

四元数完全解析及资料汇总

本文原帖出自匿名四轴论坛，附件里的资源请到匿名论坛下载：

<http://www.anobbs.com/forum.php>

感谢匿名的开源分享，感谢群友的热心帮助。

说什么四元数完全解析其实都是前辈们的解析，小弟真心是一个搬砖的，搬得不好希望大神们给以批评和指正，在此谢过了。因为本人是小菜鸟一枚，对，最菜的那种菜鸟……所以对四元数求解姿态角这么一个在大神眼里简单的算法，小弟我还是费了很大劲才稍微理解了那么一点点，小弟搬砖整理时也是基于小弟的理解和智商的，有些太基础，有些可能错了，大牛们发现了再骂过我后希望能够给与指正哈。

好，废话到此为止，开始说主体。四元数和姿态角怎么说呢？先得给和我一样的小菜鸟们理一理思路，小鸟我在此画了一个“思维导图”（我承认我画的丑），四元数解算姿态首先分为两部分理解：第一部分先理解什么是四元数，四元数与姿态角间的关系；第二部分要理解怎么由惯性单元测出的加速度和角速度求出四元数，再由四元数求出欧拉角。

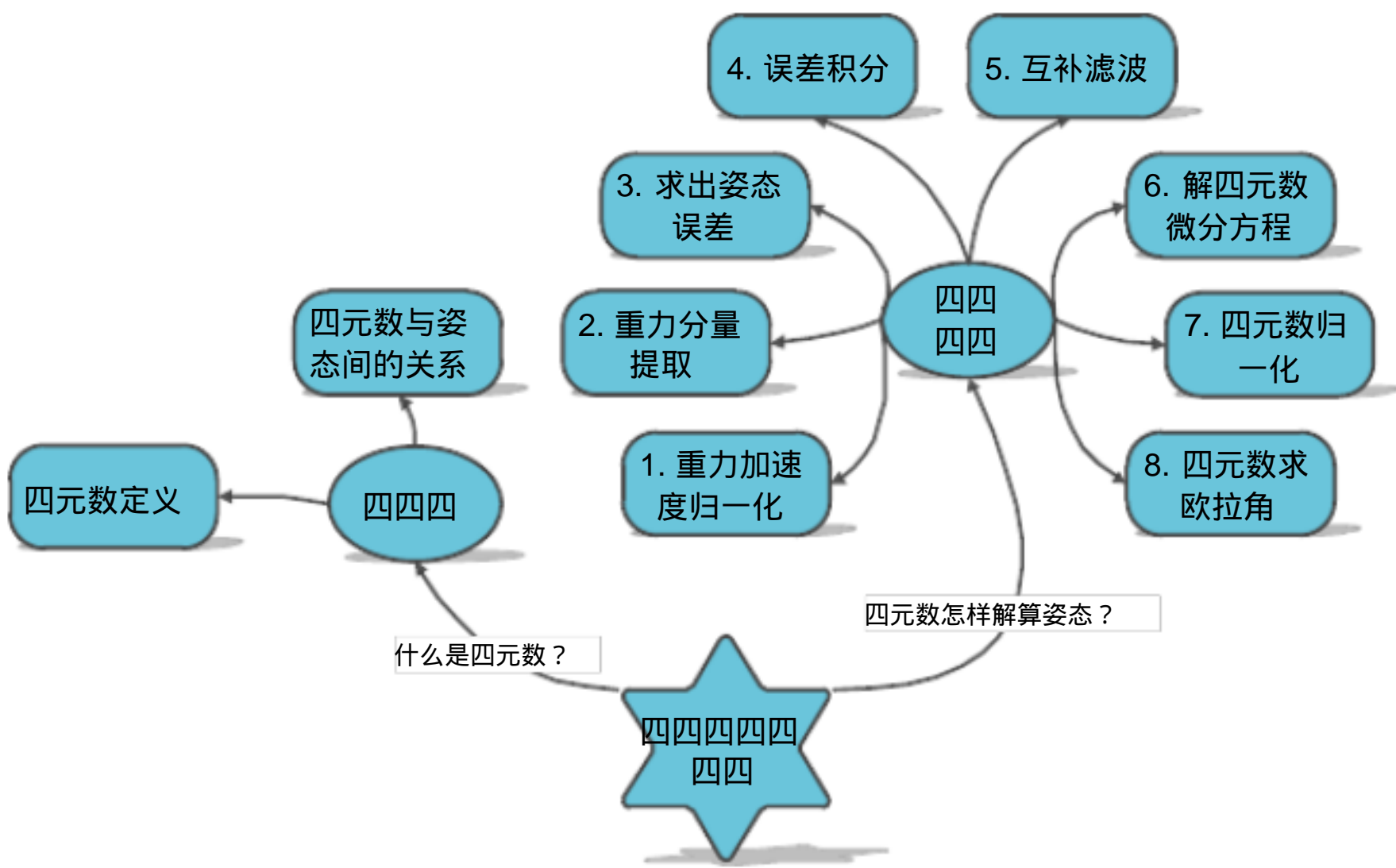


图 1 渣渣思维导图

在讲解什么是四元数时，小弟的思维是顺着说的，先由四元数的定义说起，说到四元数与姿态角间的关系。但在讲解姿态解算时，小弟的思维是逆向的，就是反推回来的，从欧拉角一步步

反推回到惯性器件的测量数据，这样逆向说是因为便于理解，因为实际在工程应用时和理论推导有很大差别。

实际应用时正确的求解顺序应该为图 1 中序号顺序，即 1->2->3->..... .

