

基于 STM32 的有感磁场定向控制（FOC）系统实现

1.1 引言

磁场定向控制也称 FOC，是一种电机控制方法，本章的设计将使用 FOC 算法对一个 PMSM 进行矢量控制，在这种控制方式下，电机能够输出很高精度的力矩和转子位置。

1.2 PMSM 数学模型

本节采用旋转坐标系（dq 轴），则 PMSM 的动态特性由公式(1-1)、(1-2)描述：

$$v_d = i_d R_s + L_d \frac{di_d}{dt} - \omega_e L_q i_q \quad (1-1)$$

$$v_q = i_q R_s + L_q \frac{di_q}{dt} + \omega_e (L_d i_d + \Psi_f) \quad (1-2)$$

式中 v_d, v_q ——d 轴和 q 轴电压（V）；

i_d, i_q ——d 轴和 q 轴电流（A）；

R_s ——定子电阻（ Ω ）。

L_d, L_q ——d 轴和 q 轴电感（H）；

ω_e ——电角速度（rad/s）；

Ψ_f ——永磁体磁链（Wb）；

对 i_d 和 i_q 进行前馈解耦：

$$v'_d = v_d + \omega_e L_q i_q \quad (1-3)$$

$$v'_q = v_q - \omega_e (L_d i_d + \Psi_f) \quad (1-4)$$

解耦后的电压方程简化为：

$$v'_d = i_d R_s + L_d \frac{di_d}{dt} \quad (1-5)$$

$$v'_q = i_q R_s + L_q \frac{di_q}{dt} \quad (1-6)$$

此时，dq 轴电流的传递函数为：

$$G_d(s) = \frac{i_d(s)}{v'_d(s)} = \frac{\frac{1}{R_s}}{1 + \frac{L_d}{R_s}s} \quad (1-7)$$

$$G_q(s) = \frac{i_q(s)}{v_q'(s)} = \frac{\frac{1}{R_s}}{1 + \frac{L_q}{R_s}s} \quad (1-8)$$

1.3 磁场定向控制算法架构

FOC 算法主要由 PI 控制器、电流坐标变换、空间矢量 PWM 调制以及转子位置反馈这几个部分组成，整体控制结构图如图 1-1 所示：

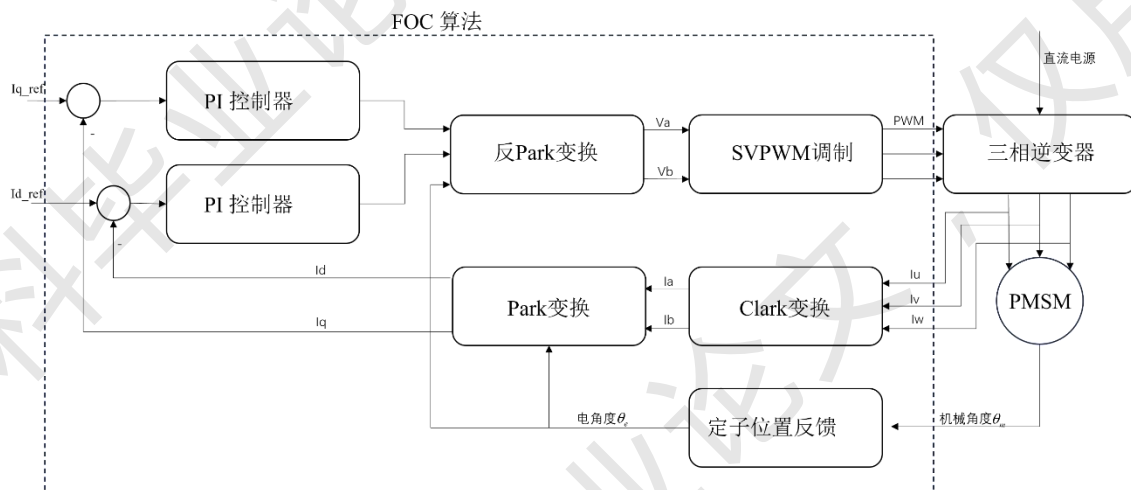


图 1-1 FOC 算法结构图

1.3.1 电流坐标变换

电流坐标变换是 FOC 算法中很重要的步骤。PMSM 的三相电流通过 Clark 变换（幅值不变）：

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_u \\ i_v \\ i_w \end{bmatrix} \quad (1-9)$$

将三相电流转换到两相静止坐标系下的两相电流 i_α, i_β ，再通过 Park 变换：

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \quad (1-10)$$

将两相电流转换到随电机转子同步旋转的坐标系下的电流 i_d, i_q ，为后续转矩电流和励磁电流的独立控制提供了便利。同时控制器输出的两相电流信号需要通过 Park 反变换：

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} \quad (1-11)$$

转换到两相静止坐标系下，便于 PWM 的调制。

在仿真软件中搭建电流坐标变换模块如图 1-2 所示，电机三相电流信号用三个幅值相等，相位依次滞后 120° 的正弦信号代替，得到仿真结果，其中图 1-3 为电机三相电流，图 1-4 为 Clark 变换后的电流，图 1-5 为 Park 变换后的 dq 轴电流。经过电流坐标变换，相互耦合三相电流被解耦为相互独立的转矩电流和励磁电流。

