

基于STM32的 PMSM FOC软件库 培训



MCU Application
Great China

蒋建国

Shanghai,
March, 2008



Plan

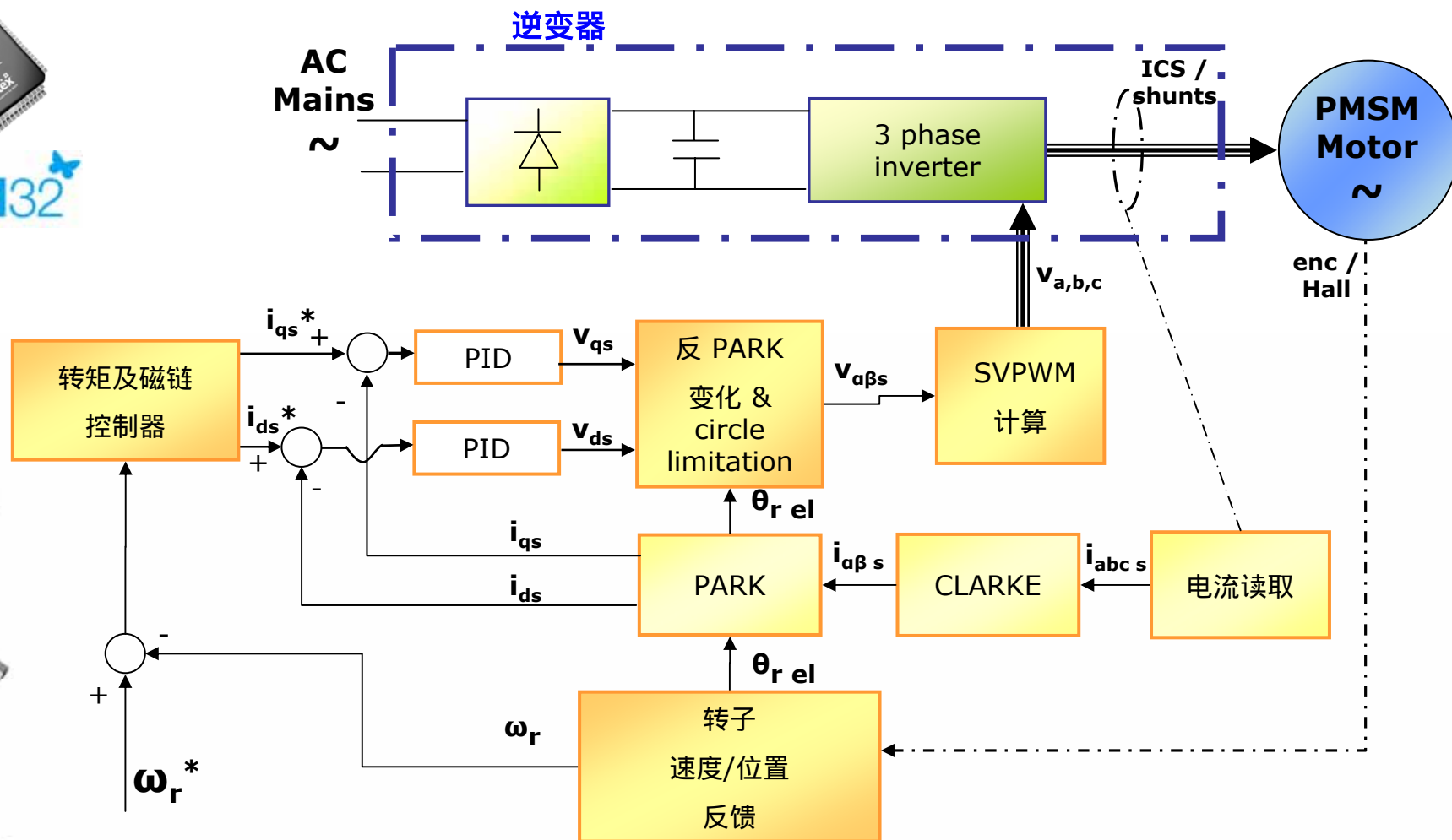
无传感器转子位置检测：

-  框图
-  观测器理论
-  应用到**PMSM**马达系统
-  **PLL**
-  起动策略
-  转子堵转检测
-  头文件 “**MC_State_observer_param.h**”

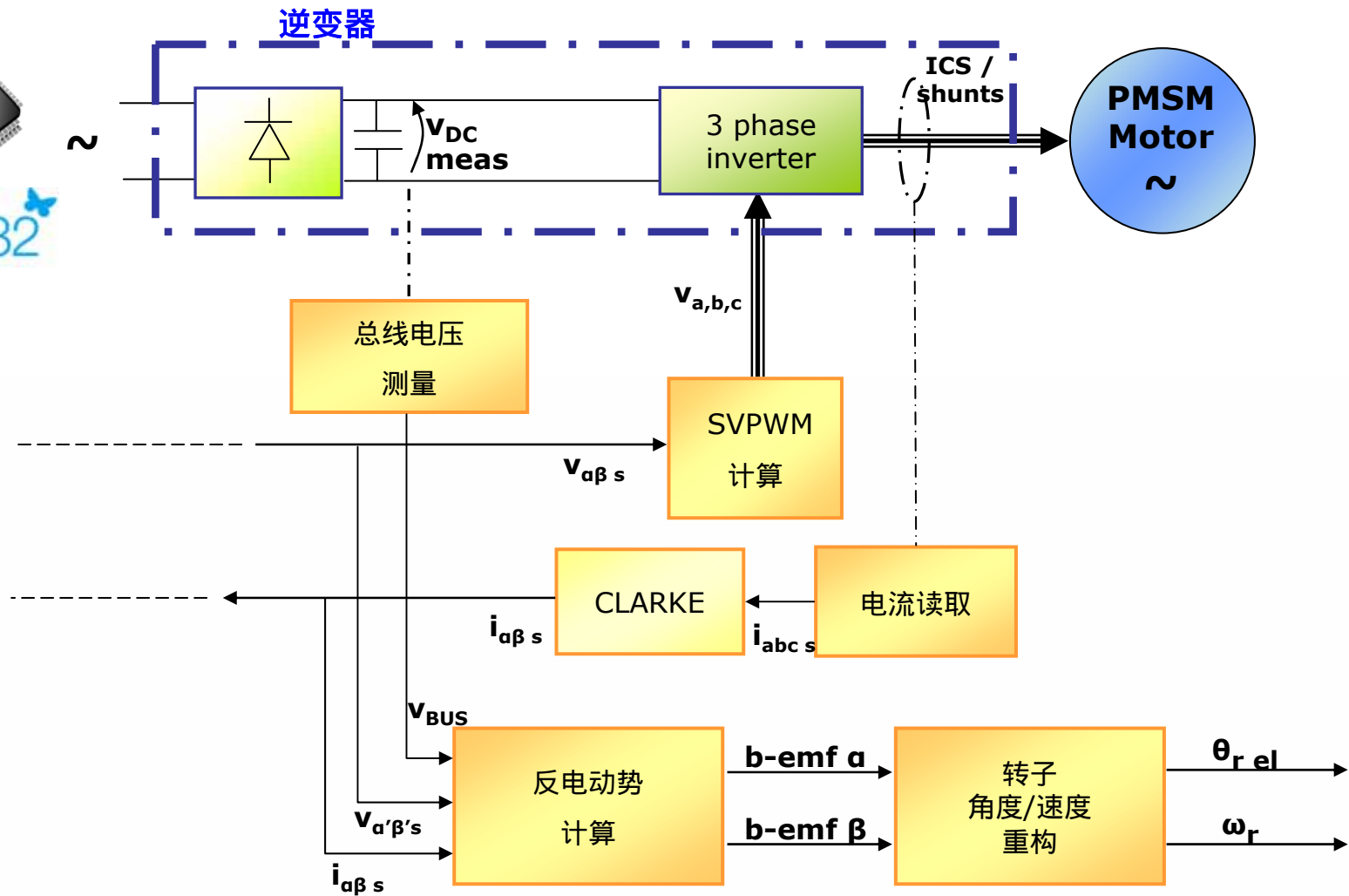
软件库结构及其它特性

-  故障管理
-  刹车
-  DAC功能

FOC 框图 (速度控制)



