

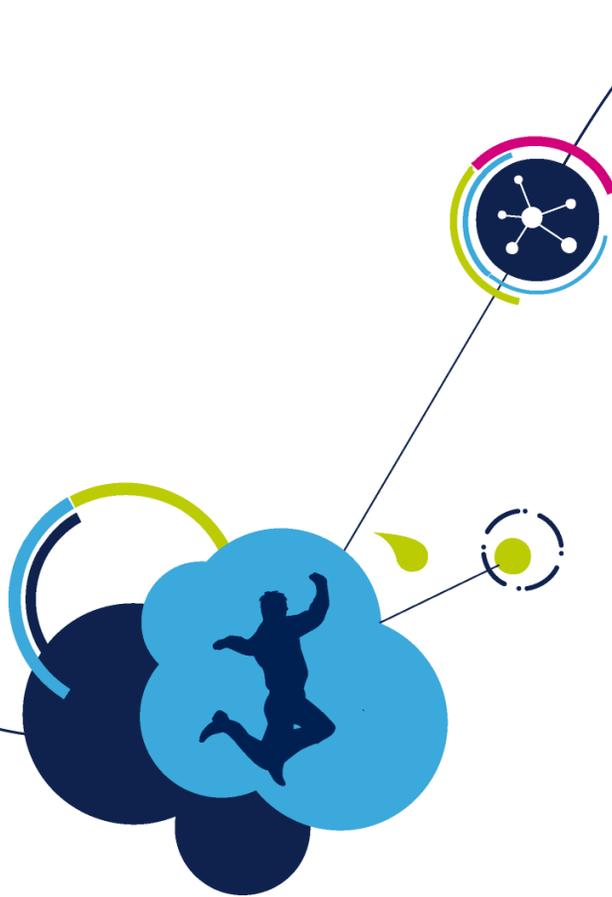
# ST MC SDK 5.x 永磁同步电动机矢量控制理论基础

STM32电动机控制应用系列讲座之二



- 永磁同步电动机转动原理
- 矢量变换
- 矢量控制
- 空间电压矢量PWM(SVPWM)
- 特别算法介绍
  - MTPA
  - 弱磁控制
  - 电流前馈控制

# 永磁同步电动机转动原理

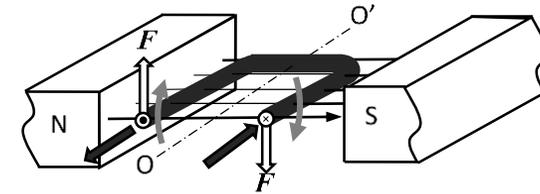
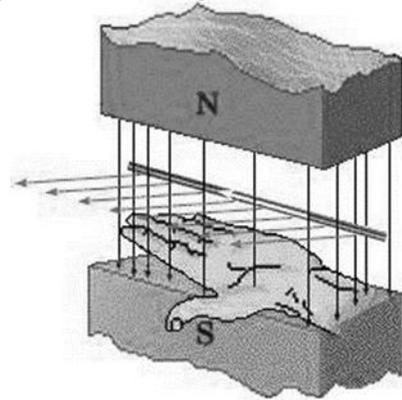


- 永磁同步电动机转动原理
- 矢量变换
- 矢量控制
- 空间电压矢量PWM(SVPWM)
- 特别算法介绍
  - MTPA
  - 弱磁控制
  - 电流前馈控制

# 永磁同步电机矢量控制原理(1/2)

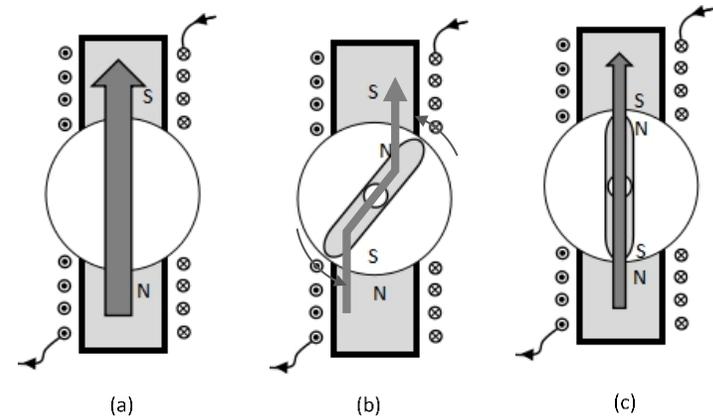
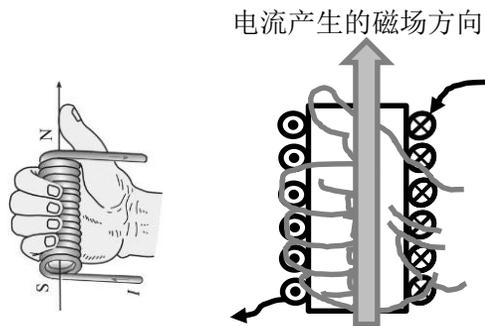
永磁力矩

Permanent magnetic torque

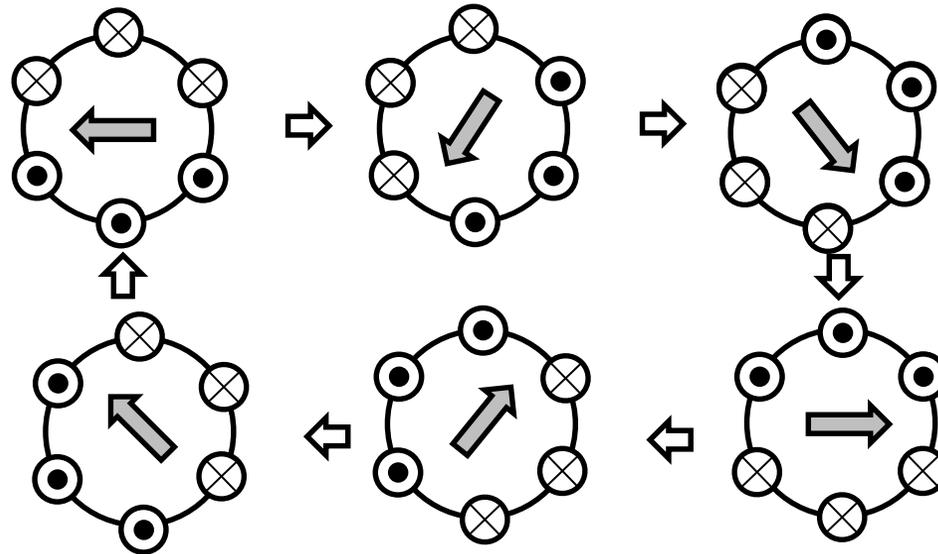
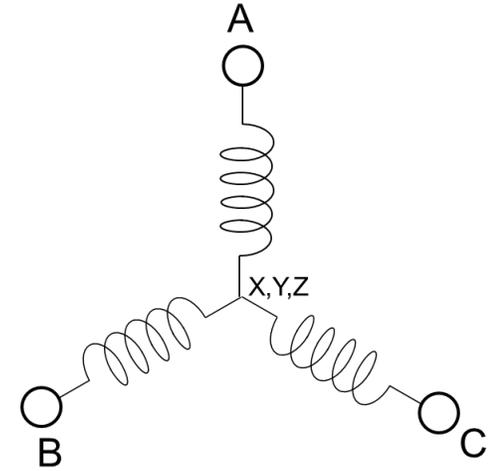
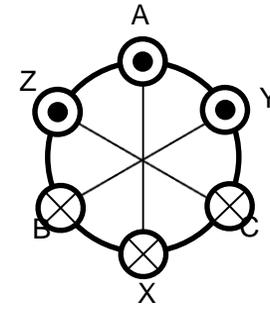
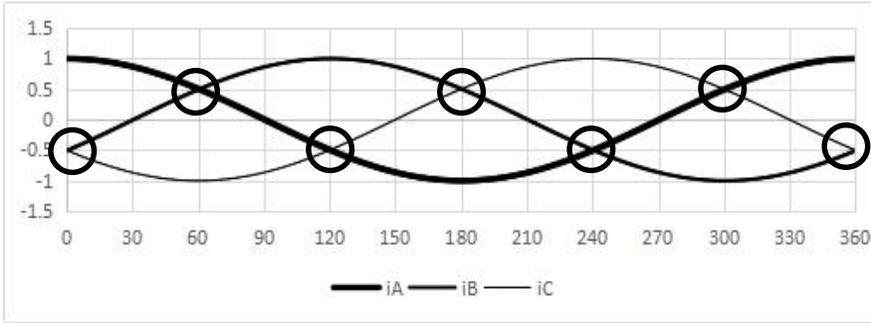


磁阻力矩

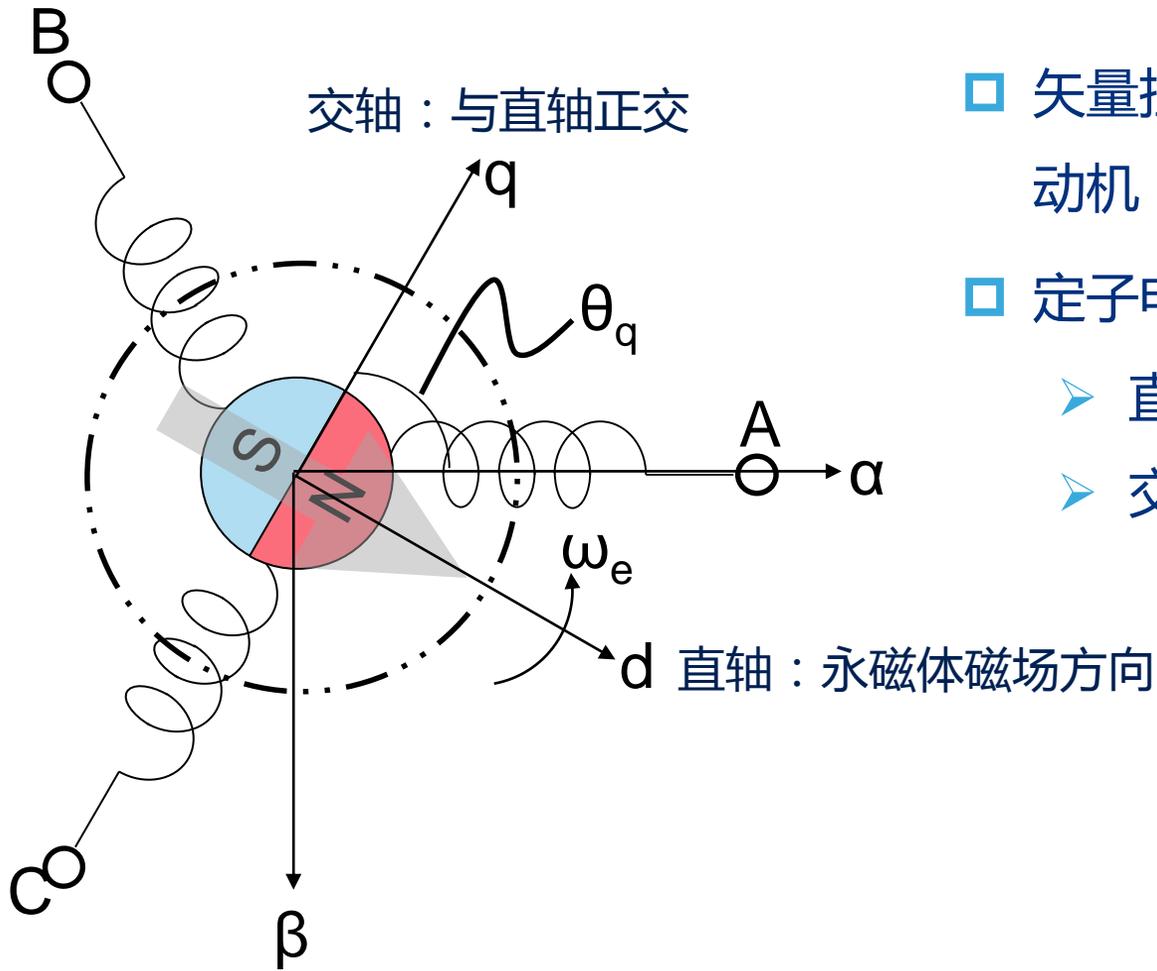
Reluctance torque



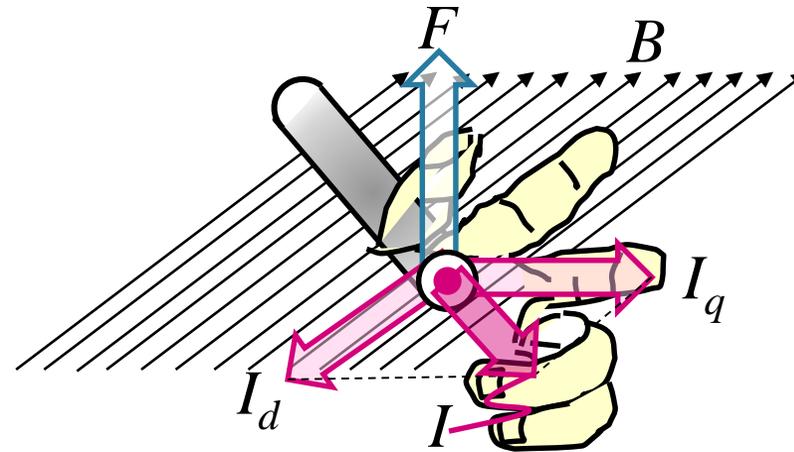
# 永磁同步电机矢量控制原理(2/2)



