

NuMotor MCU NM1244 以 SMC 实现无感测向量控制(FOC)

1.1 简介

本控制以 2 个 shunt 电阻作为电流取样的 3 相 PMSM 矢量控制驱动器的设计，没有位置传感器，该设计面向以消费者和工业应用，这个性价比高解决方案得益于 Nuvoton NuMotor NM1244 微控制器应用于电机控制。

1.2 NM1244 微控制器的优势和特点

■ Core

- ARM® Cortex® -M0 core running up to 48/60 MHz by internal RC oscillator optioned from ROMMAP
- One 24-bit system timer
- Supports low power Idle mode
- Built-in LDO for wide operating voltage ranged: 2.2 V to 5.5 V

■ Memory

- 48/64 Kbytes Flash memory for program memory (APROM)
- Configurable Flash memory for data memory (Data Flash)

■ I/O Port

- Up to 44 general-purpose I/O (GPIO) pads and 1 Reset pad

■ Timer

- Provides three channel 32-bit Timers; one 8-bit pre-scalar counter with 24-bit up timer for each timer
- Independent clock source for each timer
- Provides One-shot, Periodic, Toggle and Continuous operation modes
- Timer0, Timer1, Timer2 and SysTick provided with continuous capture function to capture at most 4

edges continuously on one signal

■ ECAP (Enhanced Input Capture)

- One units of 24-bit input capture counter

■ GDMA (General Direct Memory Access)

- Two channels
- Memory to/from memory
- Memory to/from USCI

■ WDT (Watchdog Timer)

- Programmable clock source and time-out period
- Supports wake-up function in Power-down mode and Idle mode
- Interrupt or reset selectable on watchdog time-out

■ EPWM

- Supports a built-in 16-bit PWM clock generators, providing SIX PWM outputs or three complementary paired PWM outputs

■ BPWM

One 16-bit PWM generator which supports one 8-bit pre-scalar, one clock divider, two PWM timers (down counter) and one dead-zone generator

Synchronous mode for BPWM and EPWM

USCI (Universal Serial Control Interface Controller)

USCI1 Supports to be configured as UART, SPI or I²C individually

USCI2 Supports to be configured as UART and I²C individually

Supports programmable baud-rate generator

■ ADC (Analog-to-Digital Converter)

12-bit ADC with 1 μ s conversion time

Supports 2 S/H (sample/hold)

Up to 16-ch single-end input from I/O and one internal input from band-gap.

■ OP Amplifier

Rail-to-rail OPA x 1

Unity gain frequency up to 6 MHz

Supports OP1 output as input of ADC and ACMP

■ DAC

Built-in two 12-bit DAC,

Be the reference voltage for ACMP, ADC or output to pins

■ Analog Comparator

One analog comparators with 4 reference voltage source

Built-in 12-bit DAC0 and DAC1 for comparator reference voltage

Band-gap voltage

■ Hardware Divider

32-bit dividend with 16-bit divisor calculation capacity

32-bit quotient and 32-bit remainder outputs (16-bit remainder with sign extends to 32-bit)

■ BOD (Brown-out Detector)

8 programmable threshold levels: 4.3V/4.0V/3.7V/3.0V/2.7V/2.4V/2.2V/2.0V

Supports Brown-out interrupt and reset option

96-bit unique ID

■ LVR (Low Voltage Reset)

Operating Temperature: -40°C~105°C

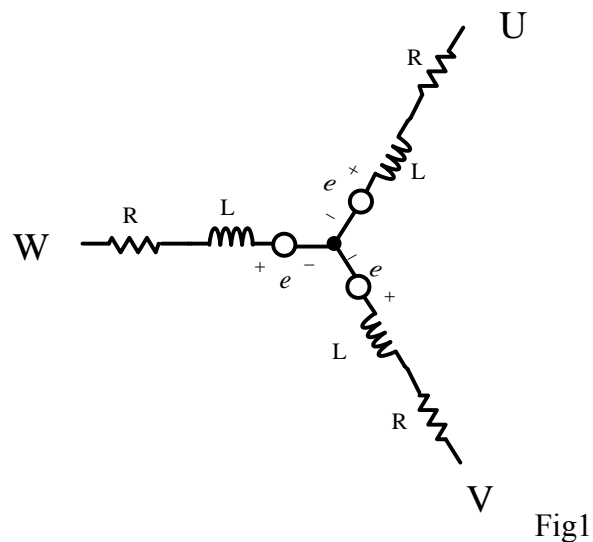
Reliability: EFT > \pm 4KV, ESD HBM pass 4KV