无传感器方案框图

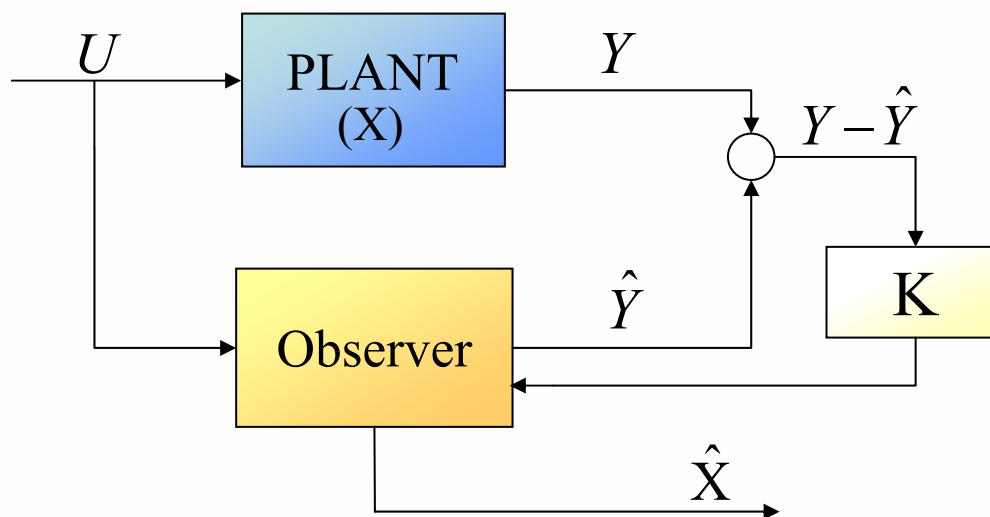


反电动势计算：观测器




根据控制理论，如果一个系统能够完全通过其检测到的输出值来重构其系统状态，则认为该系统是**可观测的**；

状态观测器根据所观测系统的输入及输出值估计其内部状态。



输入/内部状态/输出 的数学表达 1/3


 把SM-PMSM马达的电压方程变换到 $\alpha\beta$ 坐标系，并把a轴定位在 α 轴上（Clark变换）：

$$\begin{cases} v_{\alpha} = r_s i_{\alpha} + \frac{d\lambda_{\alpha}}{dt} \\ v_{\beta} = r_s i_{\beta} + \frac{d\lambda_{\beta}}{dt} \end{cases} ; \begin{cases} \lambda_{\alpha} = L_s i_{\alpha} + \sin\theta_r \cdot \Phi_m \\ \lambda_{\beta} = L_s i_{\beta} + \cos\theta_r \cdot \Phi_m \end{cases}$$

其中：

$$L_s = L_{ls} + 3L_{ms} / 2;$$

$$\theta_r = p\omega_r t$$

 把磁链方程代入电压方程，得到：

$$\begin{cases} v_{\alpha} = r_s i_{\alpha} + L_s \frac{di_{\alpha}}{dt} + \Phi_m p\omega_r \cos(p\omega_r t) \\ v_{\beta} = r_s i_{\beta} + L_s \frac{di_{\beta}}{dt} - \Phi_m p\omega_r \sin(p\omega_r t) \end{cases}$$

输入/内部状态/输出 的数学表达 2/3



得到PMSM马达的状态模型

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ y(t) = g(x(t)) \end{cases} \quad \text{为：}$$

$$\begin{cases} \frac{di_{\alpha}}{dt} = -\frac{r_s i_{\alpha}}{L_s} - \frac{\Phi_m}{L_s} p \omega_r \cos(p \omega_r t) + \frac{v_{\alpha}}{L_s} \\ \frac{di_{\beta}}{dt} = -\frac{r_s i_{\beta}}{L_s} + \frac{\Phi_m}{L_s} p \omega_r \sin(p \omega_r t) + \frac{v_{\beta}}{L_s} \\ \frac{d\theta_r}{dt} = \omega_r p \end{cases}$$

其中：状态矢量 $x = [i_{\alpha} \ i_{\beta} \ \theta_r]^t$ ，输入矢量 $u = [v_{\alpha} \ v_{\beta} \ 0]^t$ ，其输出 $y = \theta_r$ 。

很明显，该模型为非线性的；如果我们假设 L_s 为 L_q 及 L_d 的平均值，则它同样适用于**IPMSM**马达。



输入/内部状态/输出 的数学表达 3/3

引入两个新的状态变量：

$$e_{\alpha} = \Phi_m p \omega_r \cos(p \omega_r t)$$

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

那么，最后，假设机械变量相对于电变量变化缓慢得多，则：

$$\begin{cases} \frac{di_{\alpha}}{dt} = -\frac{r_s i_{\alpha}}{L_s} - \frac{e_{\alpha}}{L_s} + \frac{v_{\alpha}}{L_s} \\ \frac{di_{\beta}}{dt} = -\frac{r_s i_{\beta}}{L_s} - \frac{e_{\beta}}{L_s} + \frac{v_{\beta}}{L_s} \\ \frac{de_{\alpha}}{dt} = p \omega_r e_{\beta} \\ \frac{de_{\beta}}{dt} = -p \omega_r e_{\alpha} \end{cases}$$