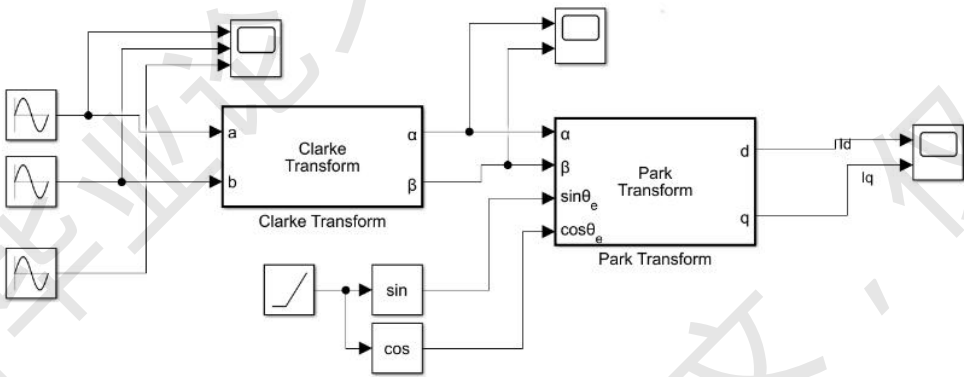


图 1-2 仿真结构图

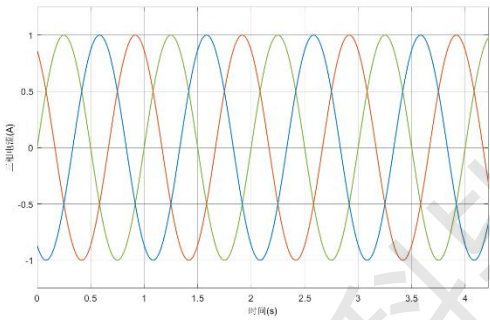


图 1-3 三相电流

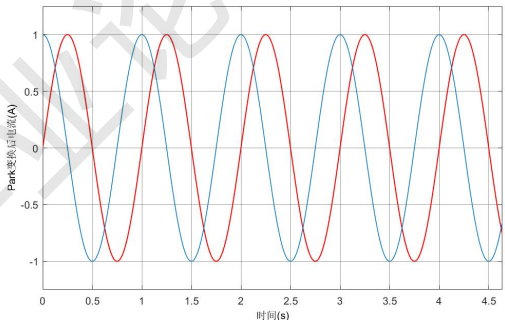


图 1-4 Clark 变换后

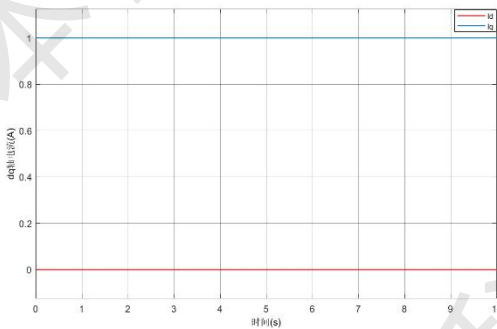


图 1-5 Park 变换后

1.3.2 空间矢量 PWM 调制

空间矢量 PWM 调制是矢量控制系统中至关重要的输出环节，它将经过坐标变换和电流调节器处理后的两相旋转坐标系(dq 轴)电压指令，通过特定的算法转换为三相逆变器所需的开关信号，从而实现对电机的高效精确控制。作为连接控制算法与功率器件的桥梁，SVPWM 通过合理利用电压矢量的空间分布和时序组

合,不仅能够显著提高直流母线电压利用率,还能有效降低谐波含量和开关损耗,是确保矢量控制系统动态性能和稳态精度的关键技术。该调制方法基于电压空间矢量理论,通过在一个 PWM 周期内合成目标电压矢量,使得电机获得近似圆形的旋转磁场,从而实现了对交流电机转矩和磁场的精准解耦控制。三相逆变器的 6 个开关管共有 8 种基本开关状态组合(6 个有效矢量+2 个零矢量),每种开关状态对应特定的空间电压矢量,作用到 PMSM 上将产生 6 个不同方向的基本磁矢量。根据这 6 个基本磁矢量的空间分布,将空间划分为 6 个独立的扇区,通过每个扇区边界的两个基本矢量,线性组合出空间中任意位置的矢量。扇区划分如图 1-6 所示:

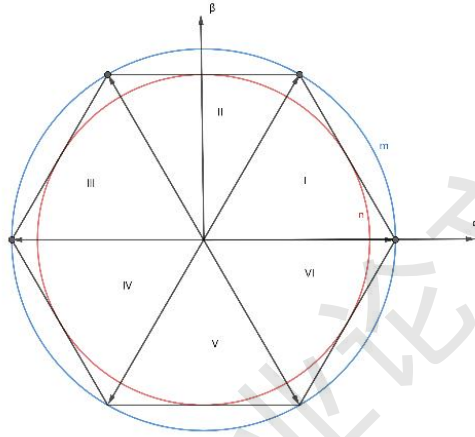


图 1-6 扇区图

SVPWM 模块的输入是待调制空间电压矢量在两相静止坐标系下的两个坐标 v_α, v_β , 首先需要根据这两个电压值判断出该空间电压矢量所处的扇区。首先定义三个中间变量:

$$\begin{cases} U_1 = v_\beta \\ U_2 = \frac{\sqrt{3}}{2}v_\alpha - \frac{1}{2}v_\beta \\ U_3 = -\frac{\sqrt{3}}{2}v_\alpha - \frac{1}{2}v_\beta \end{cases} \quad (1-12)$$

根据 U_1, U_2, U_3 的符号判断扇区, 令:

$$\begin{cases} A = \text{sig}(U_1) \\ B = \text{sig}(U_2) \\ C = \text{sig}(U_3) \\ N = 4C + 2B + A \end{cases} \quad (1-13)$$