Packages:

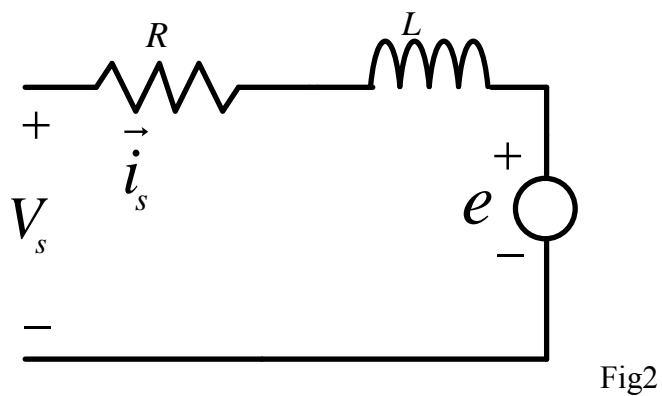
Green package (RoHS)

48-pin LQFP(7x7mm), 48-pin QFN(7x7mm), 32-pin QFN(4x4mm)

2 由 PMSM 电气 Mode 可以得到三相等效电路 Fig1



单相等效电路 Fig2



由电机等效电路就可以得到电机的电压方程

$$V_s = Ri_s + L \frac{d}{dt} i_s + e_s$$

V_s = 输入电压

i_s = 电机电流

e_s = 电机反电势(back EMF)

R = 电机电阻

L = 电机绕组电感

$\frac{d}{dt}$ = 控制周期

则解电机电流 i_s

$$\text{电压方程} = \frac{d}{dt} i_s = \left(-\frac{R}{L} \right) i_s + \frac{1}{L} (V_s - e_s), i_s = [i_\alpha \ i_\beta]^T, e_s = [e_\alpha \ e_\beta]^T$$

将电压方程离散化可得

$$\frac{i_s - i_s^{-1}}{P_s} = \left(-\frac{R}{L}\right) i_s^{-1} + \frac{1}{L} (V_s^{-1} - e_s^{-1})$$

$$\gg i_s = \left(1 - P_s \times \frac{R}{L}\right) i_s^{-1} + \frac{T_s}{L} (V_s^{-1} - e_s^{-1})$$