PMSM马达状态方程可线性表达为：

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases}$$

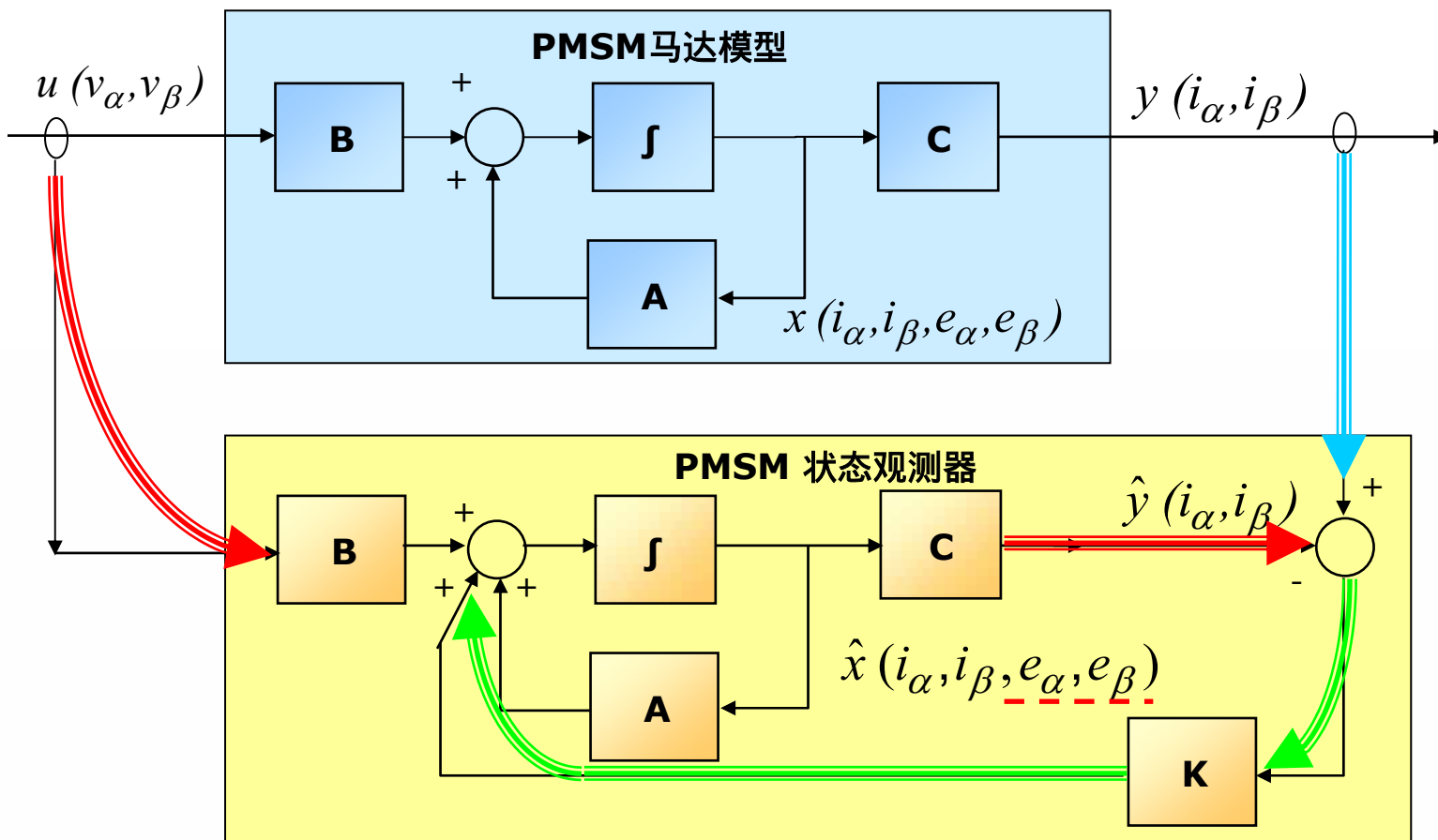
其中：

状态矢量 $x = [i_{\alpha} \ i_{\beta} \ e_{\alpha} \ e_{\beta}]^t$,

输入矢量 $u = [v_{\alpha} \ v_{\beta}]^t$,

输出变量 $y = [i_{\alpha} \ i_{\beta}]^t$

PMSM 马达的Luenberger观测器 1/2



PMSM 马达的Luenberger观测器 2/2

状态模型则表示为：

PMSM 马达模型

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases}$$

$$\begin{cases} \frac{di_{\alpha}}{dt} = -\frac{r_s i_{\alpha}}{L_s} - \frac{e_{\alpha}}{L_s} + \frac{v_{\alpha}}{L_s} \\ \frac{di_{\beta}}{dt} = -\frac{r_s i_{\beta}}{L_s} - \frac{e_{\beta}}{L_s} + \frac{v_{\beta}}{L_s} \\ \frac{de_{\alpha}}{dt} = p\omega_r e_{\beta} \\ \frac{de_{\beta}}{dt} = -p\omega_r e_{\alpha} \end{cases}$$

PMSM 状态观测器

$$\begin{cases} \hat{\dot{x}}(t) = A\hat{x}(t) + Bu(t) + K(y - \hat{y}) \\ \hat{y}(t) = C\hat{x}(t) \end{cases}$$

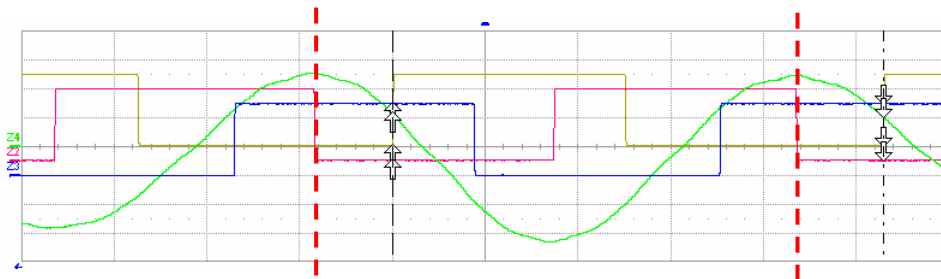
$$\begin{cases} \frac{d\hat{i}_{\alpha}}{dt} = -\frac{r_s \hat{i}_{\alpha}}{L_s} - \frac{\hat{e}_{\alpha}}{L_s} + \frac{v_{\alpha}}{L_s} + K_1(\hat{i}_{\alpha} - i_{\alpha}) \\ \frac{d\hat{i}_{\beta}}{dt} = -\frac{r_s \hat{i}_{\beta}}{L_s} - \frac{\hat{e}_{\beta}}{L_s} + \frac{v_{\beta}}{L_s} + K_1(\hat{i}_{\beta} - i_{\beta}) \\ \frac{d\hat{e}_{\alpha}}{dt} = p\omega_r \hat{e}_{\beta} + K_2(\hat{i}_{\alpha} - i_{\alpha}) \\ \frac{d\hat{e}_{\beta}}{dt} = -p\omega_r \hat{e}_{\alpha} + K_2(\hat{i}_{\beta} - i_{\beta}) \end{cases}$$

Luenberger观测器的离散化

根据上述的马达方程，引入估测电流（ $\hat{i}_\alpha, \hat{i}_\beta$ ）的反馈值并离散化（ T 为采样时间），可得：

$$\begin{cases} \hat{i}_\alpha(k+1) = \hat{i}_\alpha(k) - \frac{r_s T}{L_s} \hat{i}_\alpha(k) + K_1 T (\hat{i}_\alpha(k) - i_\alpha(k)) - \frac{T}{L_s} \hat{e}_\alpha(k) + \frac{T}{L_s} v_\alpha(k) \\ \hat{i}_\beta(k+1) = \hat{i}_\beta(k) - \frac{r_s T}{L_s} \hat{i}_\beta(k) + K_1 T (\hat{i}_\beta(k) - i_\beta(k)) - \frac{T}{L_s} \hat{e}_\beta(k) + \frac{T}{L_s} v_\beta(k) \\ \hat{e}_\alpha(k+1) = \hat{e}_\alpha(k) + K_2 T (\hat{i}_\alpha(k) - i_\alpha(k)) + p \bar{\omega}_r \hat{e}_\beta(k) T \\ \hat{e}_\beta(k+1) = \hat{e}_\beta(k) + K_2 T (\hat{i}_\beta(k) - i_\beta(k)) - p \bar{\omega}_r \hat{e}_\alpha(k) T \end{cases}$$

状态信号捕捉



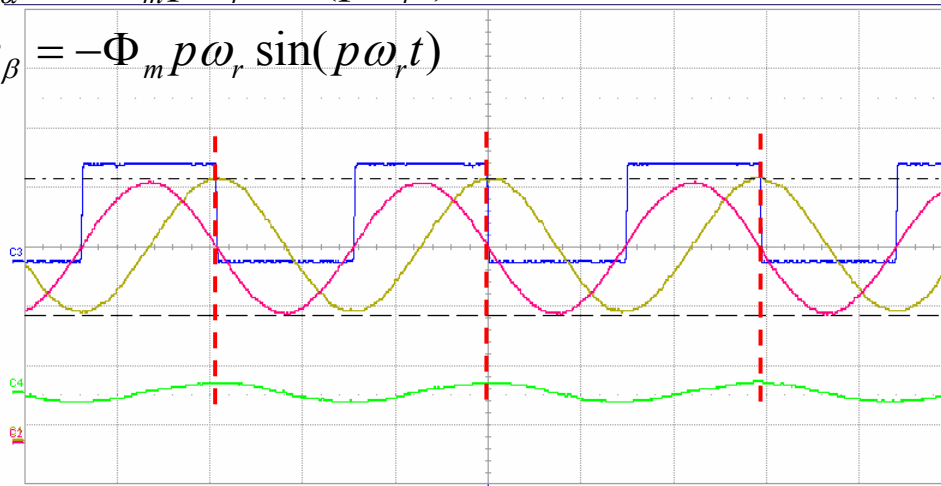
C1 C2 C3: Hall sensors
C4: back-emf phase A-0