但在笔者讲解姿态求解时思路是如图 2 的。

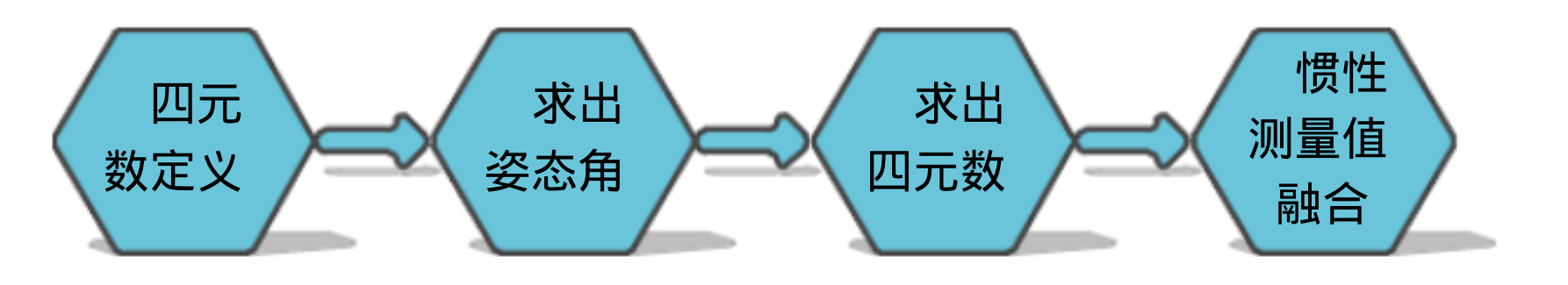


图 2 逆向讲解思路

大家在看四元数时最好结合着代码一块看，小弟看的是 匿名四轴 的代码，感觉写的非常好也非常清晰，粘出来大家一块观摩。红色部分是核心代码，总共分为八个步骤，和图 1 中的八个步骤是一一对应的。讲解介绍时也是和代码对比起来讲解的。代码可以去匿名官网上下载，都是开源的，不是小弟的，所以小弟不方便加在附件中。

```
//四元数更新姿态
#define Kp 2.0f //加速度权重，越大则向加速度测量值收敛越快
#define Ki 0.001f //误差积分增益
void ANO_IMU::Quaternion_CF(Vector3f gyro,Vector3f acc, float deltaT)
{
    Vector3f V_gravity, V_error, V_error_I;
    //1.重力加速度归一化
    acc.normalize();
    //2.提取四元数的等效余弦矩阵中的重力分量
    Q.vector_gravity(V_gravity);
    //3.向量叉积得出姿态误差
    V_error = acc % V_gravity;
    //4.对误差进行积分
    V_error_I += V_error * Ki;
    //5.互补滤波，姿态误差补偿到角速度上，修正角速度积分漂移
    Gyro += V_error * Kp + V_error_I;
    //6.一阶龙格库塔法更新四元数
    Q.Runge_Kutta_1st(Gyro, deltaT);
    //7.四元数归一化
    Q.normalize();
    //8.四元数转欧拉角
    Q.to_euler(&angle.x, &angle.y, &angle.z);
}
```

好的，下面搬砖开始！。。。。。。嘿咻嘿咻！！！！

一．什么是四元数？

关于四元数的定义摘自 秦永元的《惯性导航》 ，里面有非常好的讲解， 大家可以直接 看绪论和第九章 就可以。下面我粘贴了部分原文，粘贴的比较多比较详细，应为本人比较笨还爱较真，所以按本人的风格就要详尽一点，大牛们都可以自动忽略。

四元数定义、表达方式及运算方法——摘自《惯性导航》

-秦永元 P289-292

9.2 姿态更新计算的四元数算法

设由运载体的机体轴确定的坐标系为 b ，惯导系统所采用的导航坐标系为 n ，则由 b 系到 n 系的坐标变换矩阵 C_b^n 称为运载体的姿态矩阵。姿态更新是指根据惯性器件的输出实时计算出 C_b^n 矩阵。由于 n 系和 b 系均为直角坐标系，各轴之间始终保持直角，所以可将坐标系理解成刚体，当只研究两个坐标系间的角位置关系

时，可对一个坐标系作平移，使其原点与另一个坐标系的原点重合。因此，两坐标系间的空间角位置关系可理解成刚体的定点转动。从这一基本思想出发，可获得姿态更新的四元数算法及旋转矢量算法。本节详细介绍四元数更新算法。

9.2.1 四元数

1. 四元数定义

顾名思义，四元数是由四个元构成的数：

$$Q(q_0, q_1, q_2, q_3) = q_0 + q_1 i + q_2 j + q_3 k \quad (9.2.1)$$

其中， q_0, q_1, q_2, q_3 是实数， i, j, k 既是互相正交的单位向量，又是虚单位 $\sqrt{-1}$ ，具体规定体现在如下四元数乘法关系中：

$$\left. \begin{aligned} i \otimes i &= -1, & j \otimes j &= -1, & k \otimes k &= -1 \\ i \otimes j &= k, & j \otimes k &= i, & k \otimes i &= j \\ j \otimes i &= -k, & k \otimes j &= -i, & i \otimes k &= -j \end{aligned} \right\} \quad (9.2.2)$$

式中， \otimes 表示四元数乘法。

上述关系可叙述为：相同单位向量作四元数乘时呈虚单位特性；相异单位向量作四元数乘时呈单位向量叉乘特性。所以四元数既可看作四维空间中的一个向量，又可看作一个超复数。

2. 四元数的表达方式

(1) 矢量式

$$Q = q_0 + q \quad (9.2.3)$$

其中, q_0 称四元数 Q 的标量部分, q 称四元数 Q 的矢量部分。对照式(9.2.1), 可看出 q 是三维空间中的一个向量。

(2) 复数式

$$Q = q_0 + q_1 i + q_2 j + q_3 k \quad (9.2.4)$$

可视为一个超复数, Q 的共轭复数记为

$$Q^* = q_0 - q_1 i - q_2 j - q_3 k \quad (9.2.5)$$

Q^* 称为 Q 的共轭四元数。

(3) 三角式

$$Q = \cos \frac{\theta}{2} + u \sin \frac{\theta}{2} \quad (9.2.6)$$

式中, θ 为实数, u 为单位向量。

(4) 指数式

$$Q = e^{u \frac{\theta}{2}} \quad (9.2.7)$$

θ 和 u 同上。

• 289 •

(5) 矩阵式

$$Q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (9.2.8)$$

3. 四元数的大小——范数

四元数的大小用四元数的范数来表示:

$$\|Q\| = q_0^2 + q_1^2 + q_2^2 + q_3^2 \quad (9.2.9)$$

若 $\|Q\| = 1$, 则 Q 称为规范化四元数。

4. 四元数的运算——加减乘除

1) 加法和减法

设

$$Q = q_0 + q_1 i + q_2 j + q_3 k$$

$$P = p_0 + p_1 i + p_2 j + p_3 k$$

则

$$Q \pm P = (q_0 \pm p_0) + (q_1 \pm p_1)i + (q_2 \pm p_2)j + (q_3 \pm p_3)k \quad (9.2.10)$$

2) 乘法

$$aQ = aq_0 + aq_1i + aq_2j + aq_3k \quad (9.2.11)$$

其中, a 为标量。

$$\begin{aligned} P \otimes Q &= (p_0 + p_1i + p_2j + p_3k) \otimes (q_0 + q_1i + q_2j + q_3k) \\ &= (p_0q_0 - p_1q_1 - p_2q_2 - p_3q_3) + (p_0q_1 + p_1q_0 + p_2q_3 - p_3q_2)i \\ &\quad + (p_0q_2 + p_2q_0 + p_3q_1 - p_1q_3)j + (p_0q_3 + p_3q_0 + p_1q_2 - p_2q_1)k \\ &= r_0 + r_1i + r_2j + r_3k \end{aligned} \quad (9.2.12)$$

上式写成矩阵形式:

$$\begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & -p_3 & p_2 \\ p_2 & p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = M(P)Q \quad (9.2.13)$$

或

$$\begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} = M'(Q)P \quad (9.2.14)$$

• 290 •

其中, $M(P)$ 的构成形式为: 第一列是四元数 P 本身, 第一行是 P 的共轭四元数 P^* 的转置, 划去第一行和第一列余下的部分。

$$V_P = \begin{bmatrix} p_0 & -p_3 & p_2 \\ p_3 & p_0 & -p_1 \\ -p_2 & p_1 & p_0 \end{bmatrix} \quad (9.2.15)$$