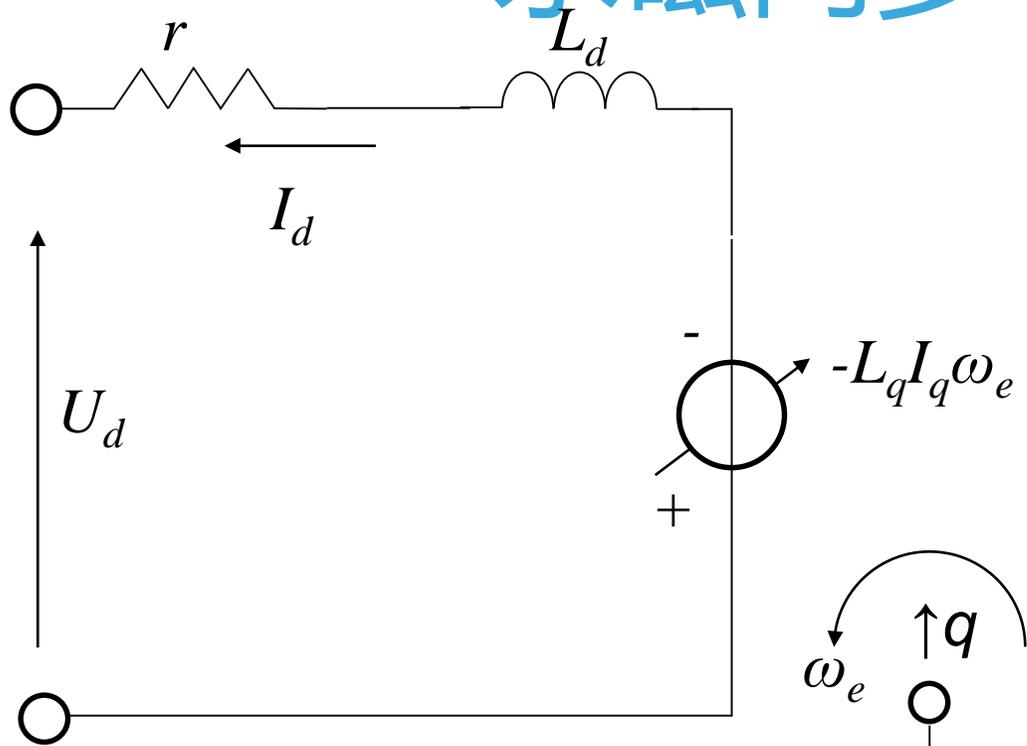
# 永磁同步电机矢量控制原理



- 矢量控制的基本思想是将交流电动机等效为他励直流电动机，转矩和励磁分别做独立的控制。
- 定子电流被分解成：
  - 直轴电流 $I_d$ ：励磁电流
  - 交轴电流 $I_q$ ：转矩电流

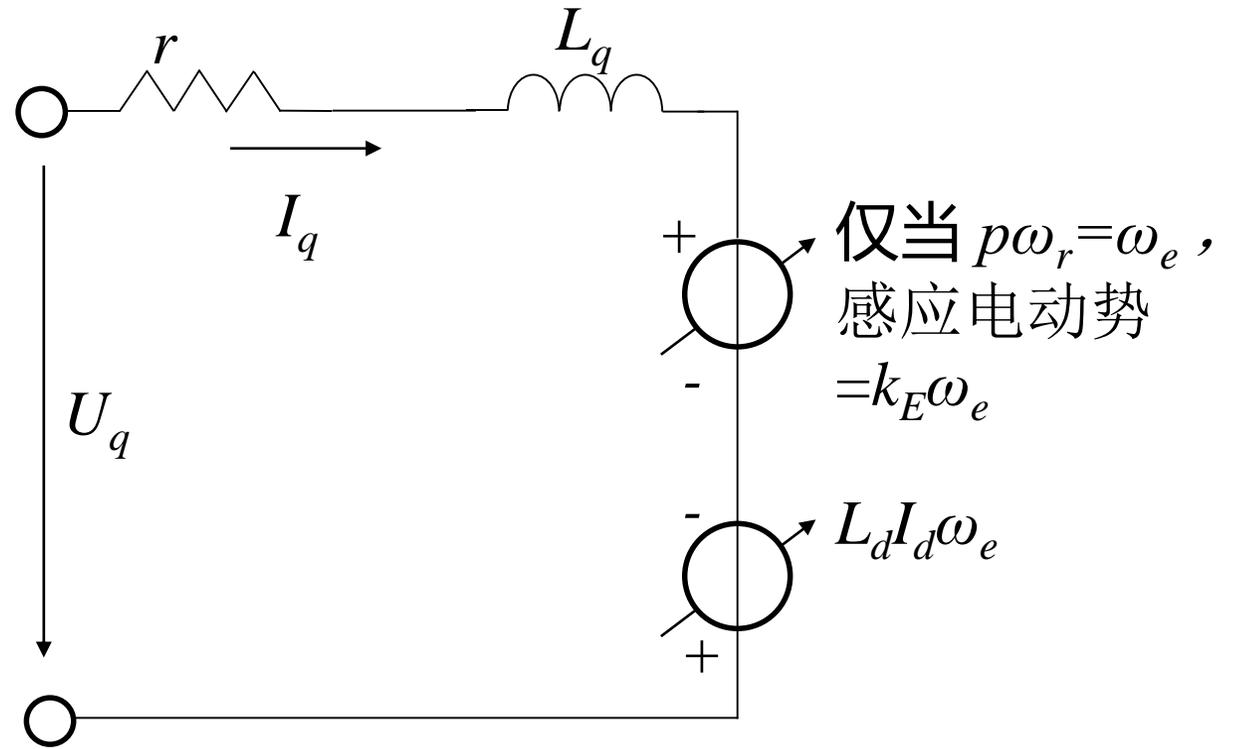


# 永磁同步电动机的数学模型(1/2)

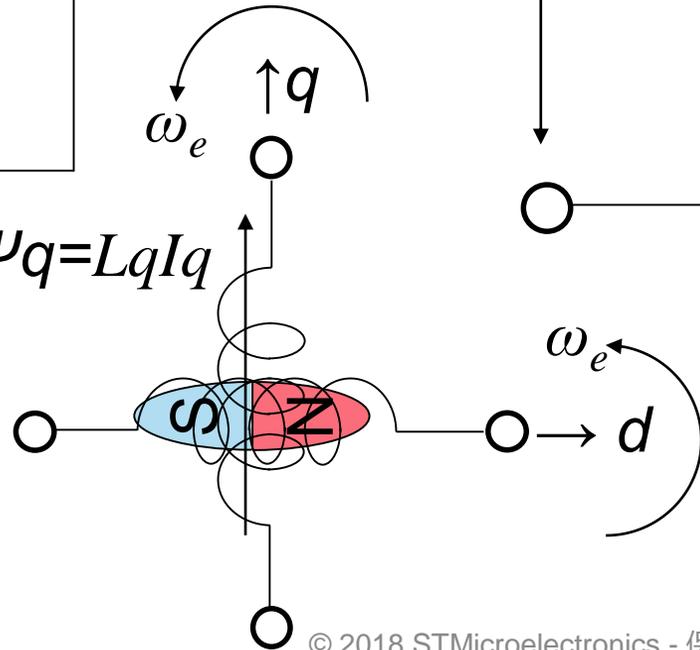


d轴线圈等效电路

$$\Psi_q = L_q I_q$$



q轴线圈等效电路

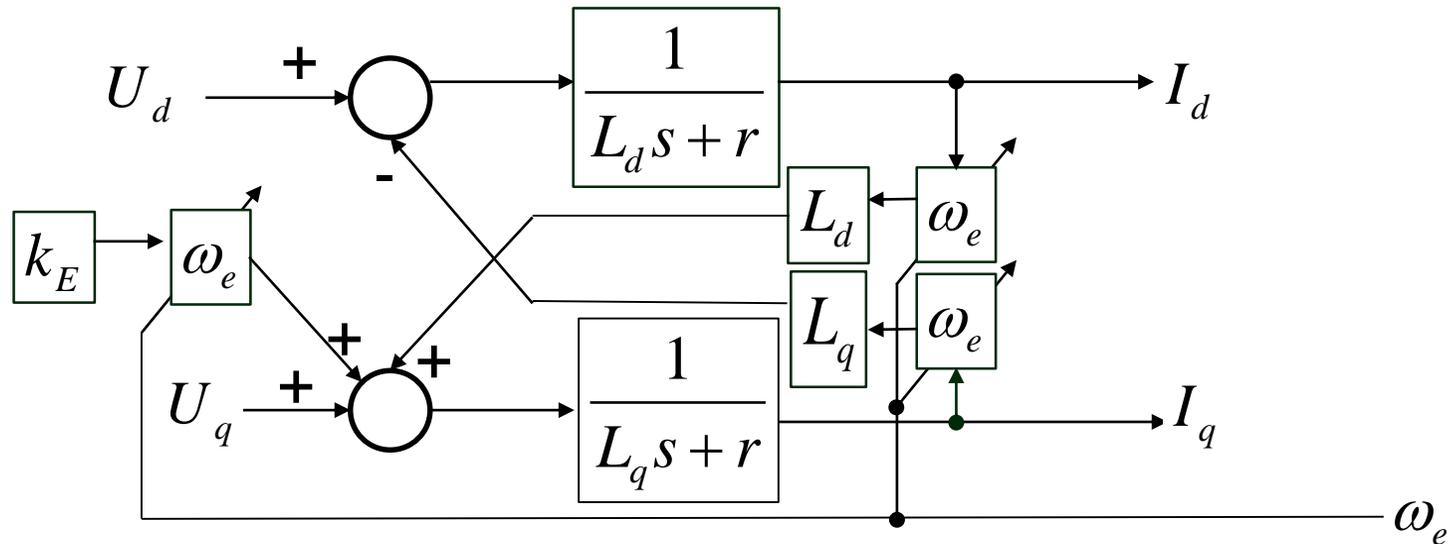


# 永磁同步电动机的数学模型(2/2)

## 永磁同步电动机的电压方程

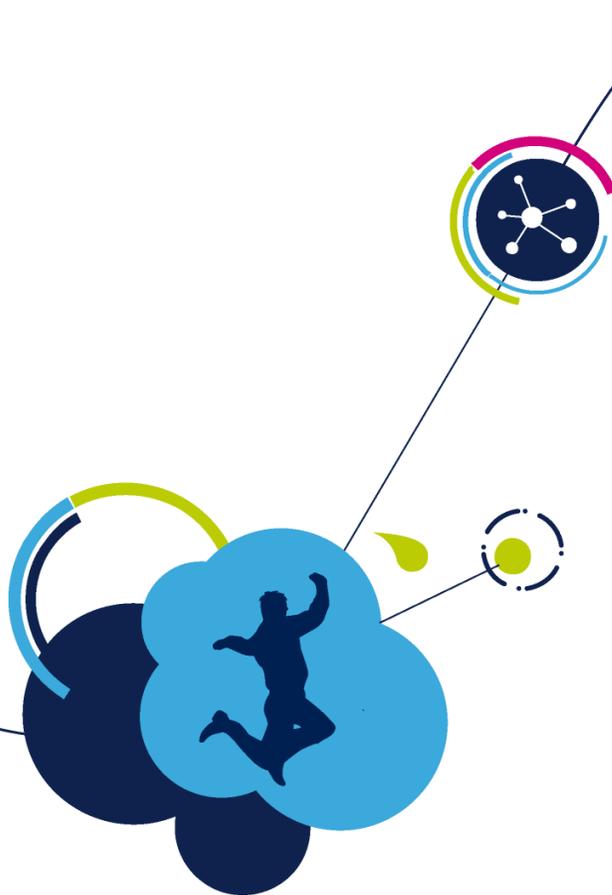
$$\begin{bmatrix} U_d \\ U_q \end{bmatrix} = \begin{bmatrix} r + \frac{d}{dt} L_d & -L_q \omega_e \\ L_d \omega_e & r + \frac{d}{dt} L_q \end{bmatrix} \begin{bmatrix} I_d \\ I_q \end{bmatrix} + \begin{bmatrix} 0 \\ k_E \omega_e \end{bmatrix}$$