← 正方向手动转动马达

$$v_{\alpha} = r_s i_{\alpha} + \frac{d(L_s i_{\alpha} + \Phi_m \sin \theta_r)}{dt} \longrightarrow v_{\alpha} = e_{\alpha} = \Phi_m \frac{d\theta_r}{dt} \cos \theta_r$$

$$e_{\alpha} = +\Phi_m p \omega_r \cos(p \omega_r t)$$

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



C1: observed e_{α}
C2: observed e_{β}
C3: Hall sensor C2
C4: phase current i_a



转子位置角/速度 计算 1/2

在得到反电动势的alpha及Beta分量后，可从中解析出转子的位置角；

由 e_α 及 e_β 的定义：

$$e_\alpha = \Phi_m p \omega_r \cos(p \omega_r t)$$

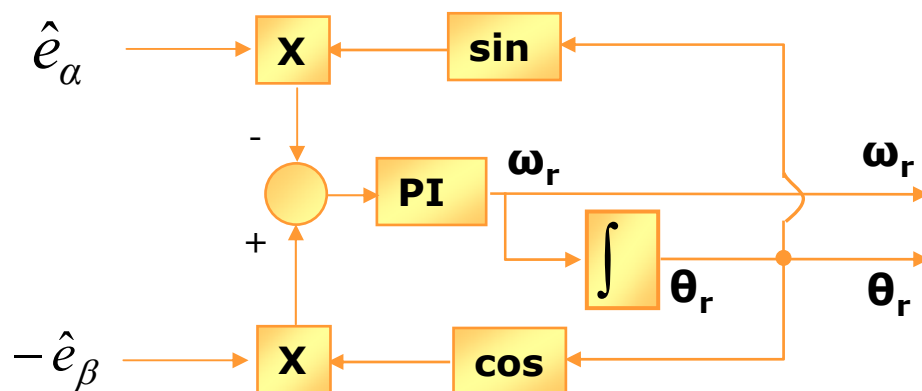
$$e_\beta = -\Phi_m p \omega_r \sin(p \omega_r t)$$

两者相除，可得到转子的位置角为：

$$\theta_r = p \omega_r t = \arctg\left(-\frac{\hat{e}_\beta}{\hat{e}_\alpha}\right)$$

这个方法是开环的，它对耦合在反电动势里的干扰非常敏感(它会使反电动势成为非正弦信号！)

转子位置角/速度 计算2/2：锁相环 (PLL)



令 $A = m p \omega_r$, 则 :

$$\hat{e}_\alpha(kT) = A \cos(\hat{\theta}_k) \quad -\hat{e}_\beta(kT) = A \sin(\hat{\theta}_k)$$

那么 :

$$-\hat{e}_\alpha(kT) \cdot \sin(\theta_{k-1}) - \hat{e}_\beta(kT) \cdot \cos(\theta_{k-1}) \cong A(\hat{\theta}_k - \theta_{k-1})$$

PI闭环作用：调节 ω_r 以保持其输入为0 (电角度将领先反电动势一个采样时间)

PLL 闭环系数计算

经过验证，Kp及Ki定义为：

$$P \text{ gain} = \frac{532 \cdot \text{Maximum electrical frequency}}{\text{Observer sampling freq}}$$

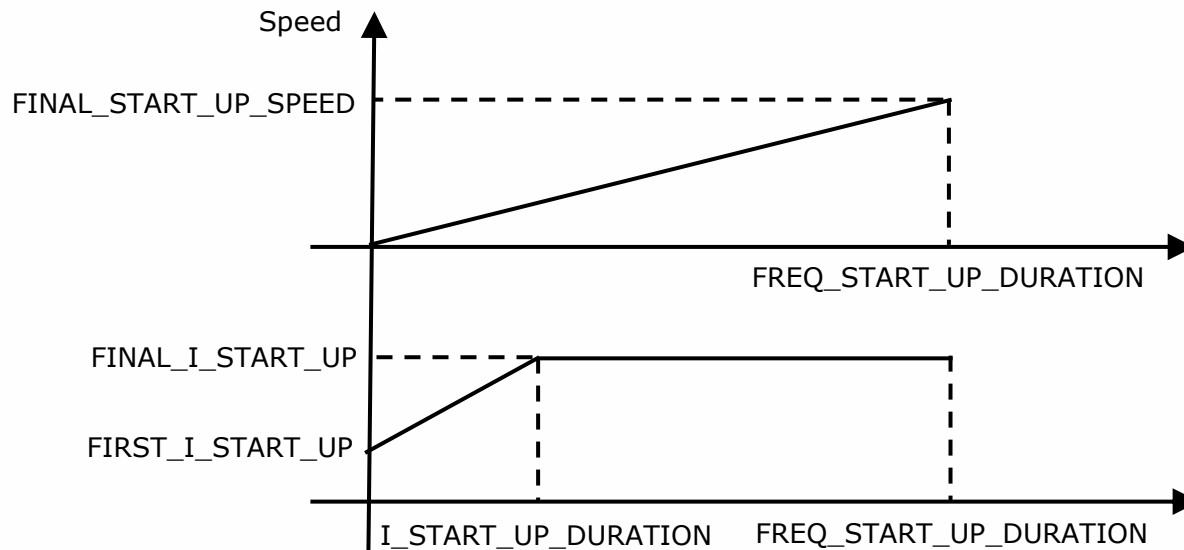
$$I \text{ gain} = \frac{1506742 \cdot \text{Maximum electrical frequency}}{(\text{Observer sampling freq})^2}$$

没有必要去修改它们的值，但是如果确实有必要修改，则必须注意：


为了增加带宽(此时，对速度变化有更好的动态响应，但 ω_r 干扰增大)，应比例地增大Kp及Ki。

起动策略

- 在“ramp-up”期间，在定子中加入幅值及频率都受控的电流；
- 特别地，电流的幅值及频率在“ramp-up”期间的变化如下图：




起动成功的判据 1/2

- 
- ❏ 为了从起动模式(在软件中：state = START的状态)切换到FOC闭环控制(state = RUN的状态)，观测器和PLL必须都收敛；
 - ❏ 根据所观测的速度FIFO，周期性地计算下列两个统计指数（该周期在软件中定义为PID_SPEED_SAMPLING_TIME）：

$$\mu = \frac{\sum_{i=1}^{64} x_i}{64} \quad \sigma^2 = \frac{\sum_{i=1}^{64} (x_i - \mu)^2}{64}$$