称为 $M(P)$ 的核, 是由四元数 P 的元构成的反对称矩阵。同理 $M'(Q)$ 的核为

$$V'_Q = \begin{bmatrix} q_0 & q_3 & -q_2 \\ -q_3 & q_0 & q_1 \\ q_2 & -q_1 & q_0 \end{bmatrix} \quad (9.2.16)$$

可见 $M'(Q)$ 与 $M(Q)$ 构成相似, 但核不同。

由式 (9.2.13) 和式 (9.2.14), 得四元数乘法的矩阵表示形式:

$$P \otimes Q = M(P)Q \quad (9.2.17a)$$

$$P \otimes Q = M'(Q)P \quad (9.2.17b)$$

由于 $M(P)$ 和 $M'(P)$ 的核不同, 所以

$$P \otimes Q = M(P)Q \neq M'(P)Q = Q \otimes P$$

上式说明四元数乘法不满足交换律。

四元数乘法满足分配律和结合律:

$$P \otimes (Q + R) = P \otimes Q + P \otimes R \quad (9.2.18)$$

$$P \otimes Q \otimes R = (P \otimes Q) \otimes R = P \otimes (Q \otimes R) \quad (9.2.19)$$

3) 除法——求逆

如果 $P \otimes R = 1$, 则称 R 为 P 的逆, 记为 $R = P^{-1}$, 或称 P 为 R 的逆, 记为 $P = R^{-1}$ 。

根据范数定义和式(9.2.12)

$$\begin{aligned} P \otimes P^* &= (p_0 + p_1 i + p_2 j + p_3 k) \otimes (p_0 - p_1 i - p_2 j - p_3 k) \\ &= p_0^2 + p_1^2 + p_2^2 + p_3^2 \\ &= \|P\| \end{aligned}$$

所以, $P \otimes \frac{P^*}{\|P\|} = 1$, 根据上述关于逆的定义, $\frac{P^*}{\|P\|}$ 即为 P 的逆, 即