P = period

由上式可得到 F 参数及 G 参数分别为

$$F = 1 - P_s \times \frac{R}{L}$$

$$G = \frac{P_s}{L}$$

2.1 Sliding mode control (SMC)电流估测器说明

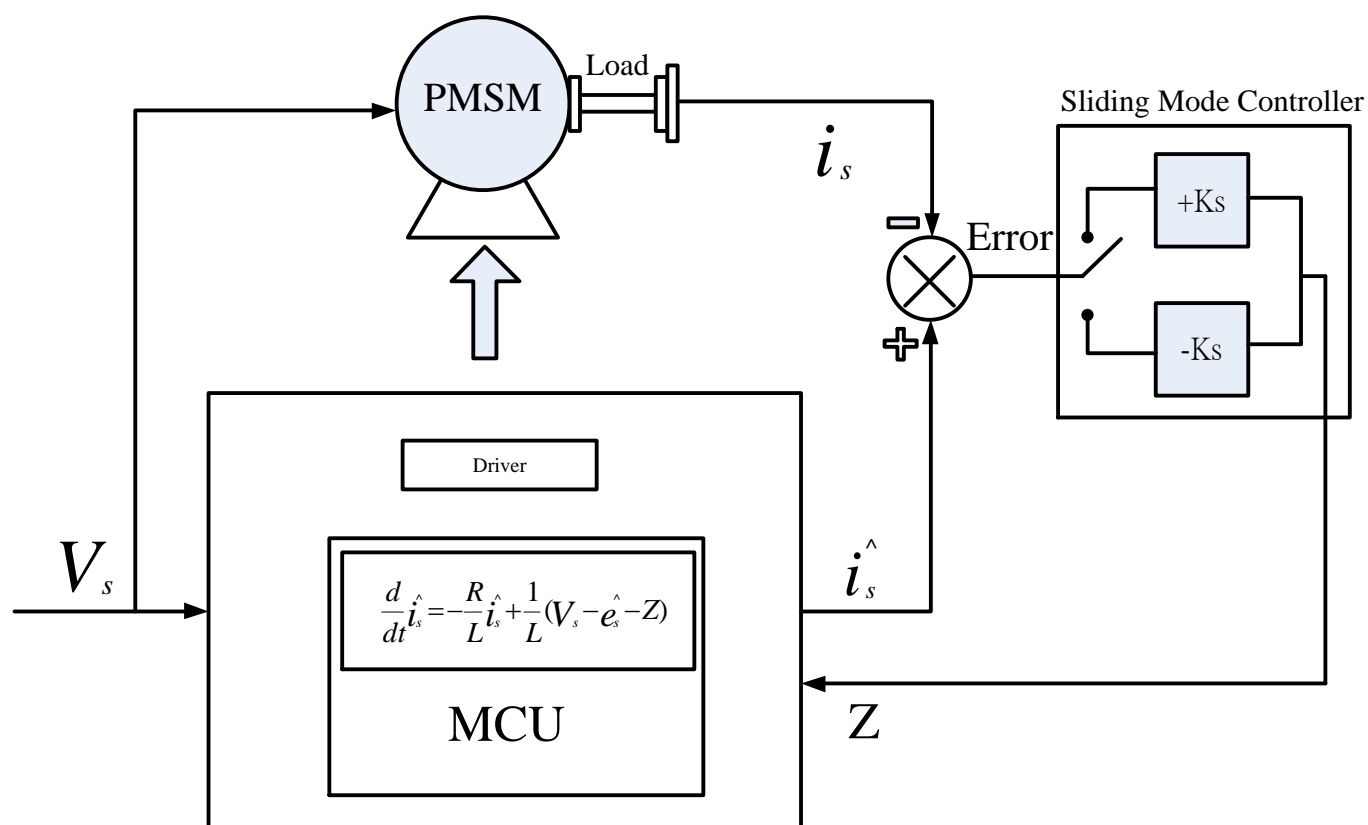
数字化模型将硬件以软件化来表示，为了匹配测量电流和估测电流数字化电机模型需要使用闭环进行修正。考虑两个电机表示，一个在硬件外加电压 v_s 和一个在软件中 v_s ，具有相同的输入的两个系统将测量电流 (i_s) 与估计电流 (i_s^{\wedge}) 匹配，从模型中，我们可以假设我们数字化模型的反电动势(e_s^{\wedge}) 是与电机的反电动势相同，滑动模式控制器(SMC)适用于补偿数字化电机模型，原理是由一个求和点组成，该求和点计算两者之间的误差 Error 来自电机电流的测量值和来自估测模型计算出的电流值相减得到误差 Error (+1 或 -1) 再乘上 SMC 增益 (K)，这个增益被加入到补偿数字化模型后，你有一个电机模型输入电压 (V_s) 和电流 (i_s^{\wedge}) 的可变值，一旦数字化模型得到补偿，通过 LPF 输出后就可以得到估测的反电动势 (e_s^{\wedge})及校正项次 (Z)，如框图所示，反电动势估测(e_s^{\wedge}) 被回授到模型在每个控制周期后更新变量 e_s^{\wedge} 值 e_{α} 和 e_{β} (e_s 的向量分)

经 $\tan^{-1} \frac{e_{\beta}}{e_{\alpha}}$ 得到本次估测的电气角度 θ 。

将电压方程加入校正项次 (Z)可得

$$\frac{d}{dt} i_s^{\wedge} = -\frac{R}{L} i_s^{\wedge} + \frac{1}{L} (V_s - e_s^{\wedge} - Z)$$