## 永磁同步电动机的电气模型



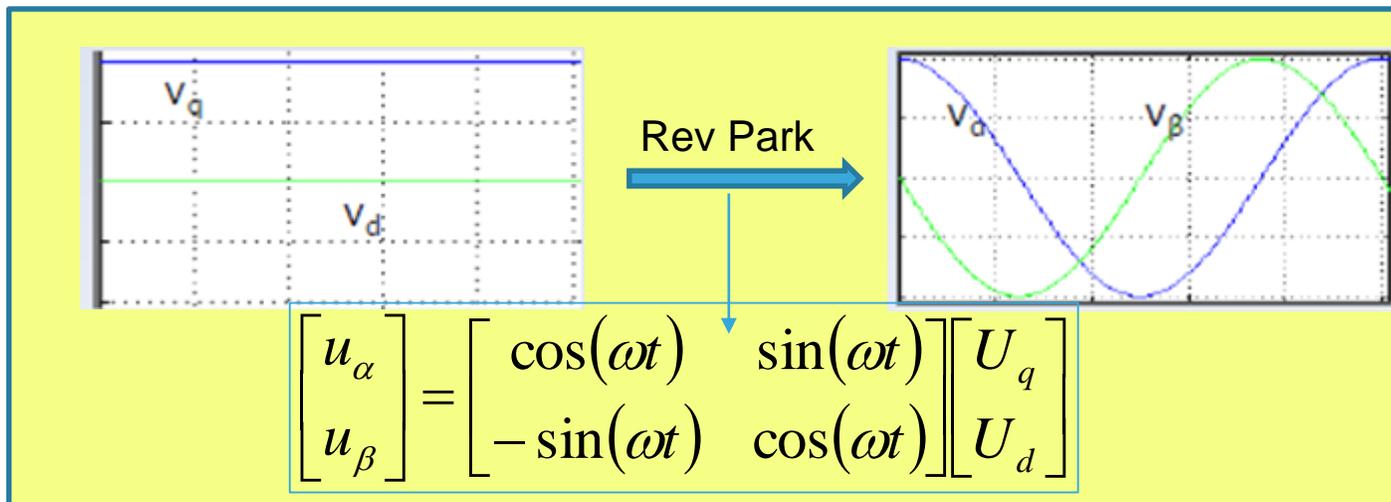
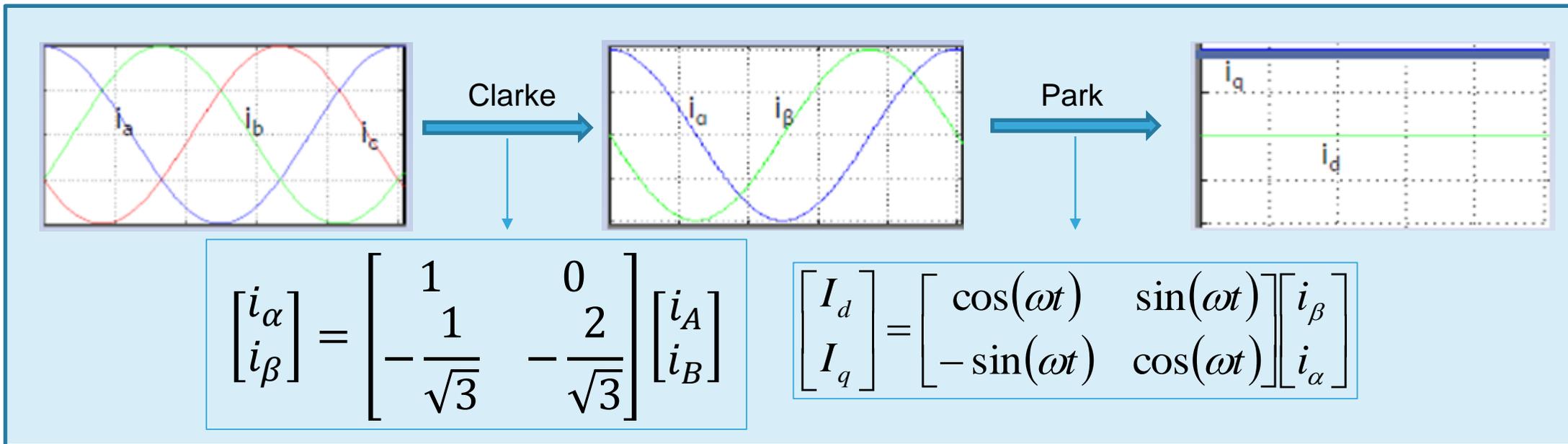


# 矢量变换

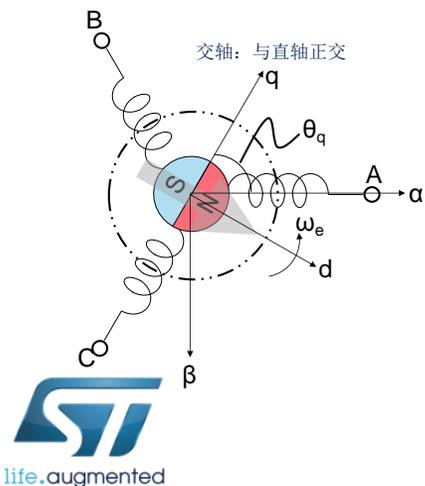


- 永磁同步电动机转动原理
- 矢量变换
- 矢量控制
- 空间电压矢量PWM(SVPWM)
- 特别算法介绍
  - MTPA
  - 弱磁控制
  - 电流前馈控制

# 矢量变换公式以及示意图



$$p_3 = \frac{3}{2} p_2$$



# ST MC SDK5.x实现矢量变换的固件源程序

➤ 具体文件夹如下：

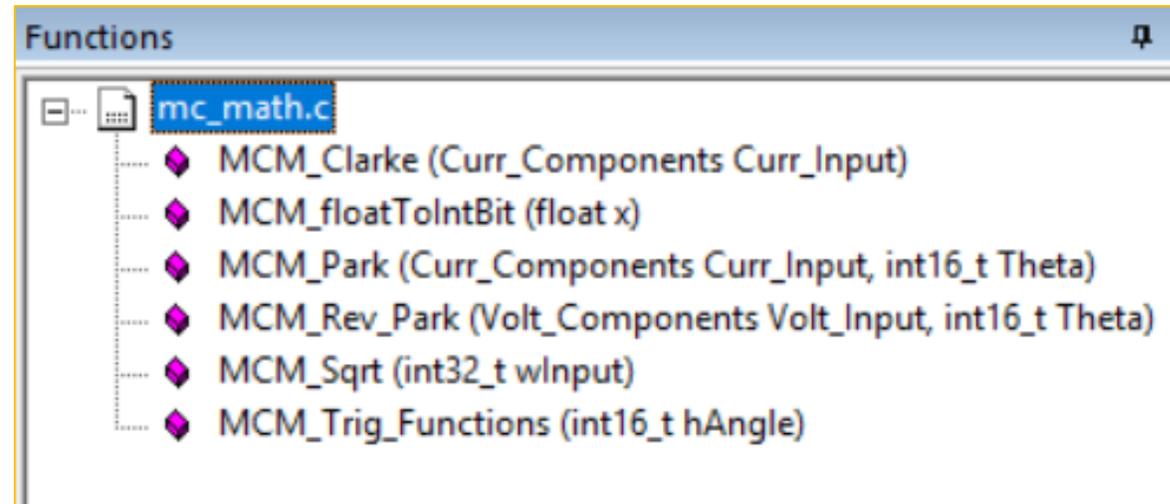
- ✓ xxx\MCSDK\_v5.2.0\MotorControl\MCSDK\MCLib\Any\Src