起动成功的判据 2/2

- 
- 如果连续多次（软件中定义为：NB_CONSECUTIVE_TESTS）下式成立，则可认为该算法收敛：

$$\sigma^2 \leq \mu^2 \cdot \text{VARIANCE_THRESHOLD}$$

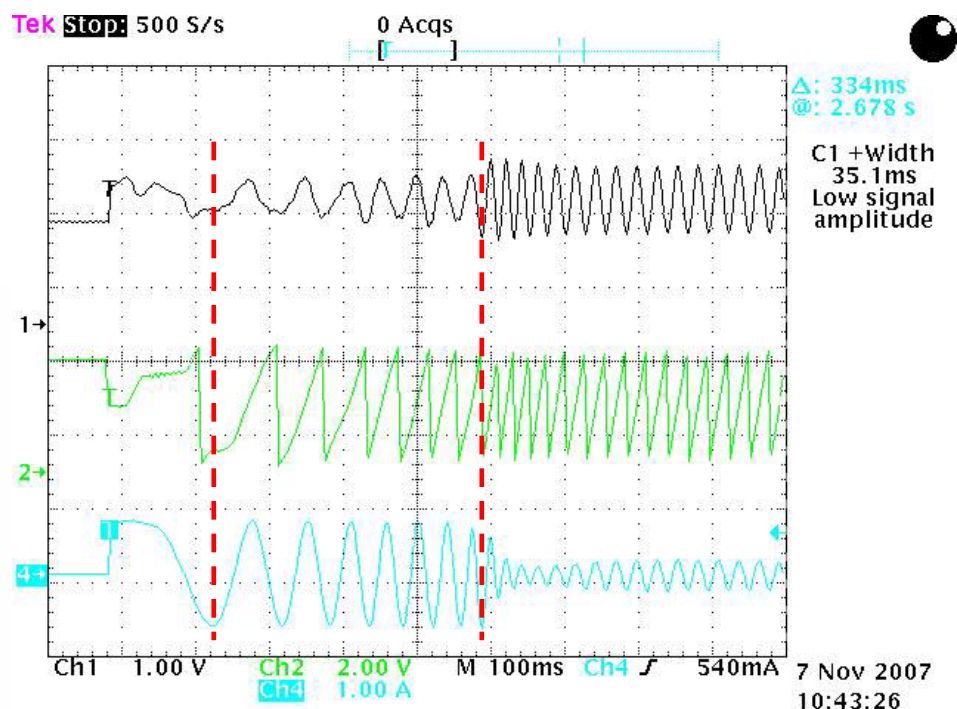
$$0.8 \cdot \omega_{\text{forced}} \leq \mu \leq \omega_{\text{forced}}$$

$$\mu > \text{MINIMUM_SPEED_RPM}$$

- 如果上式在时间START_UP_DURATION前成立，则软件中主状态机从“START”切换到“RUN”；否则，主状态机切换到“FAULT”(“Start-up failed”)



起动实例

下图为Shinano PMSM马达的起动波形：



图中：转子位置角在150ms内收敛，起动判据在500ms内满足。

转子堵转检测

- 
- 
- ❑ 转子堵转检测基于统计理论，因此它并不是100%安全的；
 - ❑ 当反电动势信号（它为PLL的输入）较低时，速度测量的误差会增大，一般这发生在马达低速的情况下；
 - ❑ 当“state=RUN”时，如果下式连续多次（软件中定义为：RELIABILITY_HYSTERESYS）满足：
$$\sigma^2 \geq \mu^2 \cdot \text{VARIANCE_THRESHOLD}$$
则“state”切换为“FAULT”（“Error on speed feedback”）
 - ❑ 如果转子在堵转位置呈正弦抖动，则观测器无法检测出转子的堵转

MC_State_Observer_Param.h

- 如果在`stm32f10x_MCconf.h`中定义了“OBSERVER_GAIN_TUNING”或“NO_SPEED_SENSORS”，则该头文件中的参数必须填写：
- MAX_CURRENT**：电流幅值(单位：[A]，为峰值)，其在库文件中的数字表达为S16MAX；
- BUS_ADC_CONV_RATIO**：直流总线电压的分压值；
- K1,K2**：Luenberger观测器闭环参数；
- F1,F2**：该系数是为了扩大观测器方程系数，以完成观测器的软件实现；
- PLL_KP_GAIN, PLL_KI_GAIN**：PLL闭环的比例及积分系数；
- VARIANCE_THRESHOLD**：速度的最大变化率(百分比值)，当大于该值时，速度被认为是不可靠的；
- RELIABILITY_HYSTERESYS**：速度检测失败的连续次数，当大于该值时，“state”切换为“FAULT”；
- FREQ_START_UP_DURATION, FINAL_START_UP_SPEED, FIRST_I_STARTUP, FINAL_I_STARTUP, I_START_UP_DURATION**：马达起动参数(详见马达起动章节)；
- NB_CONSECUTIVE_TESTS**：马达起动判据连续满足的次数，一旦满足，马达从“start-up”切换到FOC闭环；
- MINIMUM_SPEED_RPM**：马达起动时，马达必须达到的最小速度(rpm)

Plan

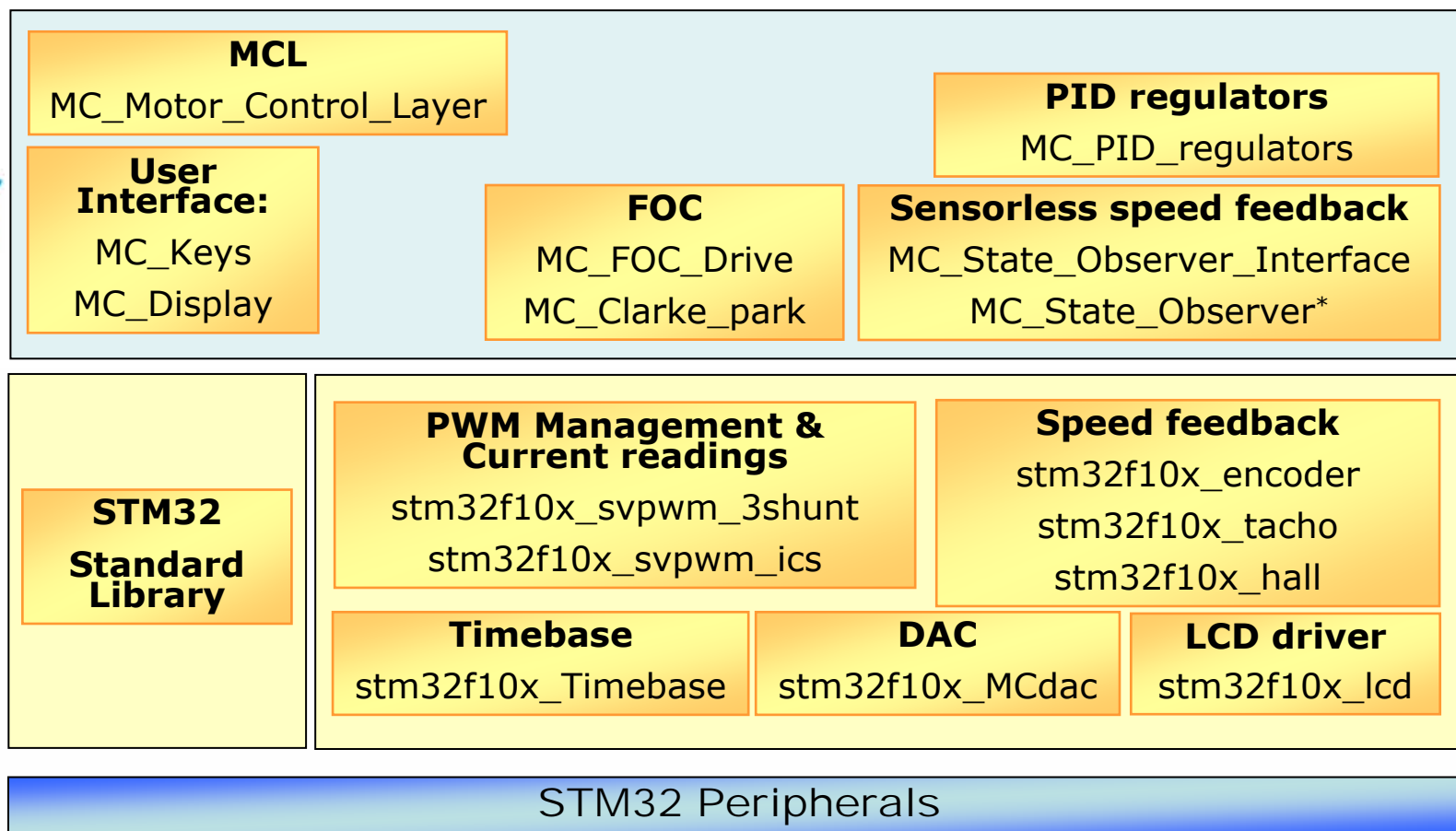
无传感器转子位置检测：

- 框图
- 观测器理论
- 应用到PMSM马达系统
- PLL
- 起动策略
- 转子堵转检测
- 头文件“MC_State_observer_param.h”