2.2 电流估测器架构



2.3 SMC 的设计

基于滑模结构原理，滑动面选择为

$$S(X) = \hat{i}_s - i_s = 0 \quad (1)$$

其中 $\hat{i}_s = [\hat{i}_\alpha \ \hat{i}_\beta]^T$ 是目前的估测值 $i_s = [i_\alpha \ i_\beta]^T$ 是实际值，PMSM 的数学模型在静止框的 SMC 表示如下，估测器是在静止框下进行，估测到的角度给同步框作框转换。

$$\frac{d}{dt} \hat{i}_\alpha = -\frac{R}{L} \hat{i}_\alpha + \frac{1}{L} e_\alpha - k_1 \frac{1}{L} \text{sgn}(\hat{i}_\alpha - i_\alpha) \quad (2)$$

$$\frac{d}{dt} \hat{i}_\beta = -\frac{R}{L} \hat{i}_\beta + \frac{1}{L} e_\beta - k_1 \frac{1}{L} \text{sgn}(\hat{i}_\beta - i_\beta) \quad (3)$$

其中 k_1 是估测器的增益也是常数，sgn 是有号的函数也是 SMC 习用的，所以 SMC 达到稳定的条件定义

$$\dot{V} = S(X)^T \dot{S}(X) < 0 \quad (4)$$

通过从(2)(3)中减去电压方程可以获得误差方程为

$$\frac{d}{dt} S_\alpha(X) = -\frac{R}{L} S_\alpha(X) + \frac{1}{L} e_\alpha - k_1 \frac{1}{L} (\hat{i}_\alpha - i_\alpha) \quad (4)$$

$$\frac{d}{dt} S_\beta(X) = -\frac{R}{L} S_\beta(X) + \frac{1}{L} e_\beta - k_1 \frac{1}{L} (\hat{i}_\beta - i_\beta) \quad (5)$$

则

$$S(X) = \begin{bmatrix} S_\alpha(X) \\ S_\beta(X) \end{bmatrix} = \begin{bmatrix} \hat{i}_\alpha - i_\alpha \\ \hat{i}_\beta - i_\beta \end{bmatrix}$$

透过以上推导可以得到稳定条件表示为

$$\dot{V} = S(X)^T \dot{S}(X) = S_\alpha \dot{S}_\alpha + S_\beta \dot{S}_\beta$$

$$= \frac{1}{L} [(\hat{i}_\alpha - i_\alpha) e_\alpha - k_1 (\hat{i}_\alpha - i_\alpha) (\hat{i}_\alpha - i_\alpha)] + \frac{1}{L} [(\hat{i}_\beta - i_\beta) e_\beta - k_1 (\hat{i}_\beta - i_\beta) (\hat{i}_\beta - i_\beta)] - \frac{R}{L} [(\hat{i}_\alpha - i_\alpha)^2 + (\hat{i}_\beta - i_\beta)^2] \leq 0$$

结果

$$k_1 > \max(|e_\alpha|, |e_\beta|)$$

因此，确保 k_1 足够大的值可以令滑动运动的存在性和渐近稳定性都在范围内的运动，一旦系统达到滑动面，那么 $\dot{S}(X) = S(X) = 0$ ，即 $\hat{i}_\alpha = i_\alpha$ ， $\hat{i}_\beta = i_\beta$ ，由(4)(5)式基于等效控制方法得出

$$\begin{aligned} z_\alpha &= k_1 (\hat{i}_\alpha - i_\alpha) \\ z_\beta &= k_1 (\hat{i}_\beta - i_\beta) \end{aligned} \quad (6)$$