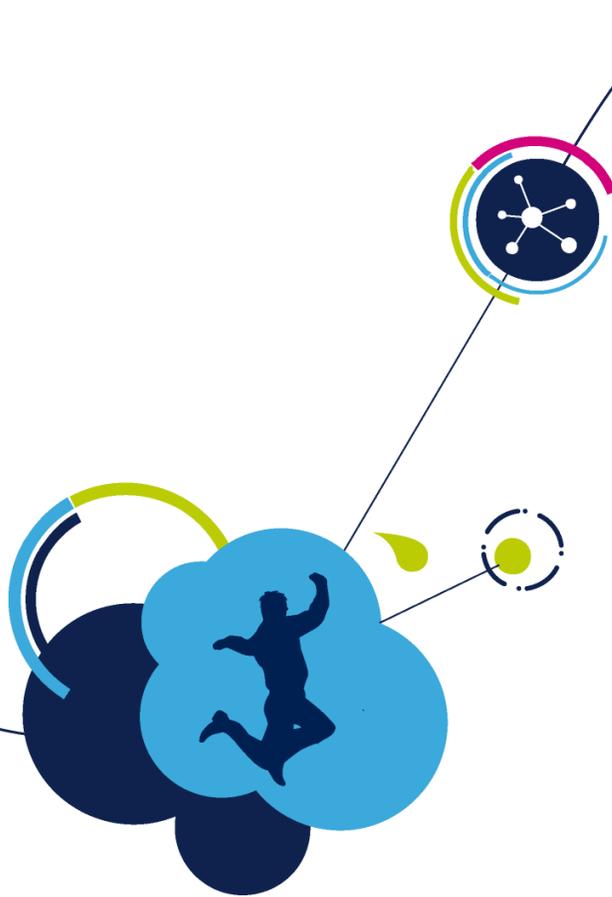
➤ 文件名如下：

- ✓ mc\_math.c

➤ 函数名：

- ✓ MCM\_Clarke
- ✓ MCM\_Park
- ✓ MCM\_Rev\_Park

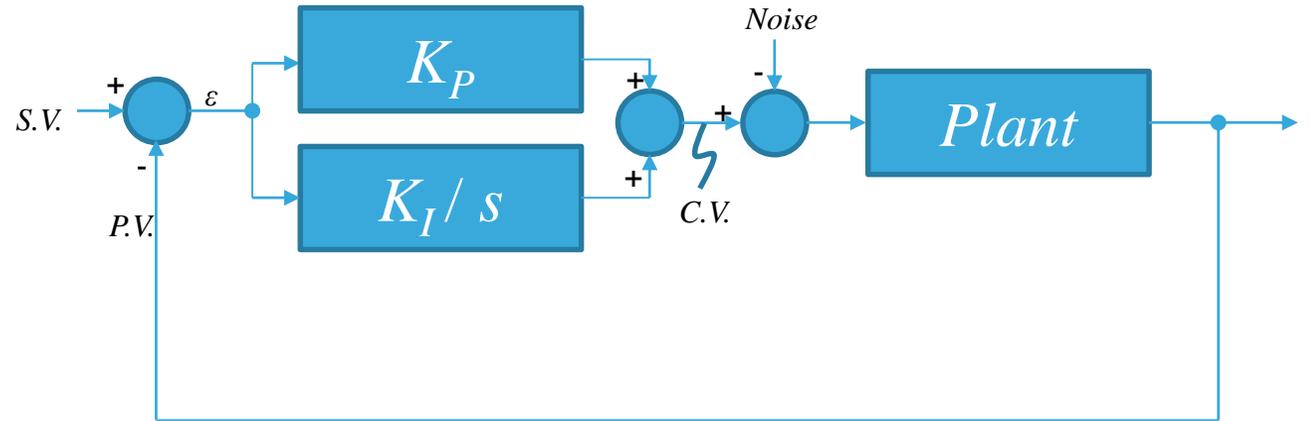
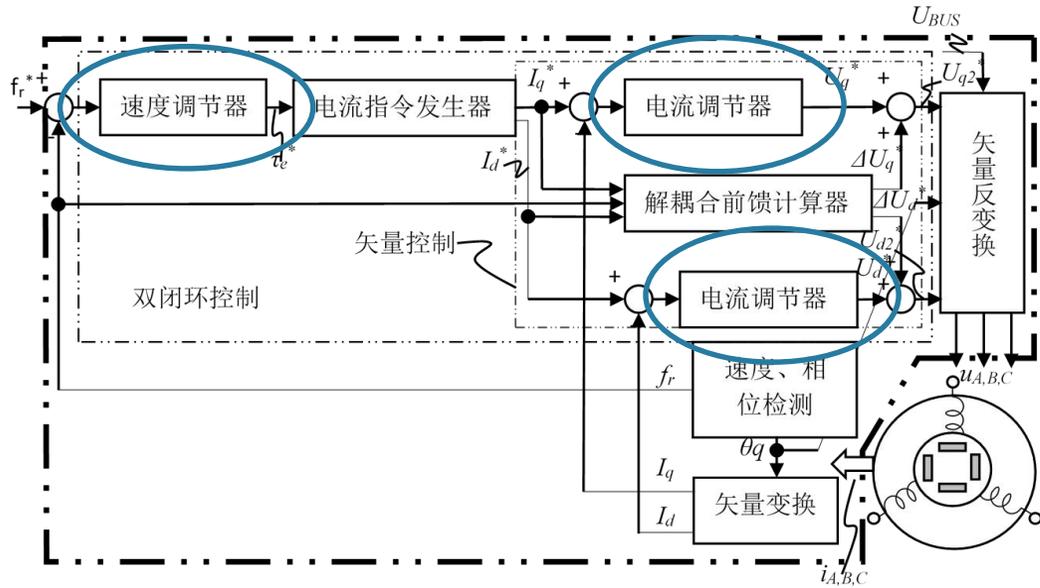




# 矢量控制

- 永磁同步电动机转动原理
- 矢量变换
- 矢量控制
- 空间电压矢量PWM(SVPWM)
- 特别算法介绍
  - MTPA
  - 弱磁控制
  - 电流前馈控制

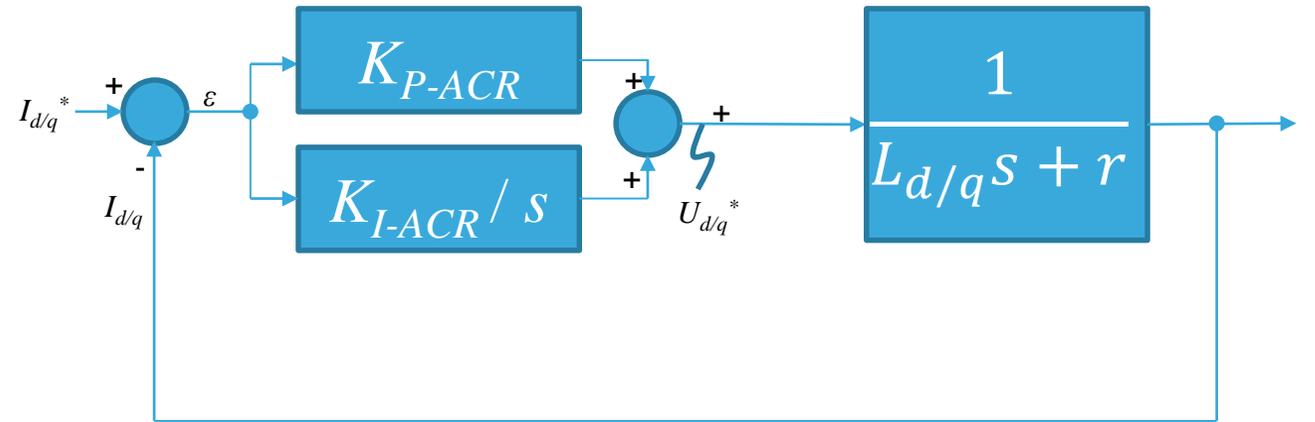
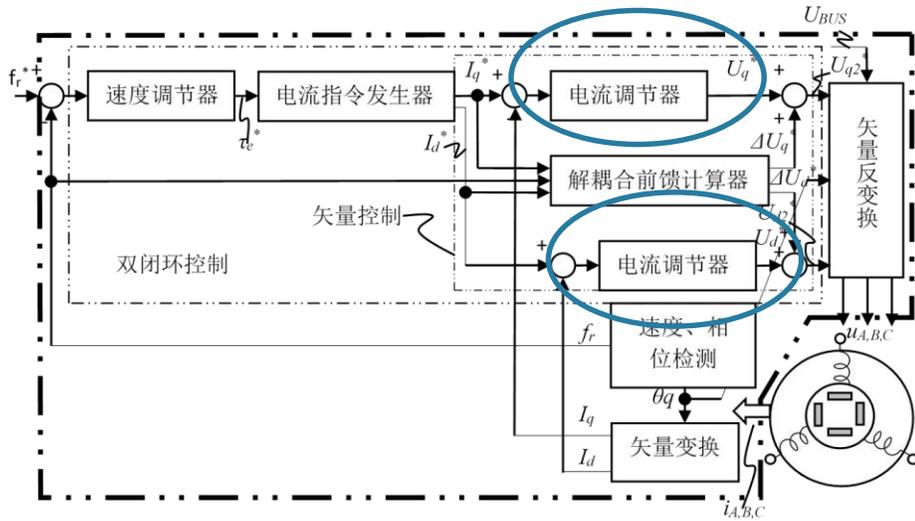
# 矢量控制 — 控制器



控制器	设定值 S.V.	实际值 P.V.	控制值 C.V.	对象传递函数
速度控制器	$f_{r-ref}(f_r^*)$	$f_{r-fb}(f_r)$	$\tau_{ref}(\tau^*)$ or $I_{q-ref}(I_q^*)$	$\frac{1}{Js + F}$
电流控制器 (忽略dq之间的耦合)	$I_{d/q-ref}(I_{d/q}^*)$	$I_{d/q-fb}(I_{d/q})$	$U_{d/q-ref}(U_{d/q}^*)$	$\frac{1}{L_{d/q}s + r}$

# 电流调节器

## --拉普拉斯域设定



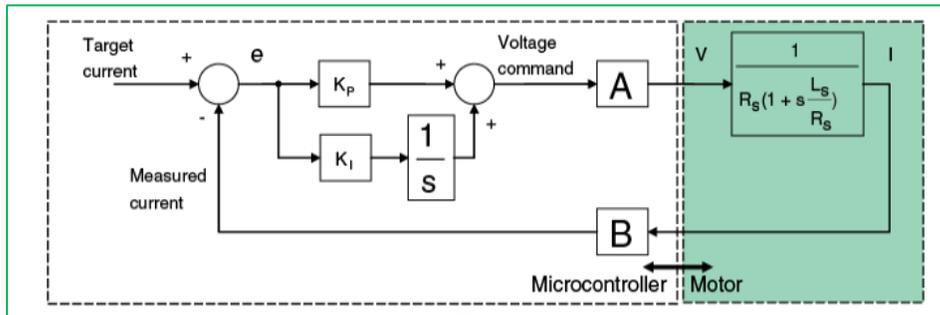
$$K_{P-ACR-d} = L_d \omega_{B-ACR}$$

$$K_{P-ACR-q} = L_q \omega_{B-ACR}$$

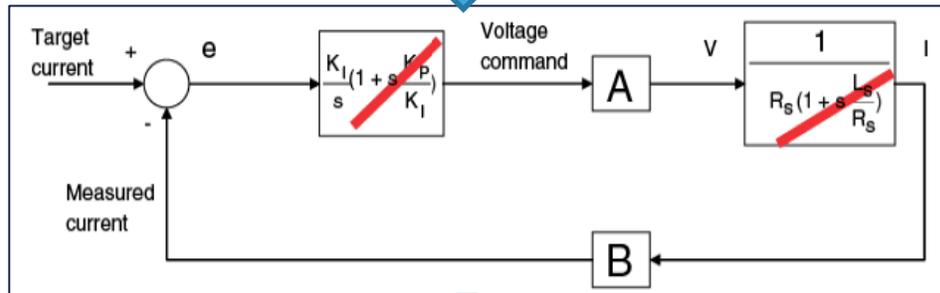
$$K_{I-ACR} = r \omega_{B-ACR}$$

□ 电流调节器的开环增益为

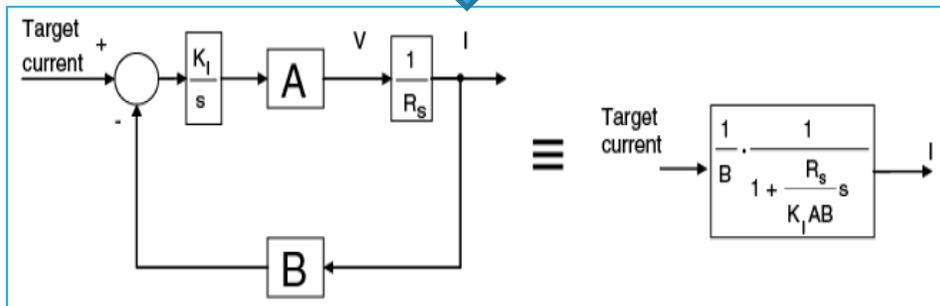
$$G_{ACR-Openloop}(s) = \left( K_{P-ACR-d/q} + \frac{K_{I-ACR}}{s} \right) \frac{1}{L_{d/q}s + r} = \frac{\omega_{B-ACR}}{s}$$



令  $K_P/K_I = L_S/R_S$



变为一阶系统



$$K_P = L_S \frac{\omega_C}{AB}$$

$$K_I = \frac{R_S \cdot \omega_C \cdot T}{AB}$$

$$AB = \frac{V_{BusDC} \cdot R_{shunt} \cdot A_{op}}{3.3}$$

$$A = \frac{V_{BusDC}}{2^{16}}$$

$$B = \frac{R_{shunt} A_{op} 2^{16}}{3.3}$$

## 拉普拉斯域PI系数设定方法

$$K_{P-ASR} = J\omega_{B-ASR}$$

$$K_{I-ASR} = F\omega_{B-ASR}$$

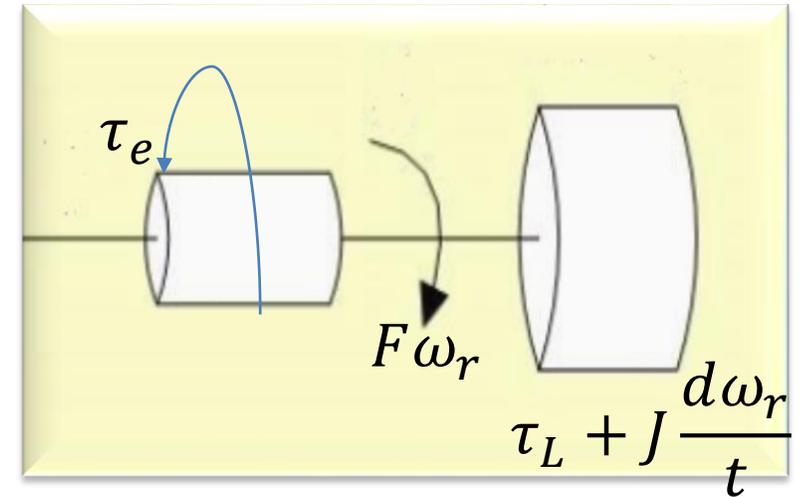
$J$ : 转动惯量 [kg/m<sup>2</sup>];

