page 86 added</li> <li>- TIM1_UP_IRQHandler on page 133 updated</li> <li>- ADC1_2_IRQHandler on page 134 updated</li> <li>- Appendix A: Additional information on page 135 updated</li> </ul> <p>Small text changes.</p>
--	--	---

**Please Read Carefully:**

文件中的信息只与ST有关。STMicroelectronics NV和子公司 (“ST”) 保留更改，更正，修正或者改善文件及这里提及的产品和服务的权利，恕不事先通知。

所有的ST产品都有依照ST的条款和销售条件销售。

购买者只有责任选择和使用ST产品和服务，ST不负责任何与选择、使用ST产品和服务相关的。

文件不会假定与知识产权相关的许可证、解释或者暗示的禁言或者其它什么。。如果这个文件任何部分涉及到第三方产品或服务，ST都不会提供使用这种第三方产品或服务甚至其中包含知识产权的的许可证，或者以任何方式使用第三方产品、服务或包含知识产权的保证书。

**除非ST团队声明，ST销售条件拒绝承认任何解释或者暗示性保证与ST产品相关的，包括出于特别目的合适性、可销性无限制的保证(或者等同于权限法律)，或者侵犯知识产权。除非由ST代表授权支持，不推荐或保证ST产品用于军事、航天飞机、太空、生命救援或者生命维持应用，不推荐或保证ST产品或系统失效和复合功能引起个人伤害、死亡或严重的环境财产破坏。未作“汽车级”声明的ST产品在做汽车上应用时由用户自己承担风险。**

从不同陈述或声明的技术特征转售ST产品都会立即将ST对ST和服务不给予保证，而且不会以任何方式创造或延伸ST法律责任。

ST和ST是ST在各个国家注册或拥有的商标。

文件中的信息替代所有之前的信息。

ST标志是STMicroelectronics公司注册商标。所有其它名称都是所有者财产。

© 2008 STMicroelectronics – 保留所有权利

STMicroelectronics 公司小组

澳大利亚-比利时-巴西-加拿大-中国-捷克共和国-芬兰-法国-德国-香港-印度-以色列-意大利-日本-马来西亚-马尔他-摩洛哥 –新加坡-西班牙-瑞典-瑞士-英国-美国

**[www.st.com](http://www.st.com)**