## 电机参数

### 状态回读

* 回读ID
* 回读角度
* 回读PID参数
* 最大最小占空比
* 回读目标角度容错偏差
* 回读零点位置偏差
* 回读初始上电所处工作模式
* 回读自我保护“反应时间”
* 回读舵机内部温度
* 回读舵机当前电压
* 回读舵机当前电流

### 控制变量

* 角速度 同启同停
* 力矩控制

### 工作模式

* 有限角度控制模式
* 多圈大角度控制模式
* 连续转动模式
* 阻尼模式
* 待机模式

## 运动控制函数

### 有限角度控制模式下的转动位置及速度函数

void set\_angle(int id\_num, float angle, int step )

参数及解释： > id\_num - 舵机编号范围0~255（除去121,126,127 共253个），默认为0；121（0x79）号为舵机的共同编号（以下统称广播编号），即所有舵机都会响应到控制 指令并执行。 > angle - 设置舵机转动角度，可选角度范围为0~270°。

step - 设置舵机运动步数，该参数可调节转动速度。舵机要达到指定角度分步的次数，以10ms为周期每周期最多运动4度，步数设置为“1”时则以最快速度转动，步数越大速度越低。

### 设置转动力矩函数

void set\_torque (int id\_num, double torque);

参数及解释： > id\_num - 舵机编号，即要设置第几号舵机的连续转动模式下的转动扭矩，这里可以用广播编号

torque - 参数范围（-3~3），参数为正，舵机则正转，反之反转。 （参数的绝对值表示力矩的相对大小）

### 回读状态（get\_state）

int get\_state(int id\_num, int para\_num，int o\_m);

返回值：

float servo\_rpara[ID,AngleNow,AngleExp,Time,StatesNow,];

参数解释： >

# 运动学反解

对于

vfgwfasgwfs

对于

对于笔尖坐标系:

通过测量得：

总公式，从 映射到驱动舵机 的公式为： $$ $$

则根据以上公式可以编写如下代码：

/\*\*  
 \* @fn float\* cla\_angle(float x\_pen, float y\_pen)  
 \* @brief 通过笔尖坐标系的坐标计算舵机角度  
 \*   
 \* 该函数根据逆运动学公式组计算从笔尖坐标 \((x\_{pen},y\_{pen})\) 到驱动舵机角度 \((\theta\_1,\theta\_4)\) 的映射值。  
 \* 公式中涉及的中间变量包括：  
 \* - 对于舵机1（\(\theta\_1\)）：   
 \* d₁ = -2·l₁·y\_c   
 \* e₁ = -2·l₁·x\_c   
 \* f₁ = x\_c² + y\_c² + l₁² - l₂²   
 \* \(\theta\_1 = 2 \arctan\Bigl(\frac{d₁ \pm \sqrt{d₁^2+e₁^2-f₁^2}}{e₁-f₁}\Bigr)\)  
 \* - 对于舵机4（\(\theta\_4\)）：   
 \* d₂ = -2·l₃·y\_c   
 \* e₂ = 2·l₄·l₅ - 2·x\_c   
 \* f₂ = x\_c² + y\_c² + l₄² - l₃² + l₅² - 2·l₅·x\_c   
 \* \(\theta\_4 = 2 arctan\Bigl(\frac{d₂ \pm \sqrt{d₂^2+e₂^2-f₂^2}}{e₂-f₂}\Bigr)\)  
 \*  
 \* 固定参数根据实际测量得到：  
 \* l₁ = 90, l₂ = 130, l₃ = 130, l₄ = 90, l₅ = 105, l₆ = 35, l\_M = 150.  
 \*  
 \* @param x\_pen 笔尖坐标系的 x 坐标  
 \* @param y\_pen 笔尖坐标系的 y 坐标  
 \* @return float\* 返回一个长度为2的数组，其中 angle[0] 为舵机1角度 θ₁，angle[1] 为舵机4角度 θ₄  
 \*/  
float\* cla\_angle(float x\_pen, float y\_pen) {  
 // 静态数组保证在函数返回后数据依然有效  
 static float angle[2];  
  