$F$ : 阻力系数 [Nm/[rad/s]];

$\omega_{B-ASR}$ : 速度调节器通带宽度 [rad/s];

$K_{P-ASR}$ : 速度调节器比例系数;

$K_{I-ASR}$ : 速度调节器积分系数。

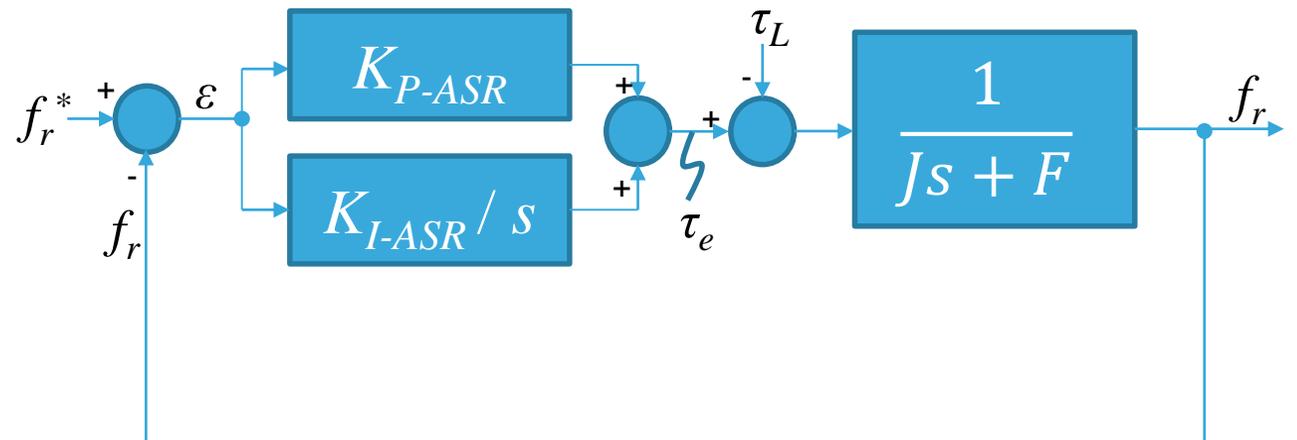


## WB中PI系数的设定

$$K_{P-ASR-WB} = \frac{J_{WB}}{1000000} k(2\pi f_{B-ASR})p$$

$$K_{I-ASR-WB} = \frac{K_{P-ASR-WB}}{1000} \frac{10}{T_{S-ASR-WB}}$$

$$k_\tau = 1.5 \sqrt{\frac{2}{3} \frac{60}{1000 \cdot 2\pi}}, k = \frac{65536 \cdot R_{shunt} \cdot G_{OPAMP}}{3.3k_\tau}, \tau_{mech} = \frac{J_{WB}}{F_{WB}}, f_{B-ASR} = \frac{0.5}{\tau_{mech}} \times 30$$



# PID在ST MC SDK5.x 固件中的实现

## ➤ 具体文件夹如下：

- ✓ xxx\MCSDK\_v5.2.0\MotorControl\MCSDK\MCLib\Any\Src

## ➤ 文件名如下：

- ✓ pid\_regulator.c

## ➤ 函数名：

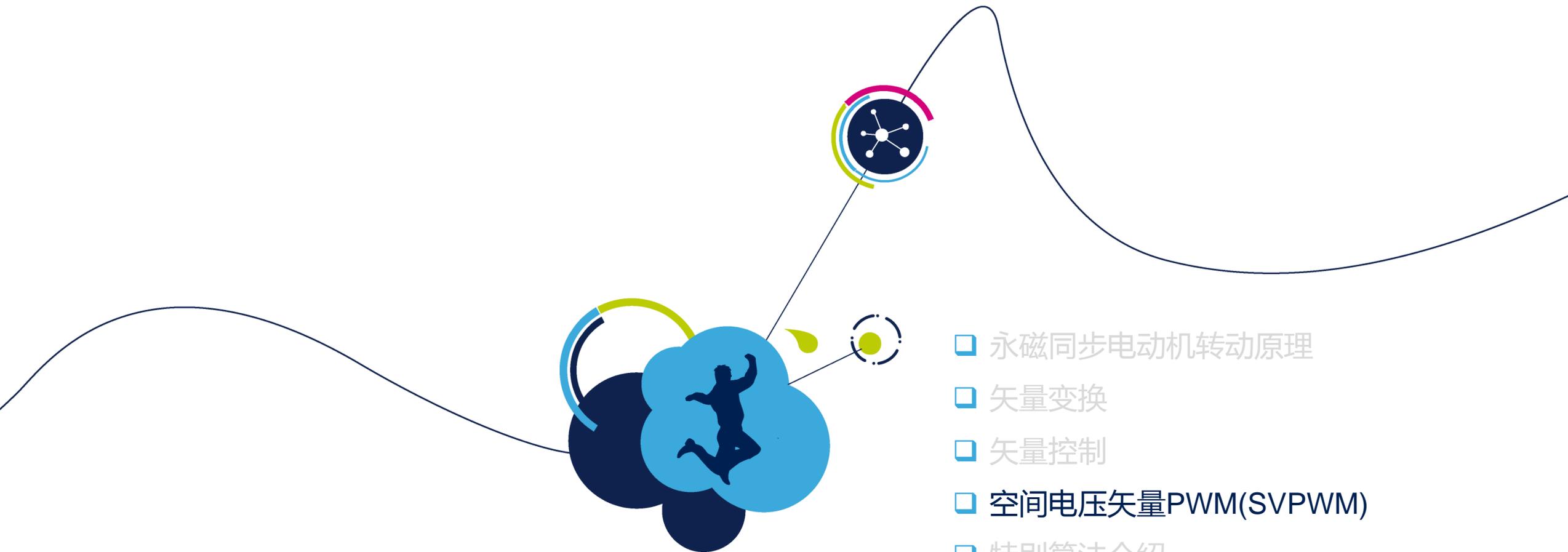
- ✓ PI\_Controller(库中只使用了PI)

```
pid_regulator.c
328 #endif
329 /**
330  * @brief This function compute the output of a PI regulator sum of its
331  *        proportional and integral terms
332  * @param pHandle: handler of the current instance of the PID component
333  * @param wProcessVarError: current process variable error, intended as the reference
334  *        value minus the present process variable value
335  * @retval computed PI output
336  */
337 int16_t PI_Controller( PID_Handle_t * pHandle, int32_t wProcessVarError )
338 {
339     int32_t wProportional_Term, wIntegral_Term, wOutput_32, wIntegral_sum_temp;
340     int32_t wDischarge = 0;
341     int16_t hUpperOutputLimit = pHandle->hUpperOutputLimit;
342     int16_t hLowerOutputLimit = pHandle->hLowerOutputLimit;
343
344     /* Proportional term computation*/
345     wProportional_Term = pHandle->hKpGain * wProcessVarError;
346
347     /* Integral term computation */
348     if ( pHandle->hKiGain == 0 )
349     {
350         pHandle->wIntegralTerm = 0;
351     }
352     else
353     {
```

# PID参数在ST MC SDK5.x固件中的位置

- 具体文件夹如下：
  - ✓ xxx\Inc
- 文件名如下：
  - ✓ drive\_parameters.h
- 参数见右图所示

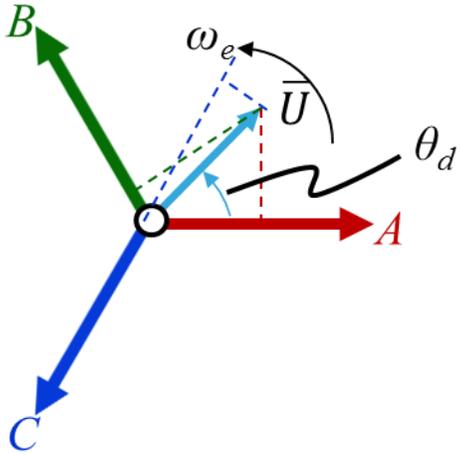
```
drive_parameters.h
217
218 /* Gains values for torque and flux control loops */
219 #define PID_TORQUE_KP_DEFAULT      1051
220 #define PID_TORQUE_KI_DEFAULT      214
221 #define PID_TORQUE_KD_DEFAULT      100
222 #define PID_FLUX_KP_DEFAULT        1051
223 #define PID_FLUX_KI_DEFAULT        214
224 #define PID_FLUX_KD_DEFAULT        100
225                                     电流环PID参数
226 /* Torque/Flux control loop gains dividers*/
227 #define TF_KP_DIV                   16384
228 #define TF_KI_DIV                   16384
229 #define TF_KD_DIV                   8192
230 #define TFDIFFERENTIAL_TERM_ENABLING DISABLE
231
232 /* Speed control loop */
233 #define SPEED_LOOP_FREQUENCY_HZ     1000 /*!<Execution
234
235 #define PID_SPEED_KP_DEFAULT         1000
236 #define PID_SPEED_KI_DEFAULT         600
237 #define PID_SPEED_KD_DEFAULT         0
238 /* Speed PID parameter dividers */ 速度环PID参数
239 #define SP_KP_DIV                    16
240 #define SP_KI_DIV                    256
241 #define SP_KD_DIV                    16
242 /* USER CODE BEGIN PID_SPEED_INTEGRAL_INIT_DIV */
243 #define PID_SPEED_INTEGRAL_INIT_DIV 1 /* */
244 /* USER CODE END PID_SPEED_INTEGRAL_INIT_DIV */
```



# 空间电压矢量PWM(SVPWM)

- 永磁同步电动机转动原理
- 矢量变换
- 矢量控制
- 空间电压矢量PWM(SVPWM)
- 特别算法介绍
  - MTPA
  - 弱磁控制
  - 电流前馈控制