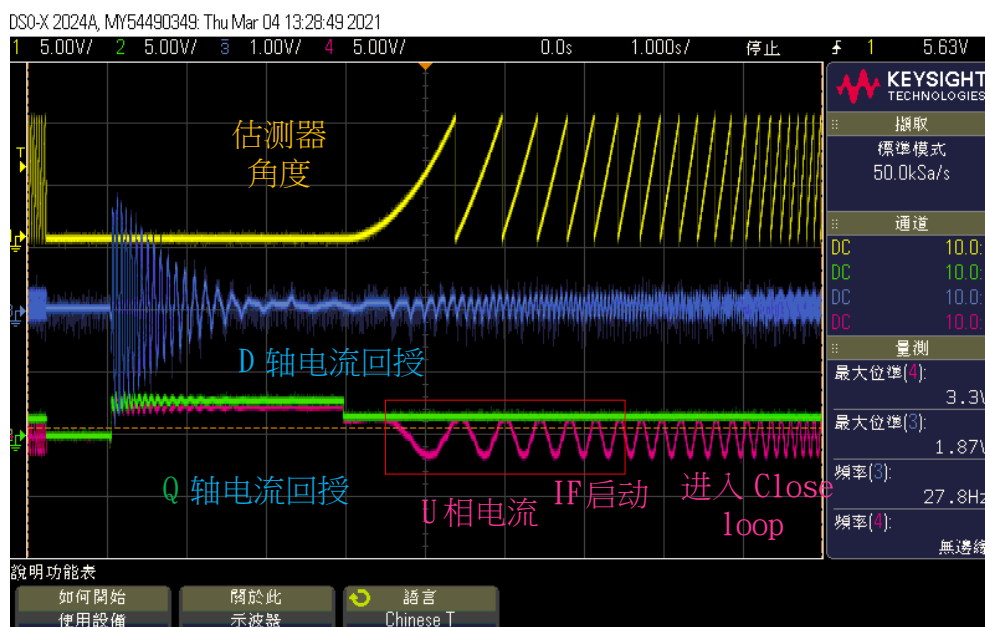
2.4 实际驱动马达如图 1,图 2, 图 3

图 1.



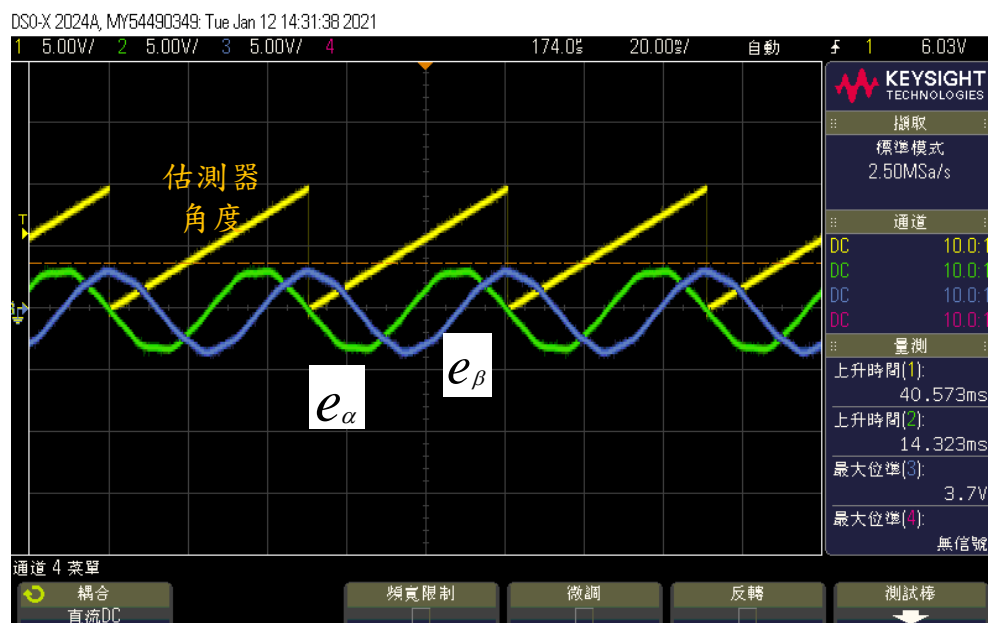
将估测器输出的角度与 Hall sensor 锁相的角度比对

图 2.



对运转中马达下零向量再以 IF 启动进 Close loop

图 3.