 // 固定参数（单位：毫米或其它统一单位）  
 const float l1 = 90.0f;  
 const float l2 = 130.0f;  
 const float l3 = 130.0f;  
 const float l4 = 90.0f;  
 const float l5 = 105.0f;  
 const float l6 = 35.0f;  
 const float lM = 95.0f;  
  
 // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 // 【注意】：  
 // 公式中涉及中间变量 x\_c, y\_c 为工作坐标系坐标，通常由笔尖坐标转换得到。  
 // 此处为简化计算，假设工作坐标系与笔尖坐标系重合，即：  
 float x\_c = x\_pen;  
 float y\_c = y\_pen;  
 // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
 // 计算 φ₀ = arctan( y\_pen / (x\_pen - l5/2) )  
 float phi0 = atan( y\_pen / ( x\_pen - l5 / 2.0f ) );  
  
 /\*\*\*\*\*\*\*\*\*\*\*\*\* 计算舵机1角度 φ₁ \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
 float d1 = 2.0f \* l2 \* x\_pen - 2.0f \* l2 \* lM \* cos(phi0);  
 float e1 = 2.0f \* l2 \* y\_pen - 2.0f \* l2 \* lM \* sin(phi0);  
 float f1 = x\_pen \* x\_pen + y\_pen \* y\_pen + l2 \* l2 + lM \* lM - l1 \* l1  
 - 2.0f \* lM \* x\_pen \* cos(phi0) + 2.0f \* lM \* y\_pen \* sin(phi0);  
 float discriminant1 = d1 \* d1 + e1 \* e1 - f1 \* f1;  
 discriminant1 = discriminant1 < 0 ? 0 : discriminant1;  
 float phi1 = 2.0f \* atan((d1 + sqrt(discriminant1)) / (e1 - f1));  
 // 转换 φ₁ 从弧度到角度  
 phi1 = phi1 \* 180.0f / 3.14159265358979323846f;  
  
 /\*\*\*\*\*\*\*\*\*\*\*\*\* 计算舵机4角度 φ₄ \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
 float d2 = 2.0f \* l4 \* y\_pen - 2.0f \* l4 \* lM \* sin(phi0);  
 float e2 = 2.0f \* l4 \* x\_pen - 2.0f \* l4 \* lM \* cos(phi0) - 2.0f \* l4 \* l5;  
 float f2 = x\_pen \* x\_pen + y\_pen \* y\_pen + l4 \* l4 + lM \* lM + l5 \* l5 - l3 \* l3  
 - 2.0f \* lM \* x\_pen \* cos(phi0) - 2.0f \* l5 \* x\_pen + 2.0f \* lM \* l5 \* cos(phi0)  
 - 2.0f \* lM \* x\_pen \* sin(phi0);  
 float discriminant2 = d2 \* d2 + e2 \* e2 - f2 \* f2;  
 discriminant2 = discriminant2 < 0 ? 0 : discriminant2;  
 float phi4 = 2.0f \* atan((d2 + sqrt(discriminant2)) / (e2 - f2));  
 // 转换 φ₄ 从弧度到角度  
 phi4 = phi4 \* 180.0f / 3.14159265358979323846f;  
  
 /\*\*\*\*\*\*\*\*\*\*\*\*\* 返回计算结果 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
 angle[0] = phi1;  
 angle[1] = phi4;  
 return angle;  
}

以上代码实现了函数 cla\_angle，通过输入笔尖坐标 计算出对应的舵机角度。注释中详细说明了各个中间变量的物理意义和计算公式，便于后续调试和维护。如果需要完善笔尖坐标到工作坐标的转换或更精确地选择双值解，请在此基础上进行扩展。