# 空间电压矢量PWM(SVPWM)(1/3)

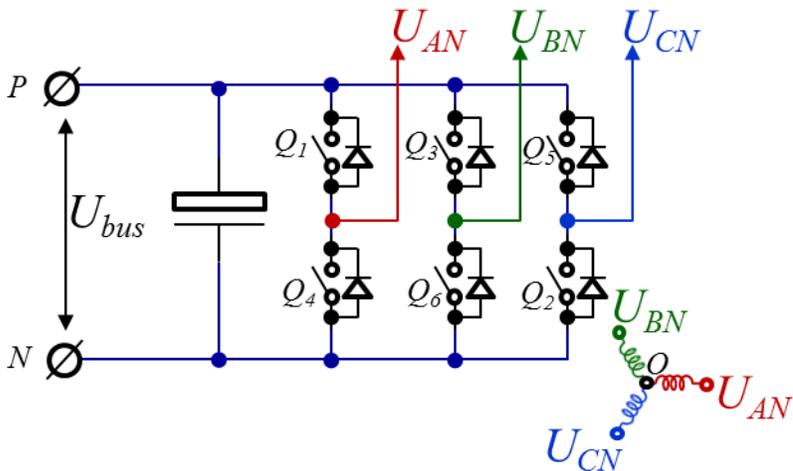


$$U_{AO} = U_m \cos(\theta_d) = U_m \cos(\omega_e t)$$

$$U_{BO} = U_m \cos(\theta_d - 120^\circ) = U_m \cos(\omega_e t - 120^\circ)$$

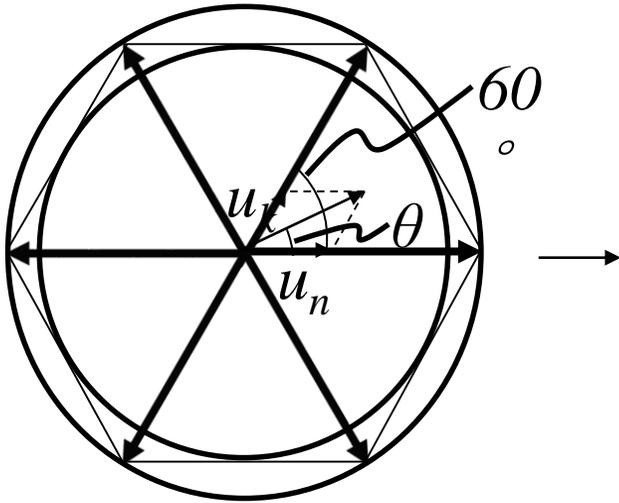
$$U_{CO} = U_m \cos(\theta_d + 120^\circ) = U_m \cos(\omega_e t + 120^\circ)$$

$$|\vec{U}| = U_m$$

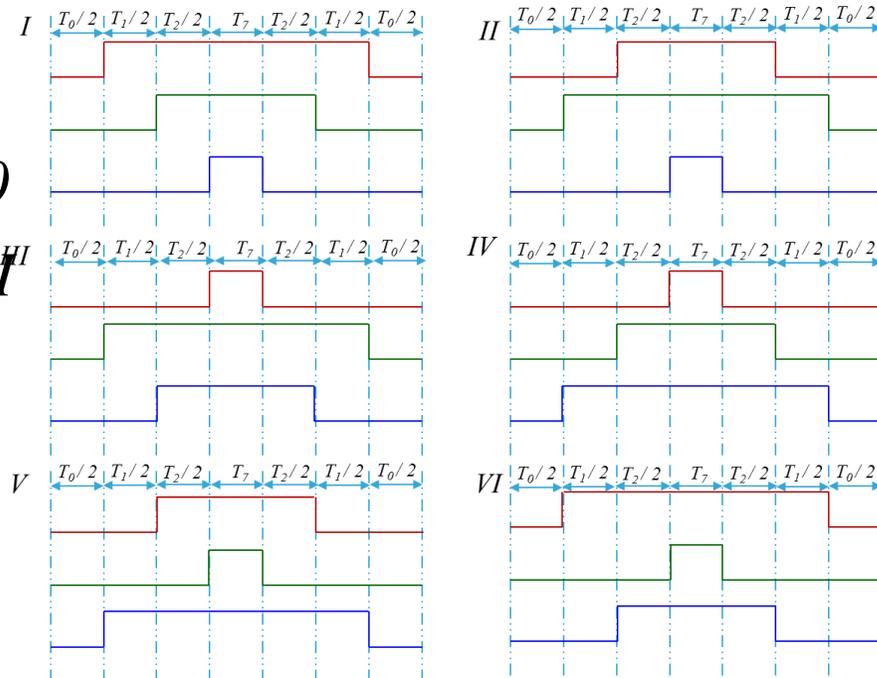
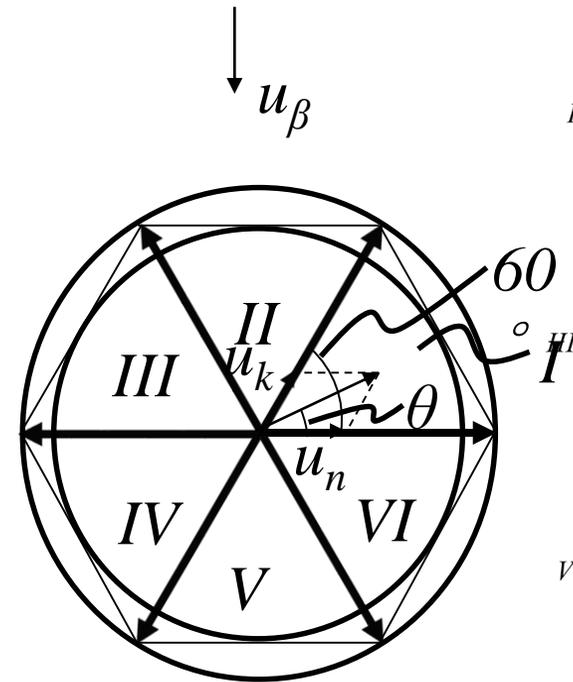


vector	$Q_1(Q_4)$	$Q_3(Q_6)$	$Q_5(Q_2)$	$U_{AN} [u]$	$U_{BN}[u]$	$U_{CN}[u]$	$U_{AO} [u]$	$U_{BO}[u]$	$U_{CO}[u]$
$\vec{U}_0$	OFF(ON)	OFF(ON)	OFF(ON)	0	0	0	0	0	0
$\vec{U}_1$	ON(OFF)	OFF(ON)	OFF(ON)	$U_{bus}$	0	0	$2U_{bus}/3$	$-U_{bus}/3$	$-U_{bus}/3$
$\vec{U}_2$	ON(OFF)	ON(OFF)	OFF(ON)	$U_{bus}$	$U_{bus}$	0	$U_{bus}/3$	$U_{bus}/3$	$-2U_{bus}/3$
$\vec{U}_3$	OFF(ON)	ON(OFF)	OFF(ON)	0	$U_{bus}$	0	$-U_{bus}/3$	$2U_{bus}/3$	$-U_{bus}/3$
$\vec{U}_4$	OFF(ON)	ON(OFF)	ON(OFF)	0	$U_{bus}$	$U_{bus}$	$-2U_{bus}/3$	$U_{bus}/3$	$U_{bus}/3$
$\vec{U}_5$	OFF(ON)	OFF(ON)	ON(OFF)	0	0	$U_{bus}$	$-U_{bus}/3$	$-U_{bus}/3$	$2U_{bus}/3$
$\vec{U}_6$	ON(OFF)	OFF(ON)	ON(OFF)	$U_{bus}$	0	$U_{bus}$	$U_{bus}/3$	$-2U_{bus}/3$	$-U_{bus}/3$
$\vec{U}_7$	ON(OFF)	ON(OFF)	ON(OFF)	$U_{bus}$	$U_{bus}$	$U_{bus}$	0	0	0

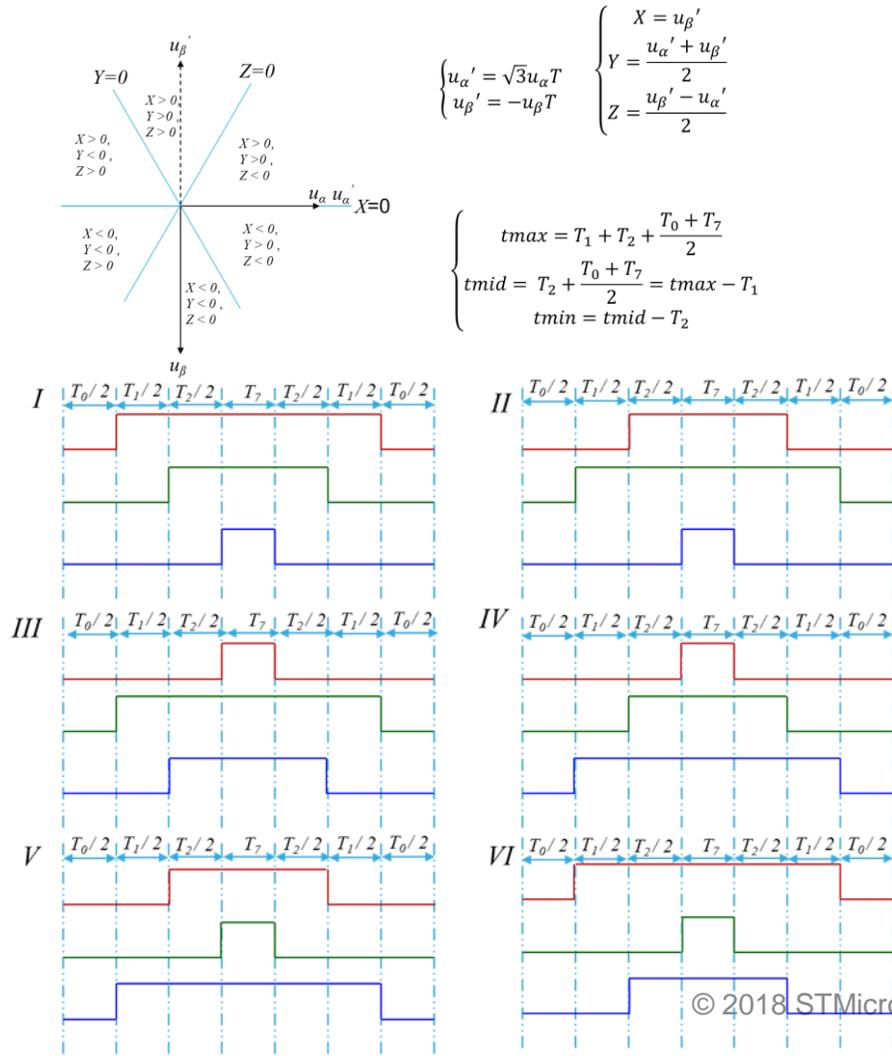
# 空间电压矢量PWM(SVPWM)(2/3)



Sector	I	II	III	IV	V	VI
<i>n</i>	1	3	3	5	5	1
<i>k</i>	2	2	4	4	6	6



# 空间电压矢量PWM(SVPWM)(3/3)



sector	T1	T2	T0+T7
I	-Z	X	T + Z - X
II	Z	Y	T - Y - Z
III	X	-Y	T - X + Y
IV	-X	Z	T + X - Z
V	-Y	-Z	T + Y + Z
VI	Y	-X	T + X - Y

sector	tA	tB	tC
I	T/2 + (X - Z)/2	tA + Z	tB - X
II	T/2 + (Y - Z)/2	tA + Z	tA - Y
III	T/2 + (Y - X)/2	tC + X	tA - Y
IV	T/2 + (X - Z)/2	tA + Z	tB - X
V	T/2 + (Y - Z)/2	tA + Z	tA - Y
VI	T/2 + (Y - X)/2	tC + X	tA - Y