根据 N 的值得到扇区号后需要计算所处扇区边界基本矢量的线性组合系数。由图 1-6 可知, 终点落在圆 m 与圆 n 之间的矢量无法通过基本矢量线性组合得

到，为防止 SVPWM 出现过调制电压失真现象，规定只有处在圆 n 以内，幅值不大于基本矢量幅值的 0.866 的矢量才能通过线性组合得到。由 PWM（脉宽调制）平均值等效原理，线性组合系数等于基本矢量作用的时间，而两个零矢量的作用时间等于载波周期减去基本矢量作用时间。最后根据电压矢量和开关管导通组合的对应关系，计算出三组互补对称 PWM 的占空比，最终得到的理想 SVPWM 输出信号应该是马鞍波。

在仿真软件中搭建 SVPWM 调试模块如图 1-7 所示，并使用 STM32G431 单片机进行实物验证，其中输入信号为两个等幅值、相位相差 90° 的正弦信号，输出为三组 PWM 信号占空比，得到结果如图 1-8 所示：

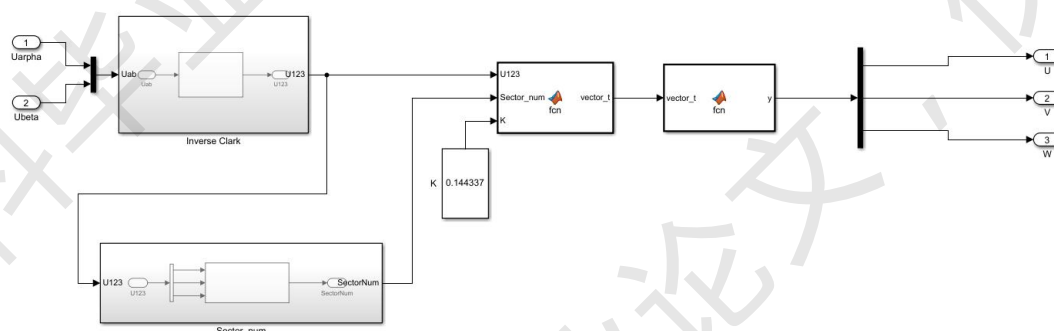


图 1-7 SVPWM 仿真结构图

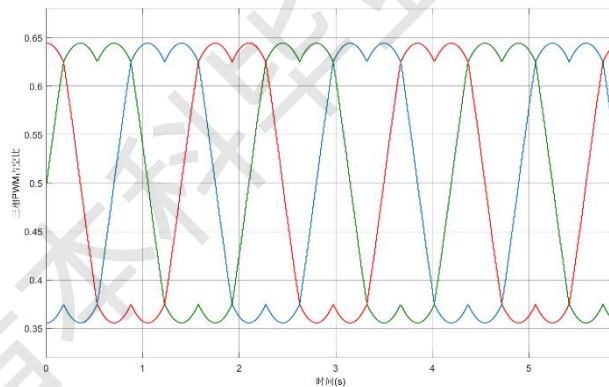


图 1-8 SVPWM 仿真输出马鞍波

1.3.3 PI 控制器参数计算

电流环采用 PI 控制器，设 PI 控制器传递函数：

$$G_c(s) = K_p + \frac{K_I}{s} \quad (1-14)$$

则系统的闭环传递函数为：

$$\Phi(s) = \frac{G_c(s)G_0(s)}{1 + G_c(s)G_0(s)}$$

(1-15)

将 1.2 节得到 PMSM 的 dq 轴传递函数代入 $G_0(s)$ ，并假设 $L_d = L_q = L$ ，于是得到：

$$\Phi(s) = \frac{\frac{1}{L}(K_p s + K_I)}{s^2 + (\frac{R_s + K_p}{L})s + \frac{K_I}{L}} \quad (1-16)$$

将闭环系统设计为二阶系统，令阻尼比 $\xi = 0.707$ ，此时自然振荡角频率 ω_n 与系统带宽有关，取 $\omega_n = 2\pi f_{BW}$ ，其中 f_{BW} 为电流环带宽，一般为 PWM 频率的 0.1~0.2，若 PWM 频率取 20KHz，电流环带宽 f_{BW} 取 2KHz，对比标准二阶系统 $s^2 + 2\xi\omega_n s + \omega_n^2$ ，可以得到：

$$\frac{R_s + K_p}{L} = 2\xi\omega_n \quad (1-17)$$

$$\frac{K_I}{L} = \omega_n^2 \quad (1-18)$$

解得：

$$\begin{cases} K_p = 2\xi\omega_n L - R_s \\ K_I = \omega_n^2 L \end{cases} \quad (1-19)$$

最后，在实际数字实现时需要进行离散化，可以使用欧拉法或双线性变换。电流环 PI 参数的推导是按典型 I 型系统设计的。实际调试时需结合带宽需求与系统稳定性微调参数。

1.4 实物验证

1.4.1 硬件电路分析

硬件电路主要分为三相逆变器、电流测量网络、微控制器以及辅助电路这几个主要部分。其中三相逆变器由三相栅极驱动器和 MOSFET 组成，三相栅极驱动器采用 FD6288Q。由于微控制器的输出电流通常仅为毫安级，无法直接驱动大功率器件，于是需要栅极驱动器为 MOSFET 的栅极提供足够的驱动电流，以确保 MOSFET 的快速充放电以实现开关状态的切换。MOSFET 构成三相逆变器的三相桥臂，由于 MOSFET 的栅极回路存在寄生电感（如 PCB 走线、器件引脚），与栅极电容可能形成 LC 谐振电路，需要在 MOSFET 的栅极各串联一个 33 欧姆的电阻，通过串联电阻增加阻尼系数，可以抑制电压振荡防止误触发或栅极电压超调。原理图如图 1-9 所示。