$$P^{-1} = \frac{P^*}{\|P\|} \quad (9.2.22)$$

好, 关于四元数定义就搬这么多, 其他的大家去附件下载《惯性导航》的

pdf 自己看吧。

下面开始搬四元数与姿态解算关系的。。。。。嘿咻嘿咻

~~~~

## 二、四元数与姿态阵间的关系

从上面我们知道了四元数的定义，可这四个数和我们要求的三个姿态角有什么关系呢？下面是详细的推导，同样摘自《惯性导航》 -秦永元 P292-297。

四元数与姿态阵间的关系——摘自《惯性导航》 -秦永元 P292-297

### 9.2.2 四元数与姿态阵间的关系

设有参考坐标系  $R$ , 坐标轴为  $x_0, y_0, z_0$ , 坐标轴方向的单位向量为  $i_0, j_0, k_0$ 。刚体相对  $R$  系作定点转动, 定点为  $O$ 。取坐标系  $b$  与刚体固联,  $b$  系的坐标轴为  $x, y, z$ , 坐标轴方向的单位向量为  $i, j, k$ 。假设初始时刻  $b$  系与  $R$  系重合。为了便于分析刚体的空间角位置, 在刚体上取一点  $A$ , 转动点  $O$  至该点引位置向量  $OA$ , 如图 9.2.1 所示。则该位置向量的空间位置实际上描述了刚体的空间角位置。

设刚体以  $\omega = \omega_x i + \omega_y j + \omega_z k$  相对  $R$  系旋转, 初始时刻位置向量处于  $OA = r$ , 经过时间  $t$  后位置向量处于  $OA' = r'$ 。根据欧拉定理, 仅考虑刚体在 0 时刻和  $t$  时刻的角位置时, 刚体从  $A$  位置转到  $A'$  位置的转动可等效成绕瞬轴  $u$  (单位向量) 转过  $\theta$  角一次完成。这样, 位置向量做圆锥运动,  $A$  和  $A'$  位于同一圆上,  $r$  和  $r'$  位于同一圆锥面上。

下面分析  $r'$  与  $r$  的关系。在圆上取一点  $B$ , 使  $\angle AO'B = 90^\circ$ , 由图得

$$OO' = (r \cdot u)u$$

$$O'A = r - OO' = r - (r \cdot u)u$$

$$O'B = u \times O'A$$

$$= u \times r - (r \cdot u)u \times u = u \times r$$

$$O'A' = O'A \cos \theta + O'B \sin \theta$$

$$= r \cos \theta - (r \cdot u)u \cos \theta + u \times r \sin \theta$$

所以

$$r' = OO' + O'A' = r \cos \theta + (1 - \cos \theta) \times (r \cdot u)u + u \times r \sin \theta$$

由三重矢积计算公式:

$$\begin{aligned} u \times (u \times r) &= u(u \cdot r) - (u \cdot u)r \\ &= (r \cdot u)u - r \end{aligned}$$

即

$$(r \cdot u)u = r + u \times (u \times r)$$

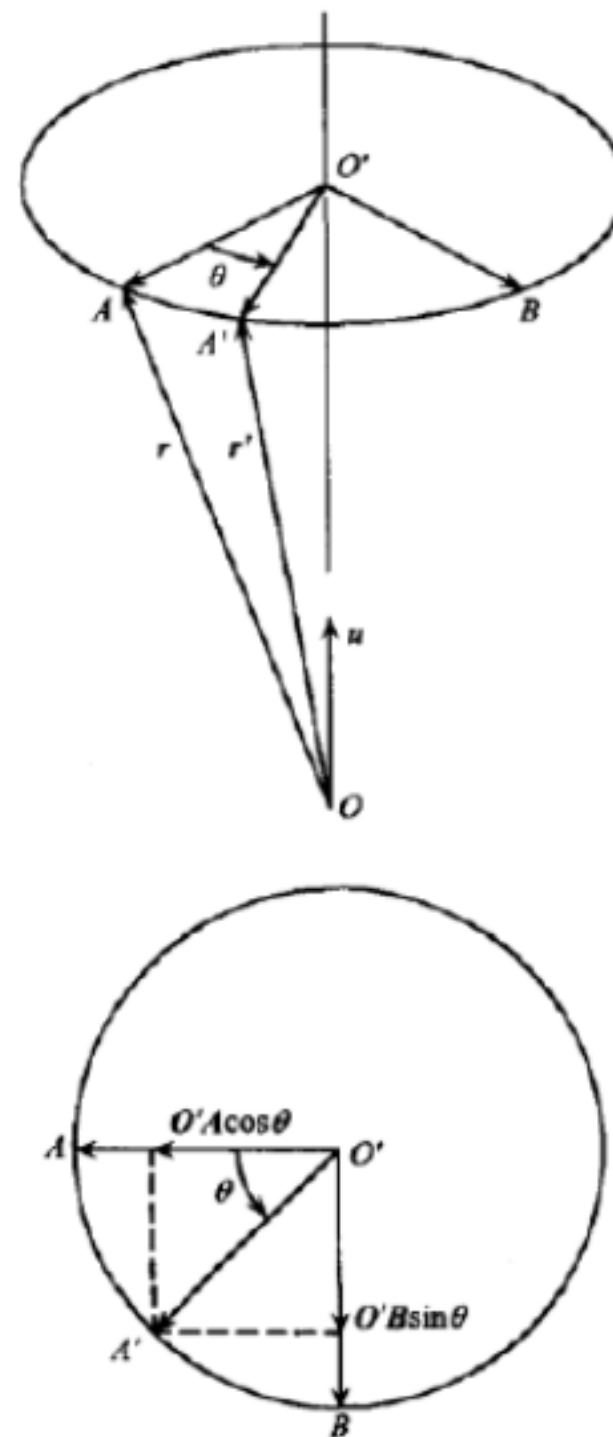


图 9.2.1 刚体的等效旋转

所以

$$\begin{aligned} \boldsymbol{r}' &= r \cos \theta + (1 - \cos \theta) [\boldsymbol{r} + \boldsymbol{u} \times (\boldsymbol{u} \times \boldsymbol{r})] + \boldsymbol{u} \times \boldsymbol{r} \sin \theta \\ &= \boldsymbol{r} + \boldsymbol{u} \times \boldsymbol{r} \sin \theta + (1 - \cos \theta) \boldsymbol{u} \times (\boldsymbol{u} \times \boldsymbol{r}) \end{aligned}$$

将上式向  $R$  系内投影:

$$\begin{aligned} \boldsymbol{r}'^R &= \boldsymbol{r}^R + (\boldsymbol{u} \times \boldsymbol{r})^R \sin \theta \\ &\quad + (1 - \cos \theta) [\boldsymbol{u} \times (\boldsymbol{u} \times \boldsymbol{r})]^R \end{aligned}$$

记

$$\boldsymbol{r}'^R = \begin{bmatrix} r'_x \\ r'_y \\ r'_z \end{bmatrix}, \quad \boldsymbol{r}^R = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}, \quad \boldsymbol{u}^R = \begin{bmatrix} l \\ m \\ n \end{bmatrix}$$

又根据叉乘关系表达式:

$$(\boldsymbol{u} \times \boldsymbol{r})^R = \begin{bmatrix} 0 & -n & m \\ n & 0 & -l \\ -m & l & 0 \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}$$

记

$$\boldsymbol{U} = \begin{bmatrix} 0 & -n & m \\ n & 0 & -l \\ -m & l & 0 \end{bmatrix} \quad (9.2.23)$$

则

$$\begin{aligned} (\boldsymbol{u} \times \boldsymbol{r})^R &= \boldsymbol{U} \boldsymbol{r}^R \\ [\boldsymbol{u} \times (\boldsymbol{u} \times \boldsymbol{r})]^R &= \boldsymbol{U} \cdot \boldsymbol{U} \boldsymbol{r}^R \end{aligned}$$

所以

$$\begin{aligned} \boldsymbol{r}'^R &= \boldsymbol{r}^R + \boldsymbol{U} \boldsymbol{r}^R \sin \theta + (1 - \cos \theta) \boldsymbol{U} \cdot \boldsymbol{U} \boldsymbol{r}^R \\ &= \left( \boldsymbol{I} + 2\boldsymbol{U} \sin \frac{\theta}{2} \cos \frac{\theta}{2} + 2 \sin^2 \frac{\theta}{2} \boldsymbol{U} \cdot \boldsymbol{U} \right) \boldsymbol{r}^R \end{aligned} \quad (9.2.24)$$

令

$$\boldsymbol{D} = \boldsymbol{I} + 2\boldsymbol{U} \sin \frac{\theta}{2} \cos \frac{\theta}{2} + 2 \sin^2 \frac{\theta}{2} \boldsymbol{U} \cdot \boldsymbol{U} \quad (9.2.25)$$

则式(9.2.24)可写成:

$$\boldsymbol{r}'^R = \boldsymbol{D} \boldsymbol{r}^R \quad (9.2.26)$$

记初始时刻的刚体固联坐标系为  $b_0$ , 由于初始时刻刚体固联坐标系与参考坐标系重合, 所以

$$\boldsymbol{r}^R = \boldsymbol{r}^{b_0} \quad (9.2.27)$$

而在转动过程中, 位置向量和  $b$  系都同刚体固联, 所以位置向量和  $b$  系的相对角位置始终不变, 即有

$$\boldsymbol{r}^{b_0} = \boldsymbol{r}'^b \quad (9.2.28)$$

将式(9.2.28)代入式(9.2.27), 得

$$\boldsymbol{r}^R = \boldsymbol{r}'^b \quad (9.2.29)$$

将式(9.2.29)代入式(9.2.26), 得

$$\boldsymbol{r}'^R = \boldsymbol{D} \boldsymbol{r}'^b$$



该式说明  $D$  即为  $b$  系至  $R$  系的坐标变换矩阵, 根据式(9.2.25)和式(9.2.23)

$$C_b^R = D = I + 2U \sin \frac{\theta}{2} \cos \frac{\theta}{2} + 2 \sin^2 \frac{\theta}{2} U \cdot U \quad (9.2.30a)$$

即

$$C_b^R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + 2 \cos \frac{\theta}{2} \begin{bmatrix} 0 & -n \sin \frac{\theta}{2} & m \sin \frac{\theta}{2} \\ n \sin \frac{\theta}{2} & 0 & -l \sin \frac{\theta}{2} \\ -m \sin \frac{\theta}{2} & l \sin \frac{\theta}{2} & 0 \end{bmatrix}$$

$$+ 2 \begin{bmatrix} -(m^2 + n^2) \sin^2 \frac{\theta}{2} & l m \sin^2 \frac{\theta}{2} & l n \sin^2 \frac{\theta}{2} \\ l m \sin^2 \frac{\theta}{2} & -(l^2 + n^2) \sin^2 \frac{\theta}{2} & m n \sin^2 \frac{\theta}{2} \\ l n \sin^2 \frac{\theta}{2} & m n \sin^2 \frac{\theta}{2} & -(m^2 + l^2) \sin^2 \frac{\theta}{2} \end{bmatrix} \quad (9.2.30b)$$

令

$$\begin{cases} q_0 = \cos \frac{\theta}{2} \\ q_1 = l \sin \frac{\theta}{2} \\ q_2 = m \sin \frac{\theta}{2} \\ q_3 = n \sin \frac{\theta}{2} \end{cases} \quad (9.2.31)$$

并以  $q_0, q_1, q_2, q_3$  构造四元数:

$$\begin{aligned} Q &= q_0 + q_1 i_0 + q_2 j_0 + q_3 k_0 = \cos \frac{\theta}{2} + (l i_0 + m j_0 + n k_0) \sin \frac{\theta}{2} \\ &= \cos \frac{\theta}{2} + u^R \sin \frac{\theta}{2} \end{aligned} \quad (9.2.32)$$

则可得如下结论:

(1) 四元数  $Q = \cos \frac{\theta}{2} + u^R \sin \frac{\theta}{2}$  描述了刚体的定点转动, 即当只关心  $b$  系相对  $R$  系的角位置时, 可认为  $b$  系是由  $R$  系经过无中间过程的一次性等效旋转形成的,  $Q$  包含了这种等效旋转的全部信息;  $u^R$  为旋转瞬轴和旋转方向,  $\theta$  为转过的角度。

(2) 四元数可确定出  $b$  系至  $R$  系的坐标变换矩阵。将式(9.2.31)代入式(9.2.30), 得

$$C_b^R = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & 1 - 2(q_1^2 + q_3^2) & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix} \quad (9.2.33)$$

由于  $\|Q\| = q_0^2 + q_1^2 + q_2^2 + q_3^2 = \cos^2 \frac{\theta}{2} + (l^2 + m^2 + n^2) \sin^2 \frac{\theta}{2} = 1$ , 所以可进一步推得如下结论:

(1) 描述刚体旋转的四元数是规范化四元数。

(2) 式(9.2.33)可写成:

$$C_b^R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (9.2.34)$$

如果参考坐标系  $R$  是导航坐标系  $n$ , 刚体固联坐标系  $b$  为机体坐标系, 则坐标变换阵  $C_b^R$  就是姿态矩阵  $C_b^n$ , 而由姿态矩阵可计算出航向角和姿态角。

设运载体的航向角为  $\Psi$  (习惯上以北偏东为正), 俯仰角为  $\theta$ , 横滚角为  $\gamma$ , 取地理坐标系  $g$  为导航坐标系, 并规定  $x_g, y_g, z_g$  的指向依次为东、北、天, 则机体坐标系  $b$  与导航坐标系  $n$  (即地理坐标系  $g$ ) 的关系如图 1.2.3 所示。

由该图可得三次基本旋转对应的坐标变换阵为

$$C_n^i = C_g^i = \begin{bmatrix} \cos\Psi & -\sin\Psi & 0 \\ \sin\Psi & \cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (9.2.37)$$

$$\begin{aligned} C_1^b = C_2^b C_1^i &= \begin{bmatrix} \cos\gamma & 0 & -\sin\gamma \\ 0 & 1 & 0 \\ \sin\gamma & 0 & \cos\gamma \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \\ &= \begin{bmatrix} \cos\gamma & \sin\theta\sin\gamma & -\cos\theta\sin\gamma \\ 0 & \cos\theta & \sin\theta \\ \sin\gamma & -\sin\theta\cos\gamma & \cos\theta\cos\gamma \end{bmatrix} \end{aligned} \quad (9.2.38)$$

$$\begin{aligned} C_n^b &= C_1^b \cdot C_n^i \\ &= \begin{bmatrix} \cos\gamma\cos\Psi + \sin\gamma\sin\Psi\sin\theta & -\cos\gamma\sin\Psi + \sin\gamma\cos\Psi\sin\theta & -\sin\gamma\cos\theta \\ \sin\Psi\cos\theta & \cos\Psi\cos\theta & \sin\theta \\ \sin\gamma\cos\Psi - \cos\gamma\sin\Psi\sin\theta & -\sin\gamma\sin\Psi - \cos\gamma\cos\Psi\sin\theta & \cos\gamma\cos\theta \end{bmatrix} \end{aligned} \quad (9.2.39)$$

记  $C_b^n = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix}$ , 由于  $n$  系至  $b$  系的旋转过程中坐标系始终保持直角坐标系, 所以  $C_b^n$  为正交矩阵

$$C_n^b = (C_b^n)^T = \begin{bmatrix} T_{11} & T_{21} & T_{31} \\ T_{12} & T_{22} & T_{32} \\ T_{13} & T_{23} & T_{33} \end{bmatrix} \quad (9.2.40)$$

比较式(9.2.39)和式(9.2.40),得

$$\begin{cases} \theta = \arcsin(T_{32}) \\ \gamma_{\pm} = \arctan\left(-\frac{T_{31}}{T_{33}}\right) \\ \Psi_{\pm} = \arctan\left(\frac{T_{12}}{T_{22}}\right) \end{cases} \quad (9.2.41)$$

航向角和横滚角的真值按表 9.2.1 和表 9.2.2 确定。

表 9.2.1 航向角 $\Psi$ 真值表

| $T_{22}$        | $T_{12}$ | $\Psi$                   |
|-----------------|----------|--------------------------|
| $\rightarrow 0$ | +        | $90^\circ$               |
| $\rightarrow 0$ | -        | $-90^\circ$              |
| +               | +        | $\Psi_{\pm}$             |
| +               | -        | $\Psi_{\pm}$             |
| -               | +        | $\Psi_{\pm} + 180^\circ$ |
| -               | -        | $\Psi_{\pm} - 180^\circ$ |

表 9.2.2 横滚角 $\gamma$ 真值表

| $\gamma_{\pm}$ | $T_{33}$ | $\gamma$                   |
|----------------|----------|----------------------------|
| +              | +        | $\gamma_{\pm}$             |
| -              |          |                            |
| +              | -        | $\gamma_{\pm} - 180^\circ$ |
| -              | -        | $\gamma_{\pm} + 180^\circ$ |

上述分析说明:如果表征 $n$ 系至 $b$ 系的旋转四元数 $Q$ 已确定,则按式(9.2.33)或式(9.2.34)可计算出姿态阵 $C_b^n$ ,再按式(9.2.41)和表 9.2.1 及表 9.2.2 可确定出运载体的航向角、俯仰角和横滚角,因此,四元数 $Q$ 包含了所有的姿态信息,捷联惯导中的姿态更新实质上是如何计算四元数 $Q$ 。

呃,粘了这么多其实就是为了想知道推导过程小伙伴好理解,真正有用的就是(9.2.34)这个公式。这个公式把四元数转换成了方向余弦矩阵中的几个元素,再用这几个元素转换为欧拉角。就求解除了姿态!

$$C_b^n = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (9.2.34)$$

先从四元数  $q_0 \sim q_3$  转成方向余弦矩阵:

$$\begin{matrix} x_{??} & q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) & ?? \\ ?? = & 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) & ?? \\ ?? & 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 & ?? \end{matrix}$$

再从方向余弦矩阵转换为欧拉角:

$$\theta = \sin^{-1}(T_{32}) \quad \text{—— 俯仰角 } \theta$$

$$\gamma = \tan^{-1}\left(-\frac{T_{31}}{T_{33}}\right) \quad \text{—— 横滚角 } \gamma$$

$$\Psi = \tan^{-1}\left(\frac{T_{12}}{T_{22}}\right) \quad \text{—— 航向角 } \Psi$$

好的，公式原理都讲清楚了，下面来看一下匿名的代码：