# SVPWM在ST MC SDK5.x固件中的实现

## ➤ 具体文件夹如下：

- ✓ xxx\MCSDK\_v5.2.0\MotorControl\MCSDK\MCLib\Any\Src

## ➤ 文件名如下：

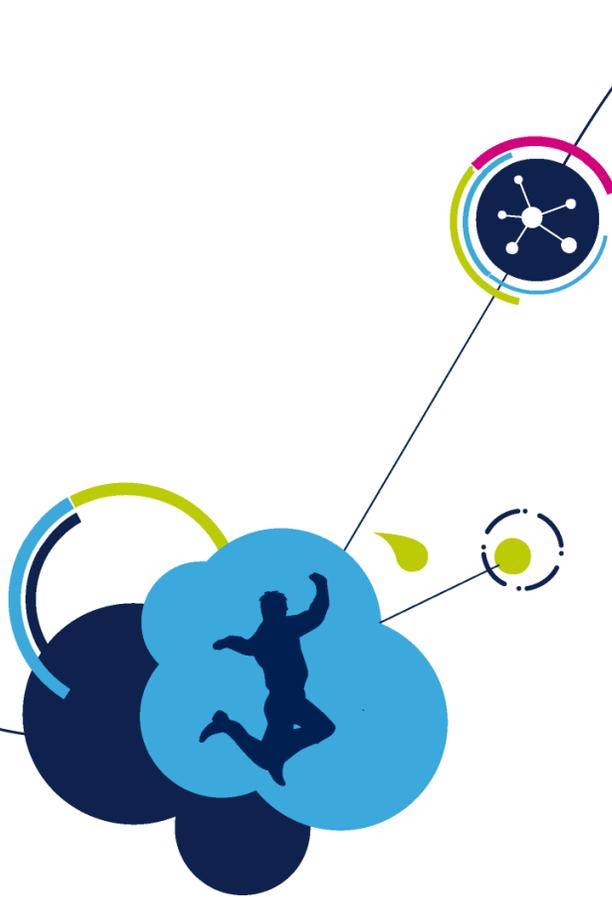
- ✓ pwm\_curr\_fdbk.c

## ➤ 函数名：

- ✓ PWMC\_SetPhaseVoltage

```
pwm_curr_fdbk.c
149 *
150 * @retval Returns #MC_NO_ERROR if no error occurred or #MC_FOC_DURATION if the duty cycles were
151 *         set too late for being taken into account in the next PWM cycle.
152 */
153 uint16_t PWMC_SetPhaseVoltage( PWMC_Handle_t * pHandle, Volt_Components Valfa_beta )
154 {
155     int32_t wX, wY, wZ, wUAlpha, wUBeta, wTimePhA, wTimePhB, wTimePhC;
156     PWMC_SetSampPointSectX_Cb_t pSetADCSamplingPoint;
157
158     wUAlpha = Valfa_beta.qV_Component1 * ( int32_t )pHandle->hT_Sqrt3;
159     wUBeta = -( Valfa_beta.qV_Component2 * ( int32_t )( pHandle->hPWMperiod ) ) * 2;
160
161     wX = wUBeta;
162     wY = ( wUBeta + wUAlpha ) / 2;
163     wZ = ( wUBeta - wUAlpha ) / 2;
164
165     /* Sector calculation from wX, wY, wZ */
166     if ( wY < 0 )
167     {
168         if ( wZ < 0 )
169         {
170             pHandle->hSector = SECTOR_5;
171             wTimePhA = ( int32_t )( pHandle->hPWMperiod ) / 4 + ( ( wY - wZ ) / ( int32_t )262144 );
172             wTimePhB = wTimePhA + wZ / 131072;
173             wTimePhC = wTimePhA - wY / 131072;
174             pSetADCSamplingPoint = pHandle->pFctSetADCSampPointSect5;
175         }
176     }
}
```

# 特别算法介绍



- 永磁同步电动机转动原理
- 矢量变换
- 矢量控制
- 空间电压矢量PWM(SVPWM)
- 特别算法介绍
  - MTPA
  - 弱磁控制
  - 电流前馈控制

# 每安培最大转矩—MTPA ( 1/8 )

- 电磁转矩方程如下：

$$\tau_e = \frac{3}{2} p [k_E I_q + (L_d - L_q) I_d I_q]$$

- 如果 $L_d=L_q$ ,那么

$$\tau_e = \frac{3}{2} p k_E I_q$$

# 每安培最大转矩—MTPA ( 2/8 )

- 电动机的电压方程：

$$\begin{cases} U_d = rI_d - L_q I_q \omega_e \\ U_q = rI_q + k_E \omega_e + L_d I_d \omega_e \end{cases}$$

- 那么，电动机消耗的有功功率为

$$P = \frac{3}{2} (U_d I_d + U_q I_q)$$

- 将电动机的方程代入功率方程，得

$$P = \frac{3}{2} r (I_d^2 + I_q^2) + [k_E I_q + (L_d - L_q) I_d I_q] \omega_e$$

# 每安培最大转矩—MTPA ( 3/8 )

- 电动机的有功功率一部分消耗在绕组电阻上发热了，另一部分则用于输出机械功率

$$P = \underbrace{\frac{3}{2} r (I_d^2 + I_q^2)}_{\text{绕组电阻发热}} + \underbrace{\frac{3}{2} [k_E I_q + (L_d - L_q) I_d I_q]}_{\text{用于输出机械功率}} \omega_e$$

绕组电阻发热

用于输出机械功率

- 电动机的机械功率为

$$P_{mech} = \tau_e \omega_{mech}$$

# 每安培最大转矩—MTPA ( 4/8 )

- 机械转速与电频率之间的关系为：

$$\omega_e = p\omega_{mech}$$

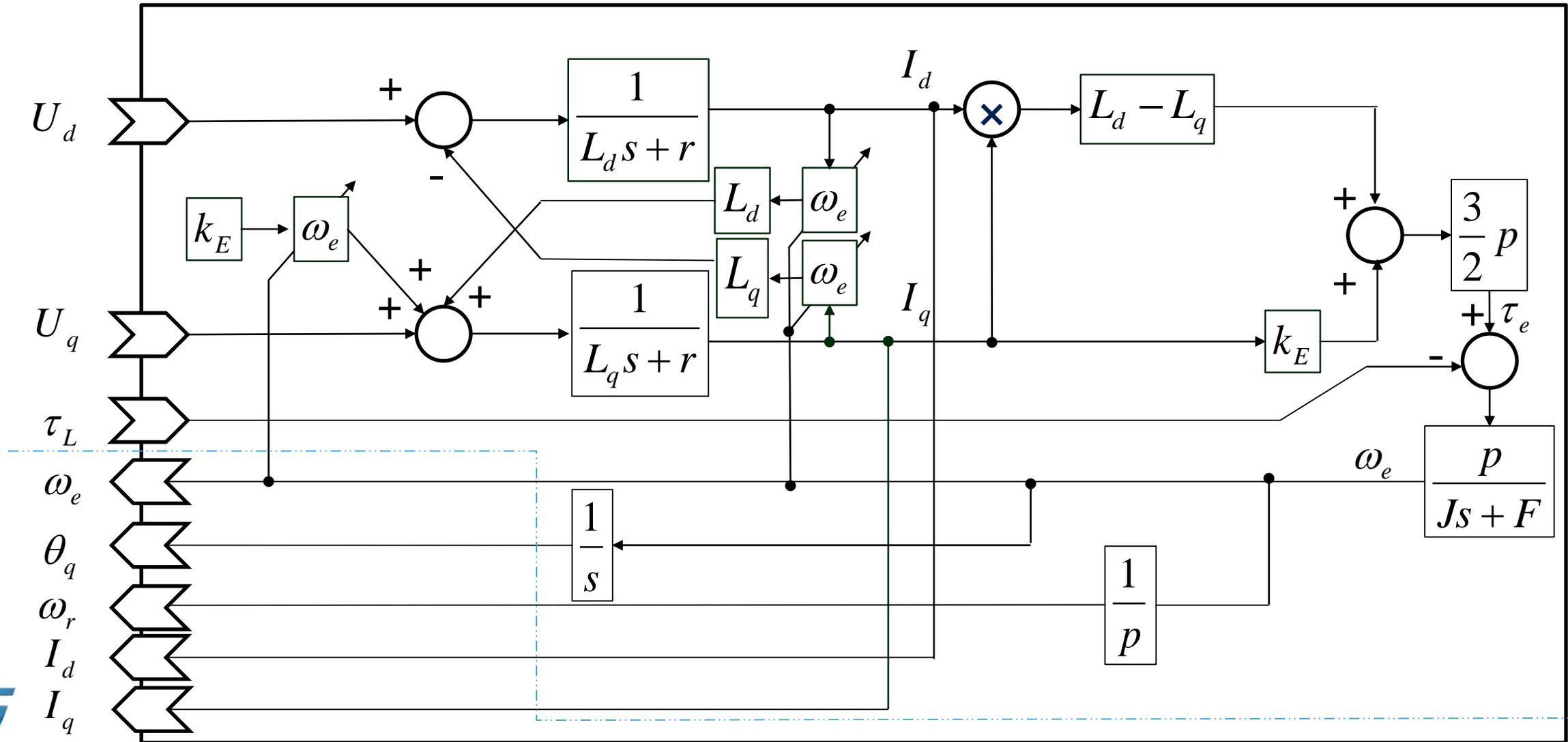
- 所以

$$\tau_e \omega_{mech} = \frac{3}{2} \left[ k_E I_q + (L_d - L_q) I_d I_q \right] p \omega_{mech}$$

- 得到了电磁转矩的公式为，

$$\tau_e = \frac{3}{2} p \left[ k_E I_q + (L_d - L_q) I_d I_q \right]$$

# 一种永磁同步电动机的同步状态工作模型



# 每安培最大转矩—MTPA ( 5/8 )

- 如果绕组中的电流峰值是 $I_s$

$$I_s^2 = I_d^2 + I_q^2 \quad (1)$$

- 当输出电磁转矩一定时，

$$I_q = \frac{\tau_e}{\frac{3}{2} p [k_E + (L_d - L_q) I_d]} \quad (2)$$

# 每安培最大转矩—MTPA ( 6/8 )

- 将(2)代入(1)

$$I_s^2 = I_d^2 + \left( \frac{\tau_e}{\frac{3}{2} p [k_E + (L_d - L_q) I_d]} \right)^2 \quad (3)$$

# 每安培最大转矩—MTPA ( 7/8 )

- (3)式可以看作是一个关于 $I_d$ 的函数，经过分析，它有最小值，也就是说，当输出电磁转矩一定时，有一个最小的峰值电流 $I_s$ 。当取得电流峰值 $I_s$ 最小时，

$$\begin{aligned} \frac{\partial I_s^2}{\partial I_d} &= 2I_d + 2I_q \frac{\tau_e}{\frac{3}{2}p} \frac{-(L_d - L_q)}{[k_E + (L_d - L_q)I_d]^2} \\ &= 2I_d + 2I_q \frac{\tau_e}{\frac{3}{2}p[k_E + (L_d - L_q)I_d]} \frac{-(L_d - L_q)}{[k_E + (L_d - L_q)I_d]} \\ &= 2I_d + 2I_q^2 \frac{-(L_d - L_q)}{[k_E + (L_d - L_q)I_d]} = 0 \end{aligned} \quad (4)$$

整理(4)，

$$(L_d - L_q)I_d^2 + k_E I_d - (L_d - L_q)I_q^2 = 0 \quad (4')$$

# 每安培最大转矩—MTPA ( 8/8 )

- 在(4)中，以 $I_d$ 为未知数，求解方程，得

$$I_d = -\frac{k_E}{2(L_d - L_q)} \pm \sqrt{\left(\frac{k_E}{2(L_d - L_q)}\right)^2 + I_q^2} \quad (5)$$

- 如果 $L_d < L_q$ ，那么当 $I_d$ 满足(6)的时候，当电磁转矩一定时，电流峰值 $I_s$ 最小

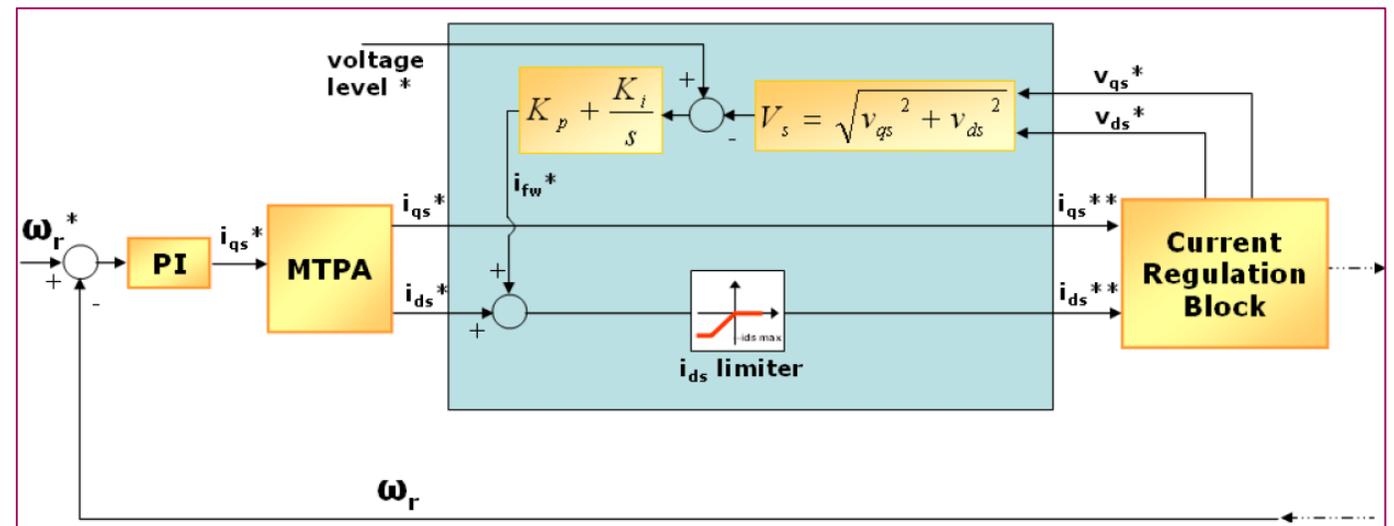
$$I_d = -\frac{k_E}{2(L_d - L_q)} - \sqrt{\left(\frac{k_E}{2(L_d - L_q)}\right)^2 + I_q^2} \quad (6)$$