估测器角度及所对应估测到的 back EMF e_α, e_β



设定参数在 main.c 档案内有名为 FUN_Data_Type const FATAB[MAX_FUNC_QTY+1]的矩阵中，参数共有 120 个，所有的设定应用参数都集中一起方便管理及提高效率，因为每个参数都有批注编号容易搜寻，没有用到的皆为保留项次，程序开始执行时会将设定值填入该参数此为 Default，所以需要改变参数时，只要填入设定值然后经 Keil C compile 在将档案 load 进 NM1244 就完成，以下是以参数//F107 作为范例说明

//F_107

```
{&FunPar.dbup_cnt,0,255,B16|DOT0,
30,50,50,50,50,50,50,50},
```

数组内容说明

1	2	3	4	5	6~12 保留无用
参数名称	参数设定下限	参数设定上限	参数长度	设定值	
&FunPar.dbup_cnt	0	255	B16(16 位)	30	

第五个字段即是设定值，填入欲改变的值在参数的上下限内经 compile 再 load 进 NM1240 就完成，参数变量宣告在 Glib_reg_def.h 内的结构名称 PARA 内，若要修改变量名称要两边同时修改否则会出现错误。

2.7 参数与软件的数字设定说明

整个软件控制架构可以区分 5 个区块, 1.估测器(Observer), 2.磁场导向控制(FOC), 3.速度回路(Speed PI), 4.IF 启动, 5.周边, 以下是一说明

(1) 估测器(Observer)

F_089 Smo_KslfValue_Min 设定值=10 KSlidef 的最小累加值

公式 $KSlidef = F_{089} + (F_{104} * I_q) / 4096$

F_090 Smo_Kslide_Value 设定值=80, 在软件内部 $(80 * 10) / 1024 = 0.7812$

F_091 Smo_G_Value 设定值=190, 在软件内部 $(190 * 10) / 1024 = 1.8554$

F_092 Smo_F_Value 设定值=51, 在软件内部 $(51 * 10) / 4096 = 0.1245$

F_095 ZalfaBetaLPFGain 设定值=80, 在软件内部 $(80 * 10) / 4096 = 0.1953$

F_096 EalfaBetaLPFGain 设定值=80, 在软件内部 $(80 * 10) / 4096 = 0.1953$

F_104 KslfGain 设定值=200, 在软件内部 $200 / 4096 = 0.0488$

F_081 FrqFbFilter 为估测器估出的电机电气频率滤波器其设定值=145

(2) 磁场导向控制(FOC)

D Axis

F_075 Id_IGain 设定值=100, 在软件内部 $100 / 32767 = 0.000355$

F_076 Id_PGAIN 设定值=200, 在软件内部 $(200 * 10) / 8192 = 0.2441$

F_078 Id_PIOutLimit 设定值=2649, 在软件内部 $2649 * 10 = 26490$

I_Limit = 32767 为固定值

Q Axis

F_082 PI_OutLimit 设定值=2785, 在软件内部 $2785 * 10 = 27850$

F_083 I_Limit 设定值=32767, 在软件内部 32767

F_084 Iq_PGAIN 设定值=1776, 在软件内部 $(1776 * 10) / 32767 = 0.542$

F_085 Iq_IGain 设定值=1069, 在软件内部 $1069 / 32767 = 0.0326$

(3)速度回路(Speed PI)

F_070 Spd_I_Limit 设定值=3276, 在软件内部 $3276 * 10 = 32760$

F_071 Spd_PI_Limit 设定值=2500, 在软件内部 $2500 * 10 = 25000$

F_072 Spd_Igain 设定值=15, 在软件内部 $15 / 8192 = 0.000183$

F_073 Spd_Pgain 设定值=20, 在软件内部 $(20 * 10) / 8192 = 0.0244$

F_077 SpdFeedfward 设定值=20, 在软件内部 $20 * 10 = 200$

(4) IF 启动

F_003 first_cmd 设定值=100, 在软件内部 $100 * 10 = 1000(10.00 \text{ Hz})$

F_005 first_acctime 设定值=150, 在软件内部 150 sec

F_086 Iq_cmd 设定值=10, 在软件内部 $10 * 100 = 1000$

F_099 IFStarAccTime 设定值=600, 在软件内部 600 sec

(5) 周边

F_000 设定值=0 时外部启动

F_001 设定值=0 时频率命令由 F_003, 设定值=1 时频率命令由外部的 VR

2.8 NM1244 EVB

