圆点博士小四轴 叉积法融合陀螺和加速度 核心程序的较详细注释（可能有误、多多指教） by 司马青衫

```
// 叉积法融合陀螺和加速度。
void mix_gyrAcc_crossMethod(quaternion_yuandian * attitude,const float gyr[3],const float acc[3],float interval)
{
    const static float FACTOR = 0.001;//两个重力矢量叉积后所乘的系数 p，用于和陀螺仪积分角度相叠加来修正陀螺仪（这里只用了比例 p，没用积分 i，）
    //FACTOR 为 1，则完全信任加速度计，为 0，则完全信任陀螺仪
    float w_q = attitude->w;//w=cos(alpha/2)
    float x_q = attitude->x;//x=ax*sin(alpha/2)
    float y_q = attitude->y;//y=ay*sin(alpha/2)
    float z_q = attitude->z;//z=az*sin(alpha/2)
    float x_q_2 = x_q * 2;
    float y_q_2 = y_q * 2;
    float z_q_2 = z_q * 2;
    //
    // 加速度计的读数，单位化。
    float a_rsqrt = math_rsqrt(acc[0]*acc[0]+acc[1]*acc[1]+acc[2]*acc[2]);
    float x_aa = acc[0] * a_rsqrt;
    float y_aa = acc[1] * a_rsqrt;
    float z_aa = acc[2] * a_rsqrt;
    //
    // 载体坐标下的重力加速度常量，单位化。//用旋转矩阵将世界坐标系的单位化重力矢量(0,0,1）不是（0,0，-1）,mpu6050 只感应非重力加速度)转换到机载坐标系中。
    //机载坐标下的重力矢量    旋转矩阵（坐标系转换矩阵的逆矩阵也就是转置矩阵，因为欧拉角解得的旋转矩阵必是正交阵）     世界坐标下的重力矢量
    //          x                       cos(T)cos(K)                    cos(T)sin(C)                -sin(C)                     0
    //        [ y ]     =       [   sin(F)sin(T)cos(K)-cos(F)sin(K)    sin(F)sin(T)sin(K)+cos(F)cos(K)    sin(F)cos(T)   ]     * [   0   ]
    //          z                   cos(F)sin(T)cos(K)+sin(F)sin(K)    cos(F)sin(T)sin(K)-sin(F)cos(K)    cos(F)cos(T)                1
    //K 是 yaw,T 是 pitch,F 是 roll,旋转顺序为 ZYX
    //                                      w^2+x^2-y^2-z^2     2*(x*y+w*z)       2*(x*z-w*y)
    //上式中的旋转矩阵用四元数表示即为 ：[   2*(x*y-w*z)       w^2-x^2+y^2-z^2     2*(y*z+w*x)     ]
    //                                      2*(x*z+w*y)         2*（y*z-w*x）    w^2-x^2-y^2+z^2
    //
    float x_ac = x_q*z_q_2 - w_q*y_q_2;//   2*(x*z-w*y)                          =ax*az(1-cos(alpha))-ay*sin(alpha)
    float y_ac = y_q*z_q_2 + w_q*x_q_2;//   2*(y*z+w*x)                          =az*ay(1-cos(alpha))+ax*sin(alpha)
    float z_ac = 1 - x_q*x_q_2 - y_q*y_q_2;// w^2+x^2-y^2-z^2 =1-2*x^2-2*y^2   = cos(alpha)+(1-cos(alpha)*z^2)
    //
    // 测量值与常量的叉积。//测量值叉乘常量值,并以此向量表示误差角度大小与转轴方向，用于修正陀螺仪积分角度
    float x_ca = y_aa * z_ac - z_aa * y_ac;
    float y_ca = z_aa * x_ac - x_aa * z_ac;
    float z_ca = x_aa * y_ac - y_aa * x_ac;
    //
    // 构造增量旋转。//可看成分别绕 xyz 轴的三次旋转的叠加。sin(delta/2)近似为 delta/2,cos(delta/2)近似为 0
    float delta_x = gyr[0] * interval / 2 + x_ca * FACTOR;//绕 x 轴旋转角度的一半，记 d_x  看作绕 x 轴的一次旋转：w=1,x=d_x,y=0,z=0
    float delta_y = gyr[1] * interval / 2 + y_ca * FACTOR;//绕 y 轴旋转角度的一半，记 d_y  看作绕 y 轴的一次旋转：w=1,x=0,y=d_y,z=0
    float delta_z = gyr[2] * interval / 2 + z_ca * FACTOR;//绕 z 轴旋转角度的一半，记 d_z  看作绕 z 轴的一次旋转：w=1,x=0,y=0,z=d_z
    //三次旋转叠加为一次旋转，即三个四元数相乘
    //四元数乘法公式：q3=q1*q2
```

圆点博士小四轴 叉积法融合陀螺和加速度 核心程序的较详细注释（可能有误、多多指教） by 司马青衫

```
    //(w1*w2 - x1*x2 - y1*y2 - z1*z2) = w3
    //(w1*x2 + x1*w2 + y1*z2 - z1*y2) = x3
    //(w1*y2 - x1*z2 + y1*w2 + z1*x2) = y3
    //(w1*z2 + x1*y2 - y1*x2 + z1*w2) = z3
    //合成的一次旋转：
    // w=1     - d_x*d_y*d_z(多个小角度相乘，忽略，下同)   =1
    // x=d_x + d_y*d_z（忽略）                                  =d_x
    // y=d_y - d_x*d_z（忽略）                                 =d_y
    // z=d_z + d_x*d_y（忽略）                                  =d_z
    //
    // 融合，四元数乘法。//将上面合成的旋转四元数与之前的姿态四元数相乘，得到新的姿态四元数并归一化为单位四元数。
    attitude->w = w_q            - x_q*delta_x      - y_q*delta_y - z_q*delta_z;
    attitude->x = w_q*delta_x + x_q                 + y_q*delta_z - z_q*delta_y;
    attitude->y = w_q*delta_y - x_q*delta_z    + y_q                + z_q*delta_x;
    attitude->z = w_q*delta_z + x_q*delta_y    - y_q*delta_x    + z_q;
    quaternion_normalize(attitude);//四元数归一化
}
```