- 很多应用需要马达工作在高于额定速度的范围内，这里就需要弱磁控制来实现
- 电流的电压约束条件

$$(L_q I_q)^2 + (k_E + L_d I_d)^2 \leq \frac{U_{1-limit}^2}{\omega_e^2}$$

- 电流幅值约束条件

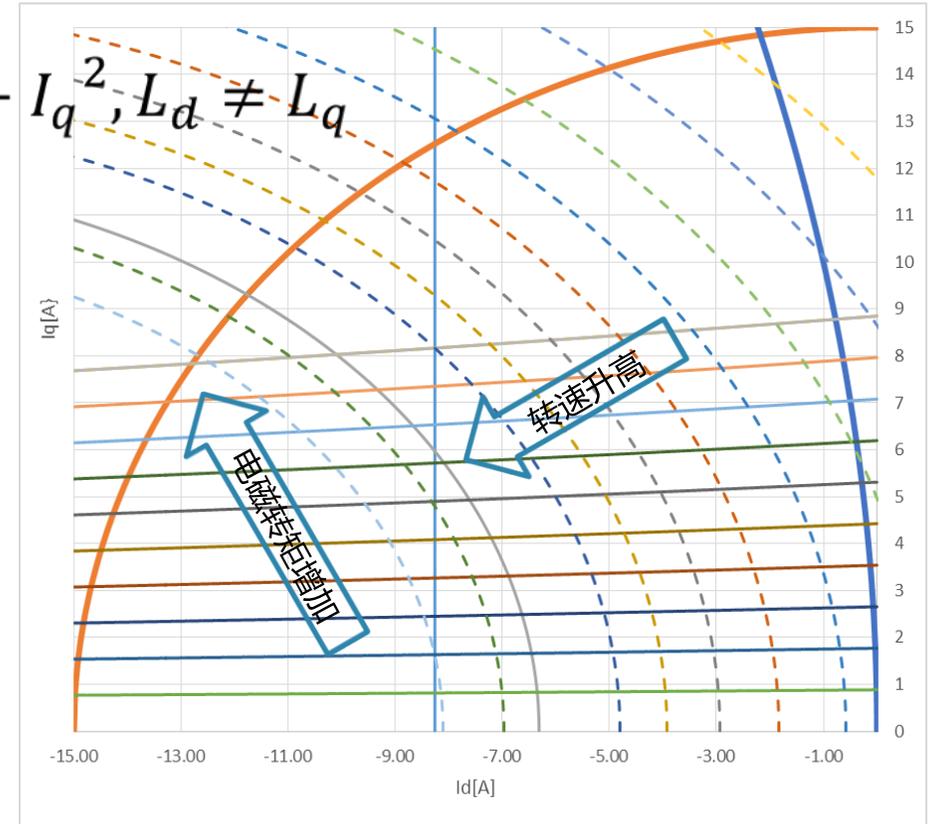
$$I_d^2 + I_q^2 \leq I_{s-limit}^2$$

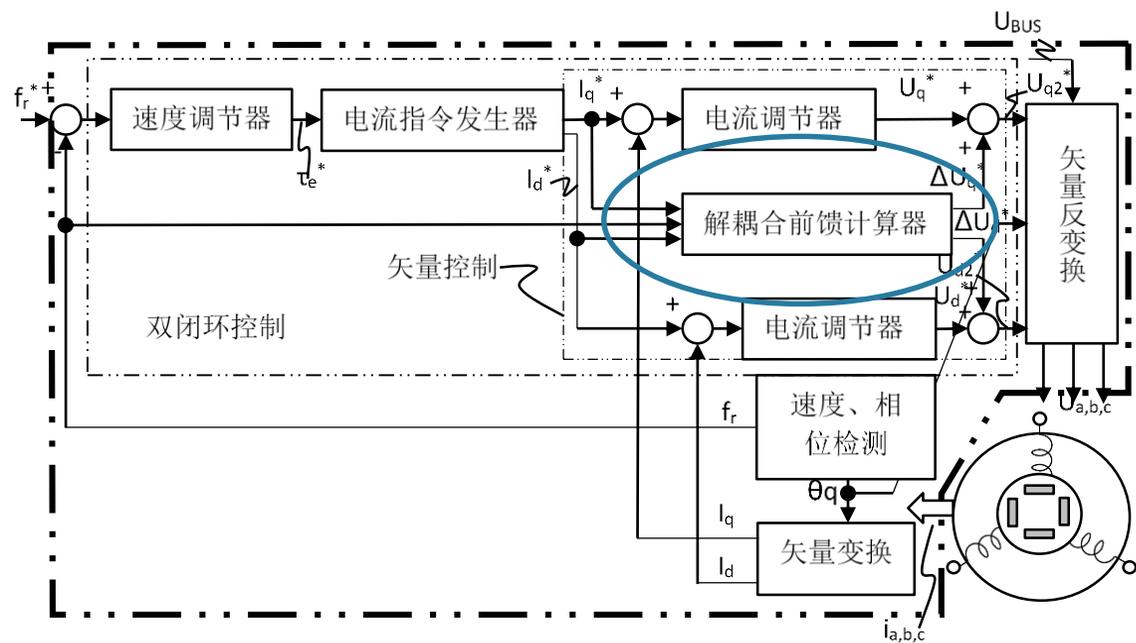


$$\text{MTPA: } I_d = \begin{cases} -\frac{k_E}{2(L_d - L_q)} + \text{sign}(L_d - L_q) \sqrt{\left(\frac{k_E}{2(L_d - L_q)}\right)^2 + I_q^2}, & L_d \neq L_q \\ 0, & L_d = L_q \end{cases}$$

$$\text{sign}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}$$

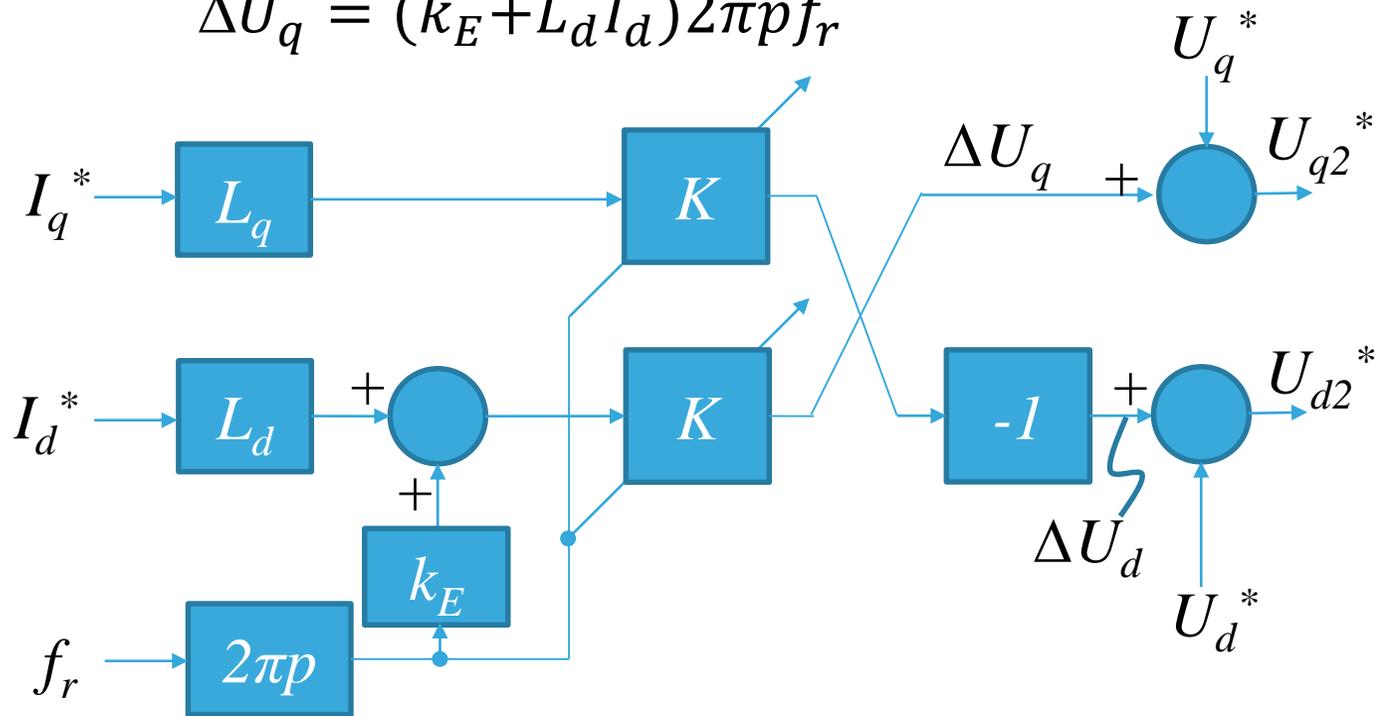
$$\text{弱磁: } I_d = \begin{cases} -\frac{k_E}{L_d} + \frac{\sqrt{\left(\frac{U_{1\text{-limit}}}{\omega_e}\right)^2 - (L_q I_q)^2}}{L_d}, & \frac{U_{1\text{-limit}}}{\omega_e} \geq L_q I_d \\ n/a, & \frac{U_{1\text{-limit}}}{\omega_e} < L_q I_d \end{cases}$$





$$\Delta U_d = -L_q I_q 2\pi p f_r$$

$$\Delta U_q = (k_E + L_d I_d) 2\pi p f_r$$



# ST MC SDK5.x对特别算法的实现

➤ 库里面的特别算法以库函数方式存在，文件夹如下：

✓ xxx\MCSDK\_v5.2.0\MotorControlLib

➤ 头文件名如下：

✓ max\_torque\_per\_ampere.h

✓ flux\_weakening\_ctrl.h

✓ feed\_forward\_ctrl.h

 bus_voltage_sensor.h	
 circle_limitation.h	
 digital_output.h	
 enc_align_ctrl.h	
 encoder_speed_pos_fdbk.h	
 feed_forward_ctrl.h	电流前馈头文件
 flux_weakening_ctrl.h	弱磁头文件
 hall_speed_pos_fdbk.h	
 ics_dd_pwmncurfdbk.h	
 inrush_current_limiter.h	
 max_torque_per_ampere.h	MTPA头文件
 mc_extended_api.h	
 mc_interface.h	
 mc_irq_handler.h	
 mc_math.h	
 mc_tuning.h	
 mc_type.h	

# Releasing your creativity



- Thank you -

## 重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对ST 产品和/ 或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于ST 产品的最新信息。ST 产品的销售依照订单确认时的相关ST 销售条款。

买方自行负责对ST 产品的选择和使用， ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的ST 产品如有不同于此处提供的信息的规定，将导致ST 针对该产品授予的任何保证失效。

ST 和ST 徽标是ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。

## 版权声明

本文档为意法半导体公司及其子公司（“ST”）版权所有，未经ST允许不得复制、修改、转发或应用于商业目的。