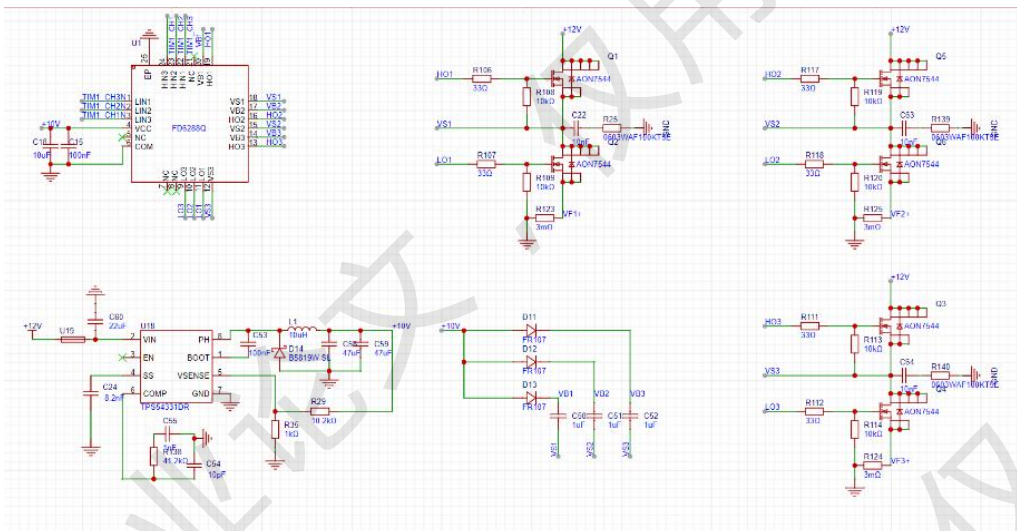


图 1-9 三相逆变器原理图

电流测量模块由电流采样电阻、运算放大器以及 A/D 转换器构成。本设计采用低端双电阻电流采样方案，电流采样电阻采用阻值为 3 毫欧，精度为 1%，功率为 3W 的纯合金贴片电阻，将它串联在三相桥的下桥臂与 GND 之间，它的作用是将电流信号转换为电压信号，但是由于阻值过小，得到的电压信号通常为毫伏级，因此需要运算放大器进行放大处理。

本设计采用 STM32G431CBU6 作为微控制器，这款微控制器内部集成了运算放大器与 12 位 A/D 转换器，但是由于运算放大器使用单电源供电只能放大正电压信号，需要在外部提供偏置电压，本设计采用电阻分压网络在 MCU 外部提供 2.08V（放大后）的偏置电压。辅助电路由电源模块与 CAN 收发模块组成，原理图如图 1-10 和图 1-11 所示。

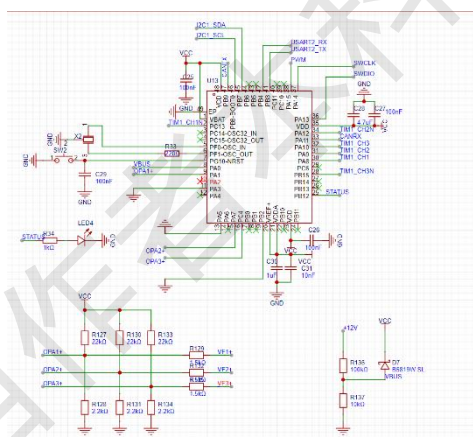


图 1-10 微控制器

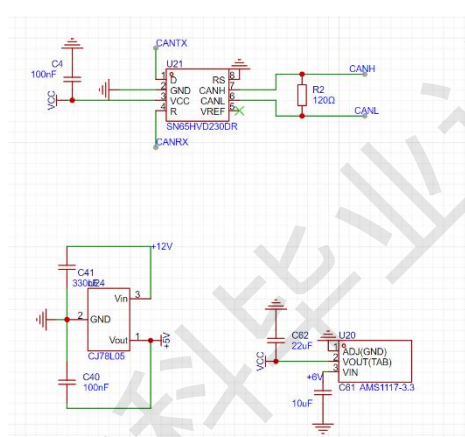


图 1-11 辅助电路

1.4.2 软件程序设计

软件程序使用 C 语言编写，首先使用 STM32 官方提供的工具 STM32CUBEMX 配置 MCU 的外设寄存器，并构建 HAL 库工程模板。需要使用到的外设有高级控制定时器 TIM1、通用定时器 TIM3、可编程增益运算放大器

OPAMP1\2、模数转换器 ADC1\2、串行外设接口 SPI1 以及控制器局域网络 FDCAN1。高级控制定时器 TIM1 输出 PWM 信号并且触发 ADC 采样,其中 TIM1 的通道 1-3 用于输出 3 路频率为 20KHz 的互补对称 PWM 信号,通道 4 用于在对称 PWM 信号的中点产生 ADC 采样触发信号,由图 1-12 可知在对称 PWM 信号的中点,三相桥的上桥臂均处于开启状态,此时 PMSM 的定子线圈将处于续流状态,适合进行电流检测。TIM3 和 SPI1 用于进行 PMSM 转子位置的测量。