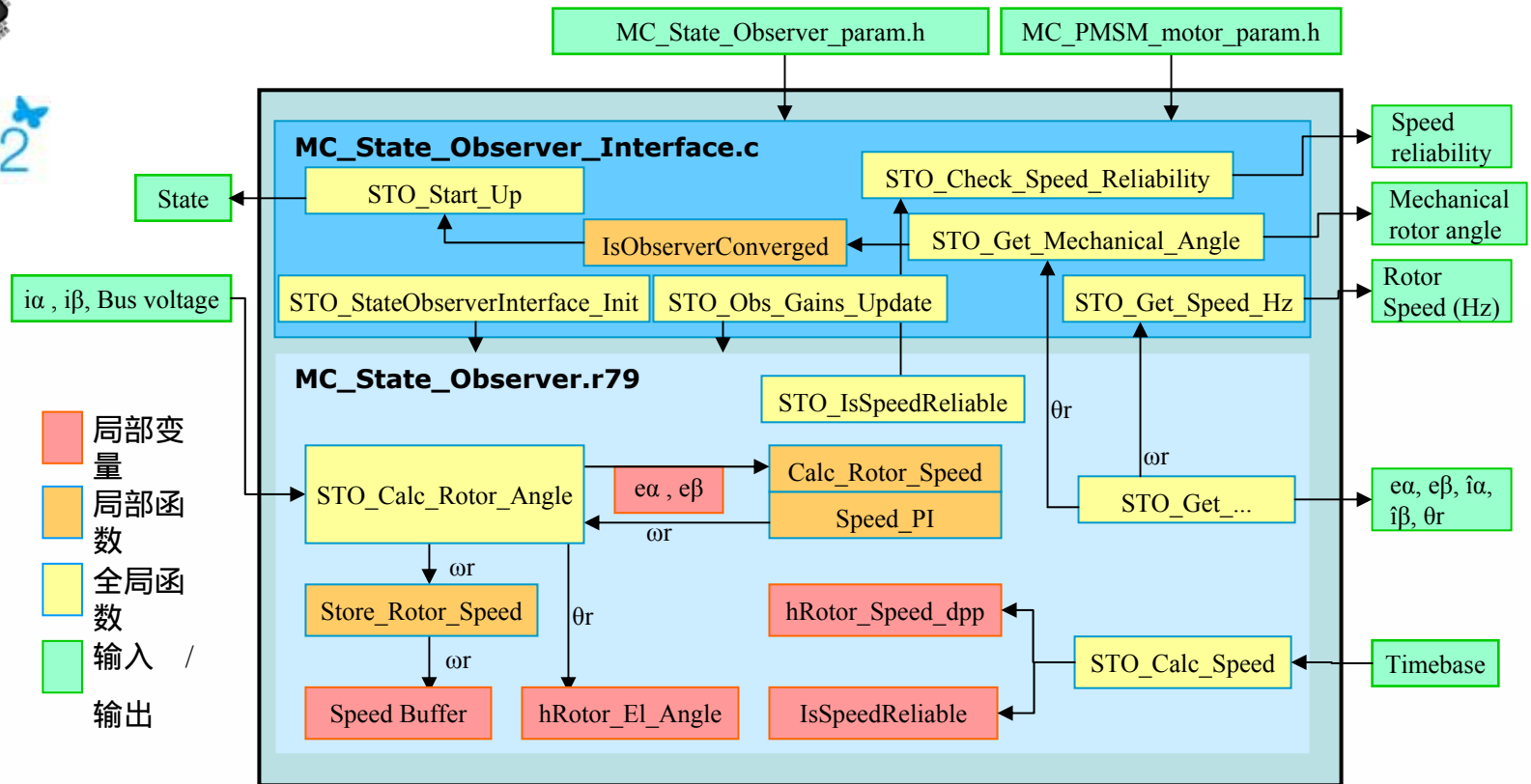
软件库结构及其它特性

- 故障管理
- 刹车
- DAC功能**

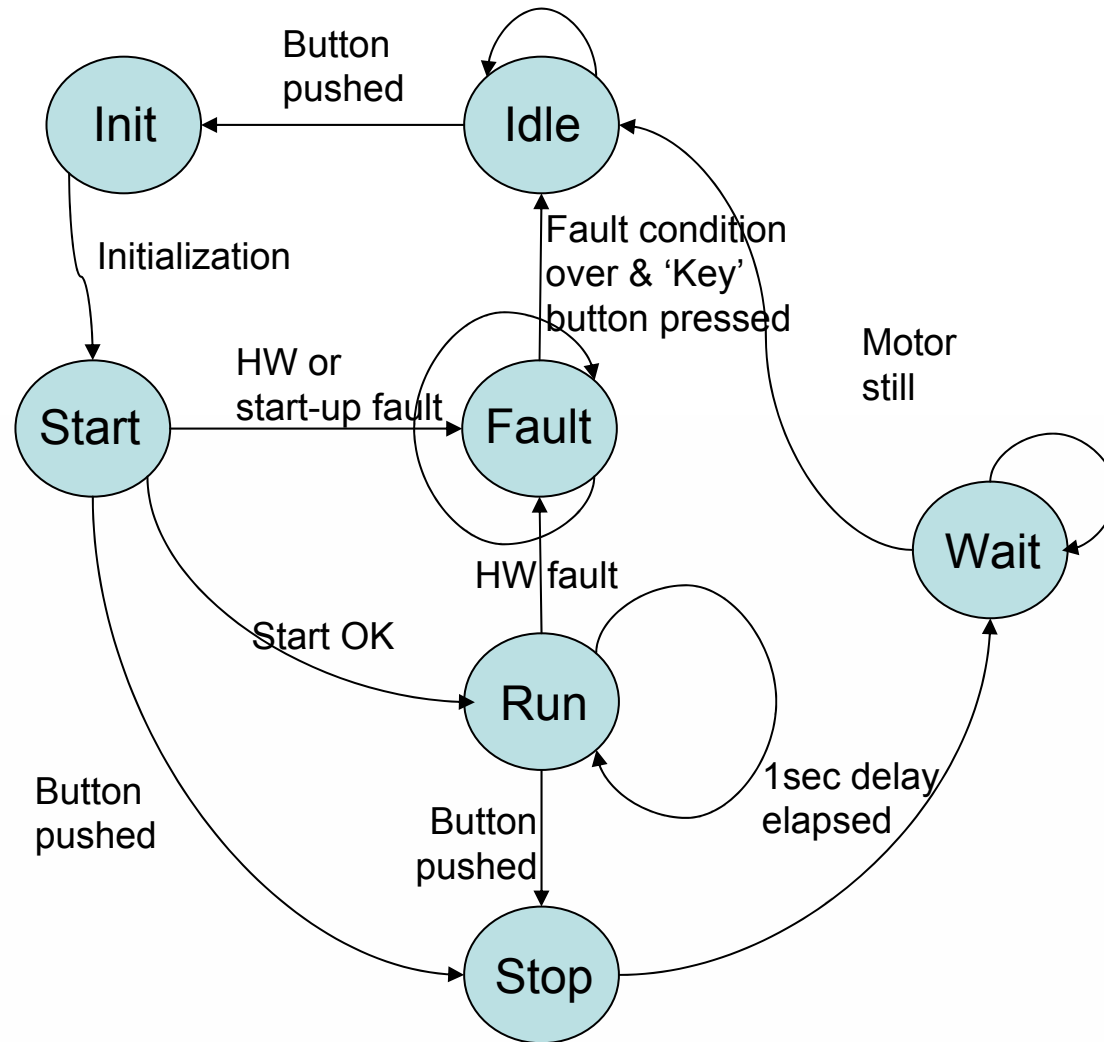
软件库结构



无传感器转子速度/位置检测 软件结构



主状态机



刹车策略



除非你的逆变器拥有再生发电能力，否则其直流母线上的电容在以下情况下会被反向充电：

- 在六个开关管都关闭的情况下，马达的运行速度大于额定值；
- 刹车

有一些方法可用来消耗马达的动能，其中包括：

- 短路马达绕组
- 采用刹车电阻

短路马达绕组



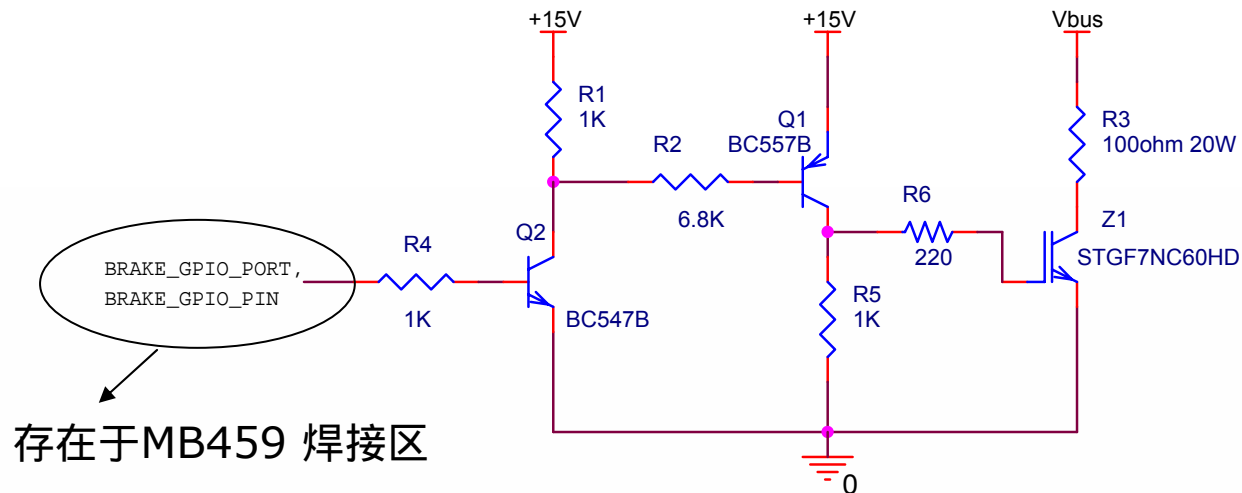
- 当前的库文件不支持该方法，但它很容易在STM32上实现(配置TIM1，使得：当MOE位被清零时所有的下桥臂打开)；
- 当马达绕组被短路时，马达电流不经过电容而仅在绕组中流动；
- 稳态电流“ I_{SC} ”被马达阻抗限定：

$$I_{SC} = \left| \frac{K_e \cdot \omega_r}{R_s + j\omega_r L_s} \right| = \frac{K_e}{\sqrt{\frac{R_s^2}{\omega_r^2} + L_s^2}}$$