```
//四元数转欧拉角， 这里四元数是 q1~q4 和公式里 q0~q3 相对应。
void Quaternion::to_euler(float *roll, float *pitch, float *yaw)
{
    if (roll) {
        *roll = degrees(atan2f(2.0f*(q1*q2 + q3*q4), 1 - 2.0f*(q2*q2 + q3*q3)));
        /*roll = degrees(atan2f(-2.0f*(q2*q4 - q1*q3), 1 - 2.0f*(q2*q2 + q3*q3)));
    }
    if (pitch) {
        // 使用 safe_asin() 来处理 pitch 接近 90/-90 时的奇点
        *pitch = degrees(safe_asin(2.0f*(q1*q3 - q2*q4)));
        /*pitch = degrees(safe_asin(2.0f*(q3*q4 + q1*q2)));
    }
    if (yaw) {
        *yaw = degrees(atan2f(2.0f*(q2*q3 - q1*q4), 2.0f*(q1*q1 + q2*q2) - 1));
        /*yaw = degrees(atan2f(2.0f*(q2*q3 - q1*q4), 2.0f*(q1*q1 + q3*q3) - 1));
    }
}
```

对比余弦矩阵转换为欧拉角的公式很容易理解了吧，注意一下，红色是匿名原版的代码，和公式还是有出入的，自己细心观察一下吧。被注释了的黑色代码是我根据上面的公式写的。但黑色的求解出来的欧拉角反映出来的姿态是不对的，具体表现为俯仰（pitch）和横滚（roll）是相反的，为啥根据公式写的是不对的？群里的小伙伴给与了我热心的解答。

这个错误主要是由于方向余弦矩阵的旋转顺序不一样，也就是公式（9.2.39）不一样，这是由于旋转的先后顺序不同引起的，具体大家参考《惯性导航》绪论来看就能明白，因为这一点小弟还有点混乱，就点到这为止。

$$\begin{aligned} C_n^s &= C_1^s \cdot C_n^1 \\ &= \begin{bmatrix} \cos\gamma\cos\Psi + \sin\gamma\sin\Psi\sin\theta & -\cos\gamma\sin\Psi + \sin\gamma\cos\Psi\sin\theta & -\sin\gamma\cos\theta \\ \sin\Psi\cos\theta & \cos\Psi\cos\theta & \sin\theta \\ \sin\gamma\cos\Psi - \cos\gamma\sin\Psi\sin\theta & -\sin\gamma\sin\Psi - \cos\gamma\cos\Psi\sin\theta & \cos\gamma\cos\theta \end{bmatrix} \end{aligned} \quad (9.2.39)$$

以上就是四元数求解欧拉角的方法。通过公式可以看到，要求欧拉角需要求得四元数的方向余弦矩阵；要求得四元数方向余弦矩阵，需要求得四元数  $q_0 \sim q_3$ ，那么如何求得  $q_0 \sim q_3$ ？接下来详细介绍。



### 三、四元数微分及龙格库塔求 Q0~Q3

首先我们先来看一下在程序里如何求解的 q0~q3：