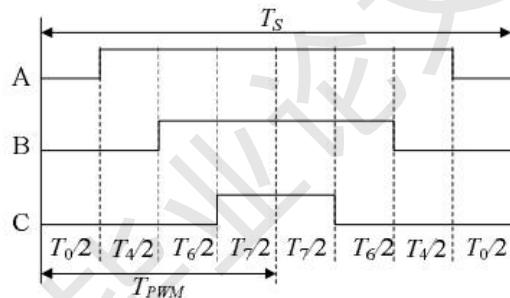


图 1-12 中心对称 PWM 信号

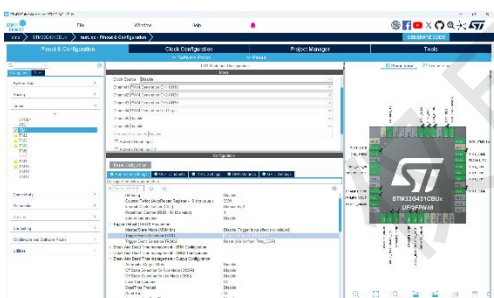


图 1-13 STM32 外设配置

MCU 通过 SPI 接口与 AS5048A 传感器通信,将 AS5048A 的 SPI_MOSI 引脚连接至高电平,即可实现芯片上电后自动发送 14 位位置数据,设置 TIM3 产生 20KHz 的中断,在中断服务中定时启用 SPI 外设,获取转子位置信息。OPAMP 外设配置为 16 倍放大系数,并在内部连接至 ADC 外设。为方便使用上位机调试以及与其他控制器通信的需要,将 FDCAN 配置为经典 CAN 模式,并配置合适的波特率以及消息过滤器。软件程序的流程图如图 1-14 所示:

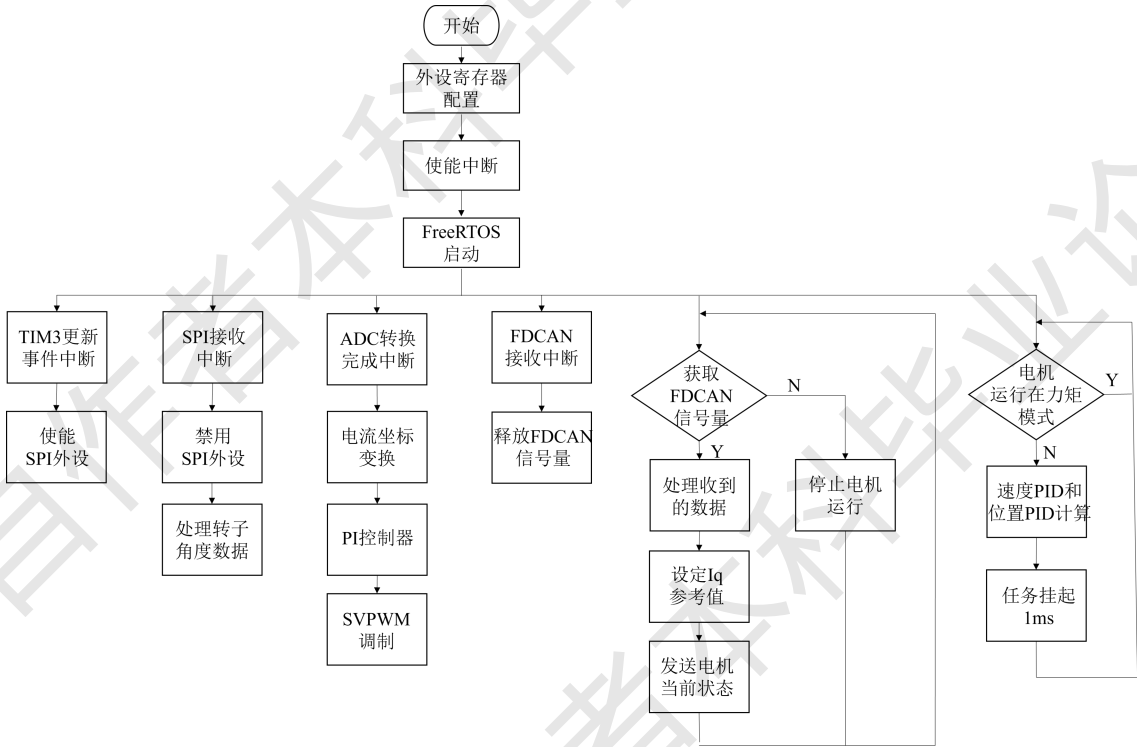


图 1-14 程序流程图

FOC 的核心运算部分在 ADC 转换完成中断中执行,通过 TIM1 通道 4 的

PWM 信号精确控制电流测量频率为 20KHz,在 ADC 中断服务中,首先清除 ADC 中断寄存器标志位,并读取注入组数据寄存器中的转换数据,然后关闭全局中断,程序进入临界区,此时获取转子角度位置以及 FreeRTOS 任务中设定电流参考值,之后退出临界区,开启中断,这样做是为了确保数据的安全传递,避免 ADC 中断与其他中断以及主进程出现同时操作共享资源,导致数据损坏。之后进行电流的坐标变换,将三相电流解耦到 dq 轴,并分别进行 d 轴和 q 轴 PI 控制器的计算。最后将 PI 控制器的输出做 Park 反变换,并通过 SVPWM 模块设定 TIM1 的 3 路互补对称 PWM 通道的输出比较寄存器。FOC 算法核心代码如图 1-15 所示。

```
void ADC1_2_IRQHandler(void)
{
    float cos_Theta,sin_Theta;
    uint16_t adcbuf[3];
    float Ua,Ub;
    LL_ADC_ClearFlag_JEOS(ADC2);

    adcbuf[0]=ADC1->JDR1;
    adcbuf[2]=ADC1->JDR2;
    adcbuf[1]=ADC2->JDR1;

    Get_CurrentSensor_data(adcbuf,&hcurrent_sensor1);
    Get_SpeedSensor_data(spi_rdata,&hspeed_sensor1);
    Process_CurrentSensor_data(&cos_Theta,&sin_Theta,hspeed_sensor1.electronic_angle_us16,&hcurrent_sensor1);
    hctrl1.Id_ctrl_pid.pre_error = hctrl1.Id_ctrl_pid.error;
    hctrl1.Id_ctrl_pid.error = hctrl1.Id_Set - hcurrent_sensor1.Id;
    PID_Controller(&(hctrl1.Id_ctrl_pid));

    hctrl1.Iq_ctrl_pid.pre_error = hctrl1.Iq_ctrl_pid.error;
    hctrl1.Iq_ctrl_pid.error = hctrl1.Iq_Set - hcurrent_sensor1.Iq;
    PID_Controller(&(hctrl1.Iq_ctrl_pid));
    /*Idq_pid_out=Udq*/
    arm_inv_park_f32(hctrl1.Id_ctrl_pid.output,hctrl1.Iq_ctrl_pid.output,&Ua,&Ub,sin_Theta,cos_Theta);

    svpwm_handle.Ualpha = Ua;
    svpwm_handle.Ubeta  = Ub;
    SVPWM_Generator(&svpwm_handle);
}
```

图 1-15 FOC 核心代码

1.4.3 实验测试性能

组装调试完成后的实际系统如图 1-16 所示：

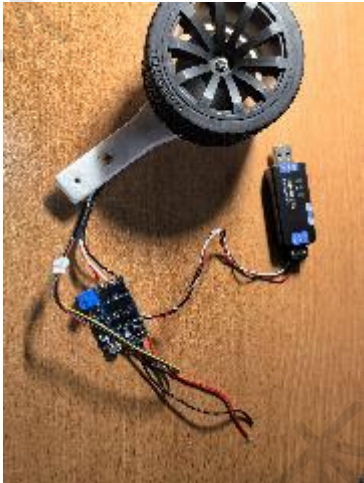


图 1-16 电机系统实物图

设置电机工作在力矩模式下,上位机通过 CAN 接口与 MCU 通信,上位机软件设定 Iq 的参考信号为正弦信号,测试电机的跟随特性,结果如图 1-17 所示。由实验结果可以看出,力矩模式下系统跟踪 10Hz 以内的正弦信号是基本没有误

差的。设置电机工作在位置模式下，上位机软件设定转子位置的参考信号为频率 1Hz，幅值 3.14 rad 的正弦信号，实验测试效果如图 1-19 所示。由图 1-19 可知，系统在位置模式下可以稳定、快速且无误差的跟踪频率为 1Hz，幅值为 3.14 rad 的正弦信号。