- 注意：可能会产生很高的电流，因此必须适当的设定L6386的限流值，以避免在该情况下关闭下桥臂

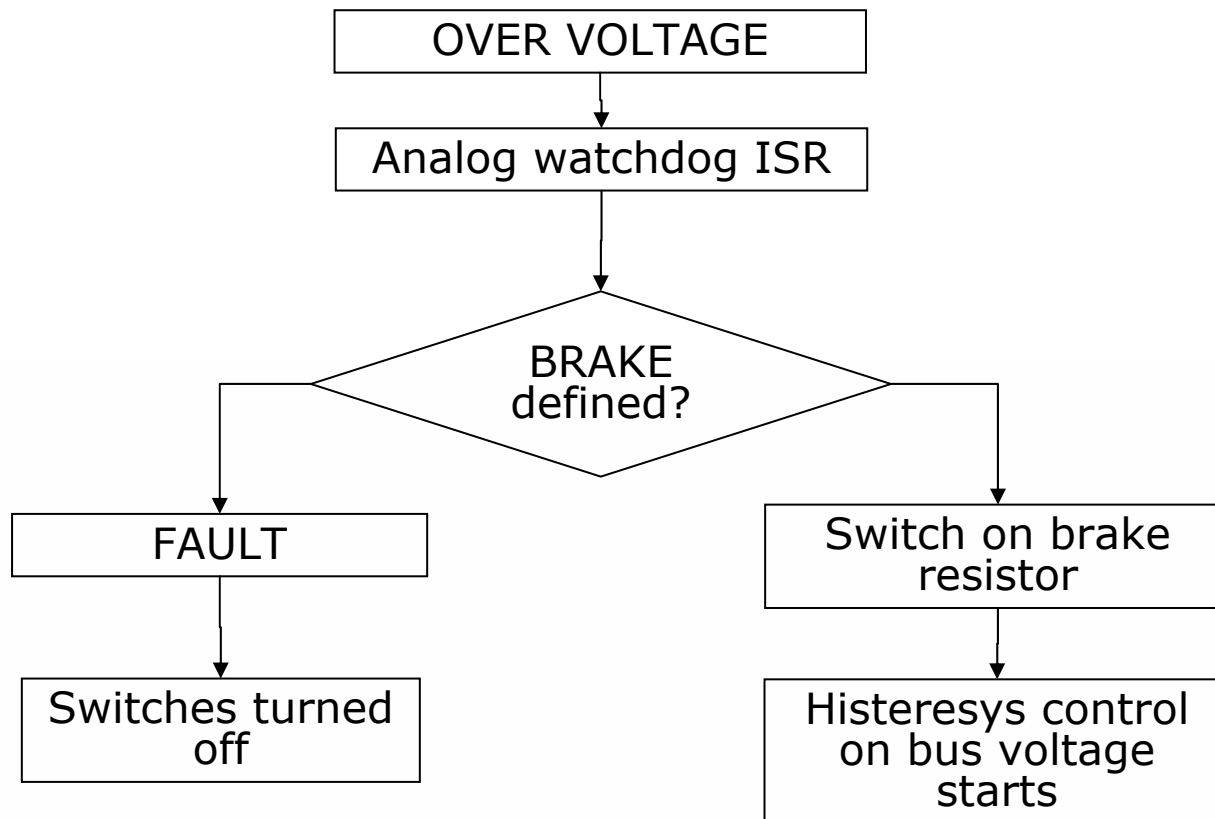
刹车电阻

- DC bus的电流流过该电阻；
- 当使用 MB459时，需要焊接以下电路：





在高压情况下，若有弱磁运行，则必须使用该电路！

过压管理



当库文件定义了“**BRAKE**”时，无过压
“**FAULT**”产生

故障管理

- 
- 
- ❏ 过压故障：DC电压超过OVERVOLTAGE_THRESHOLD_V (*MC_Control_param.h*)，但当定义“BRAKE”时，该故障不产生；
 - ❏ 欠压故障：DC电压低于UNDERVOLTAGE_THRESHOLD_V (*MC_Control_param.h*)；
 - ❏ 过温故障：高压板温度超过NTC_THRESHOLD_C，其比较滞缓为NTC_HYSTERIS_C (*MC_Control_param.h*)；
 - ❏ 起动失败故障： 仅在无传感器方案中实现；
 - ❏ 速度反馈故障：
 - ❏ 过流故障：