```
//一阶龙格库塔法更新四元数
void Quaternion::Runge_Kutta_1st(Vector3f &g, float deltaT)
{
    q1 += 0.5 * (-q2*g.x - q3*g.y - q4*g.z)* deltaT;
    q2 += 0.5 * (q1*g.x + q3*g.z - q4*g.y)* deltaT;
    q3 += 0.5 * (q1*g.y - q2*g.z + q4*g.x)* deltaT;
    q4 += 0.5 * (q1*g.z + q2*g.y - q3*g.x)* deltaT;
}
```

这就是一阶龙格库塔法求解 q 的微分方程，传入参数只需要这个周期的角速度 g.x、g.y、g.z 和周期时间 deltaT。下面一张是从某位大神的贴吧上盗的图，描绘的是一阶龙格库塔的计算式。

其一阶龙格 - 库塔法计算式为

$$q(t+T) = q(t) + T\Omega_b(t)q(t)$$

式中的  $\Omega_b(t)$  如式(7.2.19) 所示。

将式(7.3.25) 展开成元素的表达式, 有

$$\begin{aligned}\lambda(t+T) &= \lambda(t) + \frac{T}{2}[-\omega_x(t)P_1(t) - \omega_y(t)P_2(t) - \omega_z(t)P_3(t)] \\ P_1(t+T) &= P_1(t) + \frac{T}{2}[\omega_x(t)\lambda(t) + \omega_z(t)P_2(t) - \omega_y(t)P_3(t)] \\ P_2(t+T) &= P_2(t) + \frac{T}{2}[\omega_y(t)\lambda(t) - \omega_z(t)P_1(t) + \omega_x(t)P_3(t)] \\ P_3(t+T) &= P_3(t) + \frac{T}{2}[\omega_z(t)\lambda(t) + \omega_y(t)P_1(t) - \omega_x(t)P_2(t)]\end{aligned}$$

相信很多人和我一样，单看上图很难理解其中的意思和其由来，于是我又找了很多帖子，感谢前人做出的贡献，小弟在这里再次整理大神的四元数微分方程推导公式，便于大家理解。 摘自附件中 《推导\_四元数.pdf》

虽然在下也不是很懂，不过粘出来还是能起到理解的作用，这样大家就不会觉得这是凭空变出来的，本人数学功底薄弱，没有对推导进行过验证，如果有不对的地方欢迎指正。

接着使用一阶龙格库塔（Runge-Kutta）发求出 q0~q3，这一点很多人不知道一阶龙格库塔怎么推导的，下面也是这位网友的推导，大家参考着理解吧。

### 3. 四元數微分

四元數微分, 已知一四元數  $Q = \cos[\frac{\theta}{2}] + \hat{n} \cdot \sin[\frac{\theta}{2}]$ , 對時間微分

$$\frac{dQ}{dt} = -\frac{1}{2} \sin[\frac{\theta}{2}] \cdot \frac{d\theta}{dt} + \frac{d\hat{n}}{dt} \cdot \sin[\frac{\theta}{2}] + \hat{n} \cdot \frac{1}{2} \cos[\frac{\theta}{2}] \cdot \frac{d\theta}{dt}$$

已知  $\hat{n} \cdot \hat{n} = -1$ ,  $\frac{d\hat{n}}{dt} = 0$ ,  $\frac{d\theta}{dt} = \omega_{Eb}^E$ , E 為地理座標系, b 為飛行器坐標系

$$\frac{dQ}{dt} = \frac{1}{2} \hat{n} \cdot \omega_{Eb}^E \cdot (\cos[\frac{\theta}{2}] + \hat{n} \cdot \sin[\frac{\theta}{2}]) = \frac{1}{2} \vec{\omega}_{Eb}^E \cdot Q$$

因為陀螺儀在飛行器上測到的角速度為  $\vec{\omega}_{Eb}^b = \omega_x \hat{i} + \omega_y \hat{j} + \omega_z \hat{k}$ , 故將  $\vec{\omega}_{Eb}^E$  轉換成  $\vec{\omega}_{Eb}^b$  會較為方便

$$\begin{aligned} \vec{\omega}_{Eb}^b &= Q^* \vec{\omega}_{Eb}^E \cdot Q \Rightarrow Q \cdot \vec{\omega}_{Eb}^b = Q \cdot Q^* \cdot \vec{\omega}_{Eb}^E \cdot Q = \vec{\omega}_{Eb}^E \cdot Q \\ &\Rightarrow \frac{dQ}{dt} = \frac{1}{2} \vec{\omega}_{Eb}^E \cdot Q = \frac{1}{2} Q \cdot \vec{\omega}_{Eb}^b \end{aligned}$$

將  $\frac{dQ}{dt} = \frac{1}{2} Q \cdot \vec{\omega}_{Eb}^b$  展開

$$\frac{dQ}{dt} = \frac{1}{2} (q_0 + q_1 \hat{i} + q_2 \hat{j} + q_3 \hat{k}) \cdot (\omega_x \hat{i} + \omega_y \hat{j} + \omega_z \hat{k}) = \frac{1}{2} \begin{pmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{pmatrix} \cdot \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix}$$

整理為

$$\frac{dQ}{dt} = \Omega_b \cdot Q$$

其中

$$\Omega_b = \frac{1}{2} \begin{pmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{pmatrix}$$

### 4. 更新四元數

使用一階 Runge-Kutta 更新四元數, 假設有一微分方程

$$\frac{dX}{dt} = f[X[t], \omega[t]]$$

則其解為

$$X[t + \Delta t] = X[t] + \Delta t \cdot f[X[t], \omega[t]]$$

其中  $\Delta t$  為取樣週期, 將套用至四元數

$$Q[t + \Delta t] = Q[t] + \Delta t \cdot \Omega_b[t] \cdot Q[t]$$

展開上式

$$\begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix}_{t+\Delta t} = \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix}_t + \frac{\Delta t}{2} \begin{pmatrix} -\omega_x \cdot q_1 - \omega_y \cdot q_2 - \omega_z \cdot q_3 \\ +\omega_x \cdot q_0 - \omega_y \cdot q_3 + \omega_z \cdot q_2 \\ +\omega_x \cdot q_3 + \omega_y \cdot q_0 - \omega_z \cdot q_1 \\ -\omega_x \cdot q_2 + \omega_y \cdot q_1 + \omega_z \cdot q_0 \end{pmatrix}$$

只需利用角速度即可更新四元數

这里的角速度  $\omega_x$   $\omega_y$   $\omega_z$ 是由捷联陀螺的输出（对机械转子陀螺必须经过误差补偿，将在下面介绍）。

对比着匿名四轴的代码看一看（`g.x`、`g.y`、`g.z` 是捷联陀螺的输出），代码的意思就比较清楚了。在往上一步步推，我们就要求陀螺输出了，并且还要对数据进行互补滤波处理。

#### 四、惯性单元测量值融合

这部分看似很简单，但是也有让笔者难以理解的地方，希望后人能补充修正进行更好的讲解。有了上一步的龙格库塔方程，我们现在需要的就是角速度的测量值。

在四轴上安装陀螺仪，可以测量四轴倾斜的角速度，由于陀螺仪输出的是四轴的角速度，不会受到四轴振动影响。因此该信号中噪声很小。四轴的角度又是通过对角速度积分而得，这可进行一步平滑信号，从而使得角度信号更加稳定。因此四轴控制所需要的角度和角速度可以使用陀螺仪所得到的信号。由于从陀螺仪的角速度获得角度信息，需要经过积分运算。如果角速度信号存在微小的偏差，经过积分运算之后，变化形成积累误差。这个误差会随着时间延长逐步增加，最终导致电路饱和，无法形成正确的角度信号。

如何消除这个累积误差呢？可以通过上面的加速度传感器获得的角度信息对此进行校正。利用加速度计所获得的角度信息  $g$  与陀螺仪积分后的角度进行比较，将比较的误差信号经过比例  $T_g$  放大之后与陀螺仪输出的角速度信号叠加之后再积分。对于加速度计给定的角度  $g$ ，经过比例、积分环节之后产生的角度必然最终等于  $g$ 。由于加速度计获得的角度信息不会存在积累误差，所以最终将输出角度中的积累误差消除了。加速度计所产生的角度信息  $g$  中会叠加很强的有四轴运动加速度噪声信号。为了避免该信号对于角度的影响，因此比例系数  $T_g$  应该非常小。这样，加速度的噪声信号经过比例、积分后，在输出角度信息中就会非常小了。由于存在积分环节，所以无论比例  $T_g$  多么小，最终输出角度必然与加速度计测量的角度  $g$  相等，只是这个调节过程会随着  $T_g$  的减小而延长。

先把这个过程的代码粘出来，看着代码一步步理解：