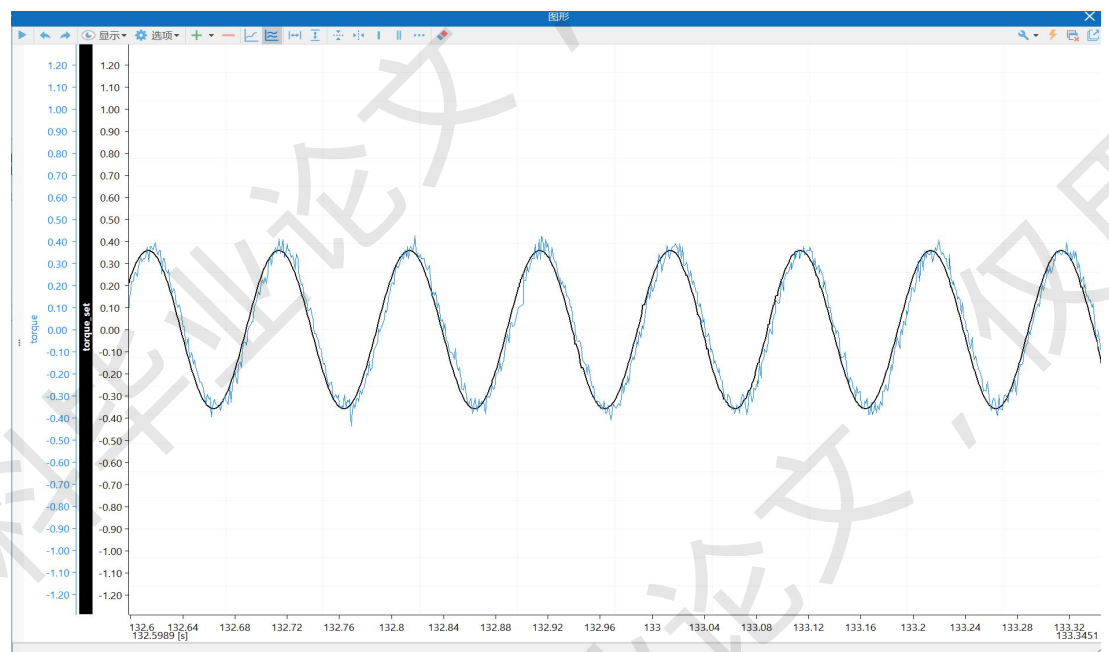


图 1-17 力矩模式 10Hz 正弦波

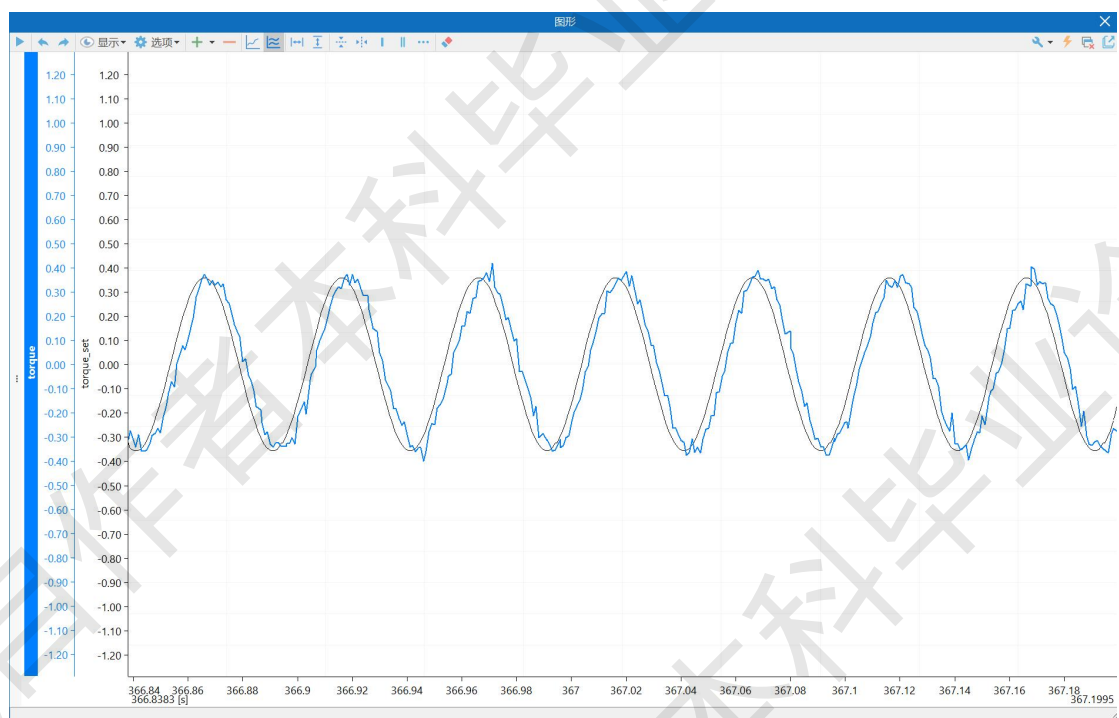


图 1-18 力矩模式 20Hz 正弦波

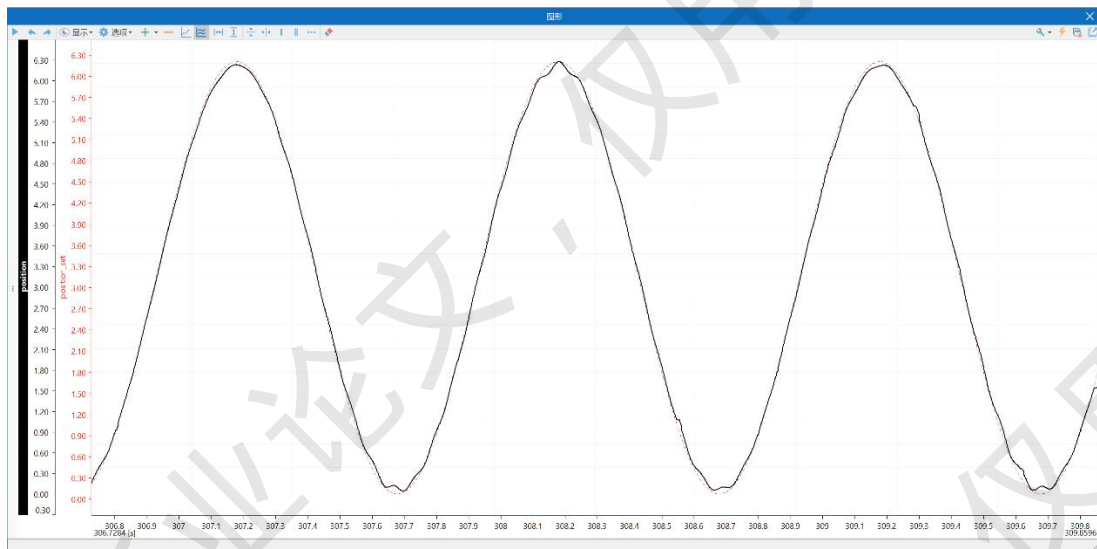


图 1-19 位置模式 1Hz 正弦波