中断优先级

▣ 下述列出了库文件中所有中断的优先级：

▣ **TIM1 Break interrupt:**

BRK_PRE_EMPTION_PRIORITY 0

BRK_SUB_PRIORITY 0

stm32f10x_svpwm_3shunt.c or *stm32f10x_svpwm_ics.c*

▣ **ADC Interrupt** (JEOC or AWD):

ADC_PRE_EMPTION_PRIORITY 1

ADC_SUB_PRIORITY 0

stm32f10x_svpwm_3shunt.c or *stm32f10x_svpwm_ics.c*

▣ **TIM1 Update Interrupt:**

TIM1_UP_PRE_EMPTION_PRIORITY 1

TIM1_UP_SUB_PRIORITY 0

stm32f10x_svpwm_3shunt.c

▣ **TIMx Interrupt:**

TIMx_PRE_EMPTION_PRIORITY 2

TIMx_SUB_PRIORITY 0

stm32f10x_encoder.c or *stm32f10x_hall.c*

▣ **SysTick Interrupt:**

SYSTICK_PRE_EMPTION_PRIORITY 3

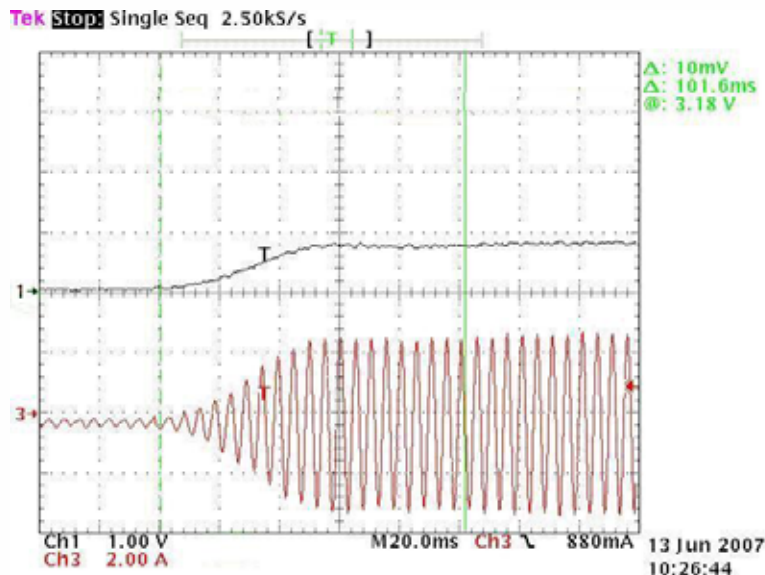
SYSTICK_SUB_PRIORITY 0

stm32f10x_Timebase.c

DAC 功能



- 使用TIM3的两个输出比较通道实现；
- 通过注释掉一行代码就可以禁止该功能；
- 可以同时监测两个软件变量，而这两个软件变量可以实时地使用专门的菜单选择，如下图：



STM32 Motor Control
PMSM FOC ver 0.2

Signal on PB0

Iqref

Signal on PB1

Ia

←→ Move ↑↓ Change

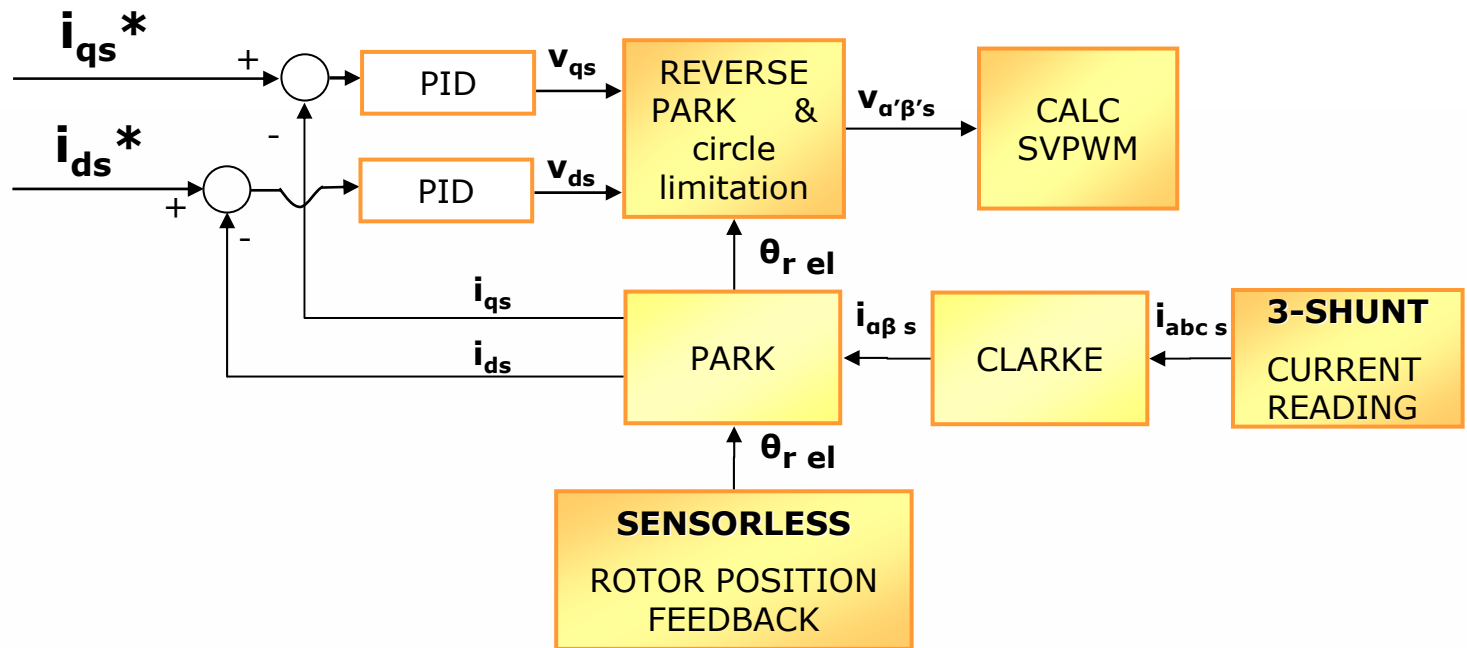
控制性能介绍



- ▣ 正弦永磁马达的最低速度可达到最大速度的5% ;
- ▣ 对绕组电阻 R_s 的不确定性或变化低敏感 ;
- ▣ 对绕组电感 L_s 的值中度敏感 ;
- ▣ FOC算法总的执行时间(无传感器/3电阻法)大约24 μ s(代码为速度优化) ;
- ▣ PMSM FOC库全部代码的长度大约为25kB(无传感器模式), 如果去掉用户界面代码则大约为13.5kB (代码为速度优化)

FOC算法的执行时间

- 由于Cortex™-M3强劲的计算能力，在无传感器/3电阻法模式下，FOC算法总的执行时间为**24.5usec**(EWARM ver4.42，速度优化)
- 使用最新的EWARM ver.5.11，其执行时间可减少至**20.5usec**



CPU load @ 10kHz sampling time < 30%

FOC 软件库代码长度



- 由于Thumb2指令组的代码密度非常高，PMSM FOC软件库的总长度为25Kb(sensorless)；
- 若排除LCD及Joystick管理，则代码长度可减少至：

13.5Kb

(EWARM 4.42, code optimized for **speed**)