```
#define Kp 2.0f           //加速度权重，越大则向加速度测量值收敛越快
#define Ki 0.001f        //误差积分增益
//1.重力加速度归一化
acc.normalize();
//2.提取四元数的等效余弦矩阵中的重力分量
Q.vector_gravity(V_gravity);
//3.向量叉积得出姿态误差
V_error = acc % V_gravity;
//4.对误差进行积分
V_error_I += V_error * Ki;
//5.互补滤波，姿态误差补偿到角速度上，修正角速度积分漂移
Gyro += V_error * Kp + V_error_I;
```

1.重力加速度归一化： 加速度计数据归一化，把加速度计的三维向量转换为单位向量， 因为单位矢量到参考性的投影， 所以要把加速度计数据单位化， 其实归一化改变的只是这三个向量的长度，也就是只改变了相同的倍数，方向并没有改变，也是为了与单位四元数对应。

2.提取四元数的等效余弦矩阵中的重力分量：

```
// 返回该四元数的等效余弦矩阵中的重力分量
void Quaternion::vector_gravity(Vector3f &v)
{
    v.x = 2*(q2*q4 - q1*q3);
    v.y = 2*(q1*q2 + q3*q4);
    v.z = 1-2*(q2*q2 + q3*q3);
}
```

将当前姿态的重力在三个轴上的分量分离出来 ,把四元数换算成方向余弦中的第三行的三个元素，根据余弦矩阵和欧拉角的定义，就是地理坐标系 (参考坐标系) 的 Z 轴的重力向量。 当我读完这句话脑子挺懵的，闹不明白啊，于是又找到了下面的资料，可以进行解释了。

利用四元數將地理的重力加速度旋轉至飛行器上面，  
再與加速度計讀出的值(已歸一化的)做外積，得出誤差，  
用此誤差對角速度做校正融合

重力加速度

$$\vec{g} = \hat{g}^z$$

做歸一化

$$\vec{g} \rightarrow \hat{g} = \hat{z}$$

設旋轉至飛行器上的重力加速度為  $\hat{g}_b$

$$\begin{aligned}\hat{g}_b &= M_q \cdot \hat{g} = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \\ &\Rightarrow \hat{g}_b = \begin{pmatrix} g_{bx} \\ g_{by} \\ g_{bz} \end{pmatrix} = \begin{pmatrix} M_{13} \\ M_{23} \\ M_{33} \end{pmatrix}\end{aligned}$$

飛行器上的加速度計測得的加速度  $\vec{a}_b$  做歸一化

$$\vec{a}_b \rightarrow \hat{a}_b$$

$$M_q = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}$$

别忘了这是个正交矩阵哦！这样就知道代码怎么来的了吧？好继续。



### 3.向量叉积得出姿态误差：

哎呀，又来棘手问题了，这个我也不太明白怎么讲啊，还是把大神的讲解粘过来吧，大家看看是不是这么回事：

acc是机体坐标参照系上，加速度计测出来的重力向量，也就是实际测出来的重力向量。  
acc是测量得到的重力向量， $V_{gravity}$ 是陀螺积分后的姿态来推算出的重力向量，它们都是机体坐标参照系上的重力向量。  
那它们之间的误差向量，就是陀螺积分后的姿态和加计测出来的姿态之间的误差。  
向量间的误差，可以用向量叉积（也叫向量外积、叉乘）来表示， $V_{error}$ 就是两个重力向量的叉积。  
这个叉积向量仍旧是位于机体坐标系上的，而陀螺积分误差也是在机体坐标系，而且叉积的大小与陀螺积分误差成正比，正好拿来纠正陀螺。（你可以自己拿东西想象一下）由于陀螺是对机体直接积分，所以对陀螺的纠正量会直接体现在对机体坐标系的纠正。

看了上面的话，小弟一直对这个误差向量感到莫名其妙，后来又找到大神的一下一段话：

这里误差没说清楚，不是指向量差。这个叉积误差是指将带有误差的加计向量转动到与重力向量重合，需要绕什么轴，转多少角度。逆向推理一下，这个叉积在机体三轴上的投影，就是加计和重力之间的角度误差。也就是说，如果陀螺按这个叉积误差的轴，转动叉积误差的角度（也就是转动三轴投影的角度）那就能把加计和重力向量的误差消除掉。（具体可看向量叉积的定义）如果完全按叉积误差转过去，那就是完全信任加计。如果一点也不转，那就是完全信任陀螺。那么把这个叉积的三轴乘以 $x\%$ ，加到陀螺的积分角度上去，就是这个 $x\%$ 互补系数的互补算法了。

这个看了好像终于理解点了，再看下代码：