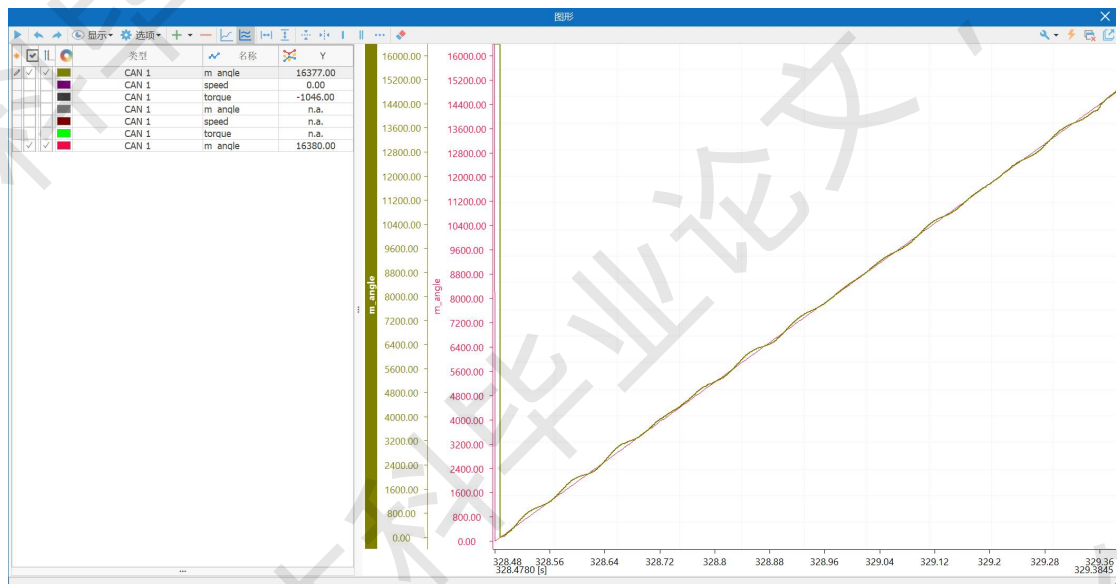


图 1-20 位置模式跟踪 $2\pi/s$ 的斜坡信号



图 1-21 速度模式跟踪 1Hz 正弦波

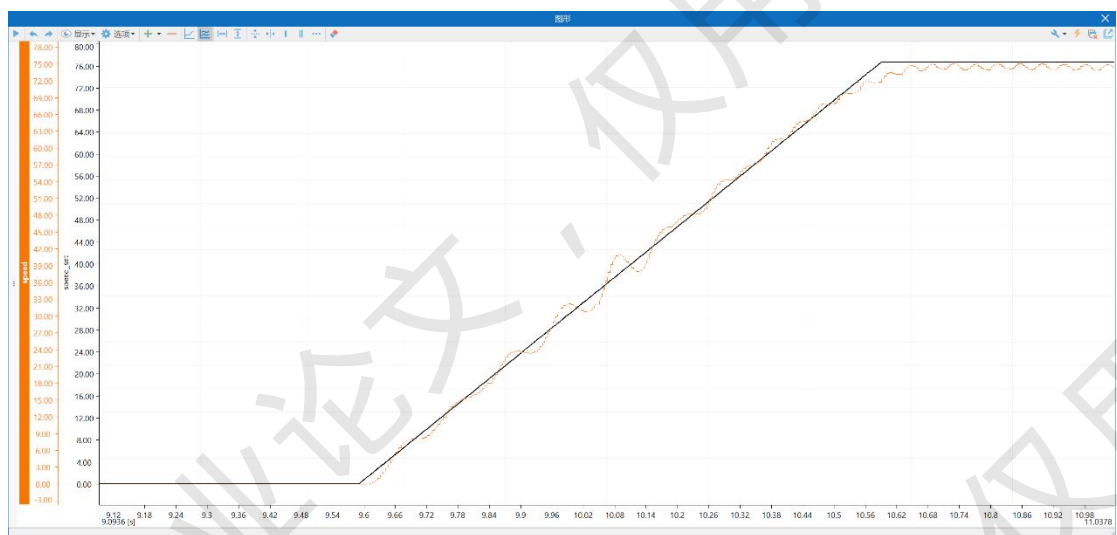


图 1-22 速度模式跟踪斜坡信号

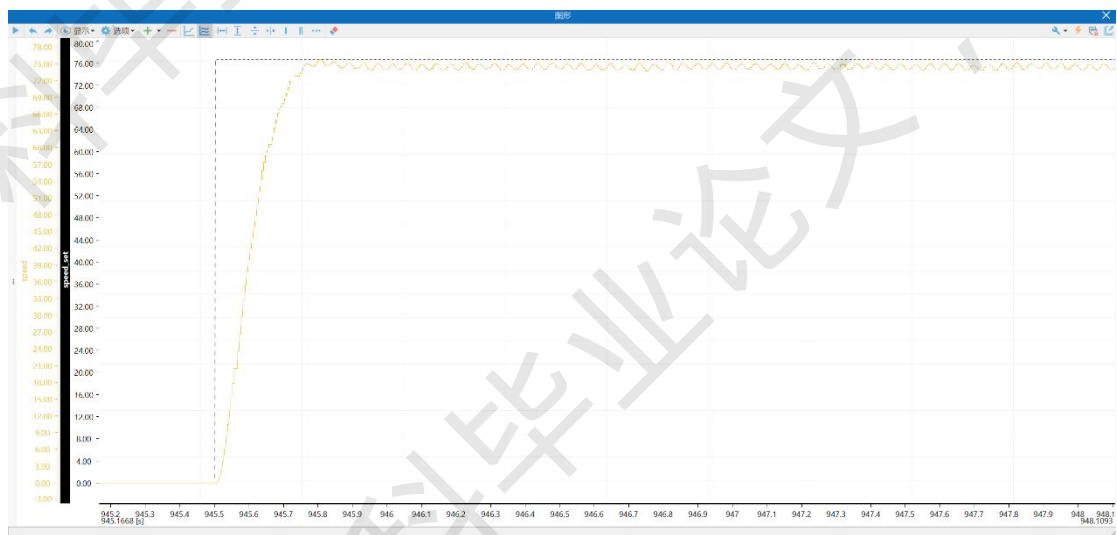


图 1-23 速度模式跟踪阶跃信号