```
//3.向量叉积得出姿态误差  
V_error = acc % V_gravity;
```

这里“%”已经重定义为叉乘的算法了。

### 4.对误差进行积分：

积分求误差，关于当前姿态分离出的重力分量，与当前加速度计测得的重力分量的差值进行积分消除误差

```
V_error_I += V_error * Ki;
```

### 5.互补滤波，姿态误差补偿到角速度上，修正角速度积分漂移

系数不停地被陀螺积分更新，也不停地被误差修正，它和公式所代表的姿态也在不断更新。将积分误差反馈到陀螺仪上，修正陀螺仪的值。将该误差 $V_{error}$ 输入PI控制器后与本次姿态更新周期中陀螺仪测得的角速度相加，最终得到一个修正的角速度值，将其输入四元数微分方程，更新四元数。

```
Gyro += V_error * Kp + V_error_I;
```

Gyro就是得到的修正角速度值，可以用于求解四元数 $q_0 \sim q_3$ 了。

到这里回顾一下八个步骤还漏了一个第七步：

7.四元数归一化：

规范化四元数作用：

1.表征旋转的四元数应该是规范化的四元数，但是由于计算误差等因素，计算过程中四元数会逐渐失去规范化特性，因此必须对四元数做规范化处理。

2.意义在于单位化四元数在空间旋转时是不会拉伸的，仅有旋转角度。这类似与线性代数里面的正交变换。

3.由于误差的引入，使得计算的变换四元数的模不再等于1，变换四元数失去规范性，因此再次更新四元数。

计算欧拉角时候必须要对四元数归一化处理。

```
Q.normalize();
```

呃，关于四元数求解姿态的砖好像搬完了。为什么要用四元数法求解姿态呢？再搬一点关于欧拉角法和旋转矢量法的介绍的。

欧拉角算法通过求解欧拉角微分方程直接计算航向角、俯仰角和横滚角。欧拉角微分方程关系简单明了，概念直观，容易理解，解算过程中无需作正交化处理，但方程中包含有三角运算，这给实时计算带来一定困难。而且当俯仰角接近  $90^\circ$  时方程出现退化现象，这相当于平台惯导中惯性平台的锁定，所以这种方法只适用于水平姿态变化不大的情况，而不适用于全姿态运载体的姿态确定。

方向余弦法对姿态矩阵微分方程作求解，避免了欧拉角法中方程的退化问题，可全姿态工作。但姿态矩阵微分方程实质上是包含九个未知量的线性微分方程组，与四元数法相比，计算量大，实时计算困难，所以工程上并不实用。

四元数法只需求解四个未知量的线性微分方程组，计算量比方向余弦法小，且算法简单，易于操作，是较实用的工程方法。但四元数法实质上是旋转矢量法中的单子样算法，对有限转动引起的不可交换误差的补偿程度不够，所以只适用于低动态运载体（如运输机等）的姿态解算。而对高动态运载体，姿态解算中的算法漂移会十分严重。

旋转矢量法可采用多子样算法实现对不可交换误差做有效补偿，算法关系简单，易于操作，并且通过对系数的优化处理使算法漂移在相同子样算法中达到最小，因此特别适用于角机动频繁激烈或存在严重角振动的运载体的姿态更新。

四元数法和旋转矢量法都通过计算姿态四元数实现姿态更新，但前者直接求解姿态四元数微分方程，而后者通过求解姿态变化四元数再求解姿态四元数，两者的算法思路并不相同。

搬砖搬得好累啊，不过搬得差不多了，感觉挺乱的？呢，主要是由于比较多吧，那我再串一遍？拉倒吧，你看得都累，我写着不累？没闹明白再自己串一遍吧，相信第二遍就能明白了。

哎~对于我这样的渣渣而言也就能理解到这一步了，这也是我好几天的心血整理了一下，也许有和我一样的菜鸟呢，对他们也许能有点帮助，做得不好希望大神们能耐心给与指正，而不是嗤之以鼻，或者喷我一顿就走。。。毕竟整理了两天呢（我还以为一中午就能搞定呢）。渣渣的学习之路也是挺不容易的，因为基础渣渣，学校渣渣，所以难以得到有效地帮助和指导，有时在群里寻求帮助，无聊的群友会告诉你看书去，呵呵。。。我也知道看书啊。。。哪怕你能告诉我我的问题在那本书的那部分能有相似吧？一句看书去，上网查啊，等于没回答，如果一直这样自己看下去可能半年也解决不了，因为渣渣的学习环境是有局限性的。

不过好在有更多很热心的群友能提供给我帮助，把他们收集的好贴发给我，或者干脆手写一个公式推导，一个电路图，然后拍照发给我，还有的帮我下载照片，分类命名给我，艾玛！热泪盈眶啊有木有！！！再次感谢这些热心帮助我的小伙伴 @ 奇点，@ 杜掌柜，@ 廉价物品，@ 忘记名字的小伙伴 .....

下面附上被我搬砖的几个好贴，谢谢大神们的乐于分享：

对四元数解算姿态的理解（基于匿名六轴），感谢社区：<http://www.playuav.com/article/79>

软件姿态解算：<http://www.crazepony.com/wiki/software-algorithm.html>

捷联惯导算法心得：  
<http://www.amobbs.com/forum.php?mod=viewthread&tid=5492189&highlight=>

附件：匿名姿态解算代码

《惯性导航》秦永元

《推导\_四元数.pdf》