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

$$\boldsymbol{q}(t+T) = \boldsymbol{q}(t) + T\boldsymbol{\Omega}_b(t)\boldsymbol{q}(t)$$

式中的 $\boldsymbol{\Omega}_b(t)$ 如式(7.2.19)所示。

将式(7.3.25)展开成元素的表达式，有

$$\lambda(t+T) = \lambda(t) + \frac{T}{2}[-\omega_X(t)P_1(t) - \omega_Y(t)P_2(t) - \omega_Z(t)P_3(t)]$$

$$P_1(t+T) = P_1(t) + \frac{T}{2}[\omega_X(t)\lambda(t) + \omega_Z(t)P_2(t) - \omega_Y(t)P_3(t)]$$

$$P_2(t+T) = P_2(t) + \frac{T}{2}[\omega_Y(t)\lambda(t) - \omega_Z(t)P_1(t) + \omega_X(t)P_3(t)]$$

$$P_3(t+T) = P_3(t) + \frac{T}{2}[\omega_Z(t)\lambda(t) + \omega_Y(t)P_1(t) -$$

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《推導_四元數.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

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

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

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

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 Δt 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w_x、w_y、w_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 θg 与陀螺仪积分后的角度θ 进行比较,将比较的误差信号经过比例 Tg 放大之后与陀螺仪输出的角速度信号叠加之后再进行积分。对于加速度计给定的角度θg ，经过比例、积分环节之后产生的角度θ必然最终等于θg 。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θ中的积累误差消除了。加速度计所产生的角度信息θg 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θ 的影响，因此比例系数 Tg 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 Tg 多么小，最终输出角度θ必然与加速度计测量的角度θg 相等，只是这个调节过程会随着 Tg 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， 因为是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

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

做歸一化

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

設旋轉至飛行器上的重力加速度為 $\hat{g}_b$

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

$$\Rightarrow \hat{g}_b = \begin{pmatrix} g_{bx} \\ g_{by} \\ g_{bz} \end{pmatrix} = \begin{pmatrix} M_{13} \\ M_{23} \\ M_{33} \end{pmatrix}$$

飛行器上的加速度計測得的加速度 $\vec{a}_b$ 做歸一化

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

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

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：

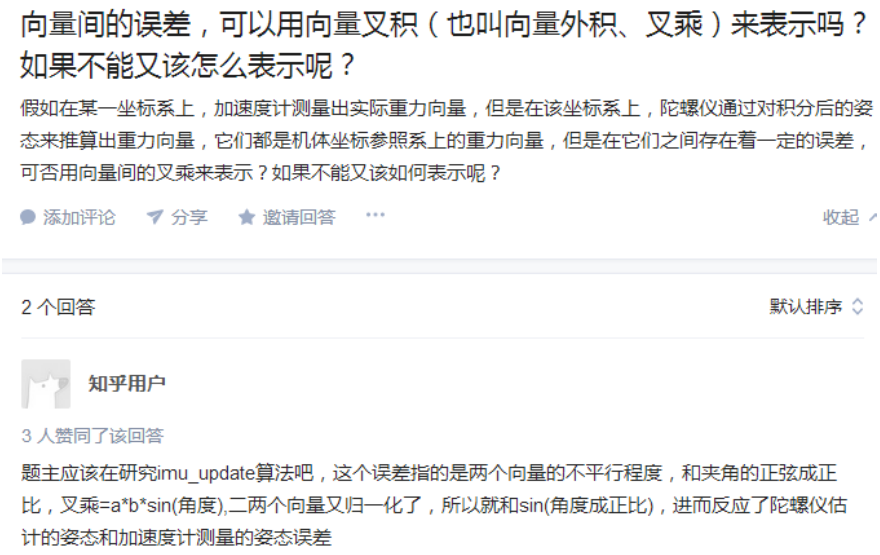
acc 是机体坐标参照系上，加速度计测出来的重力向量，也就是实际测出来的重力向量。

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。



摘自：http://www.playuav.com/article/79

向量叉积得出姿态误差，通过加速度计测得的重力坐标系下的单位向量与上一次四元数转换成的单位向量进行叉乘，以此得到其误差量外积，在相减得到差分就是误差，把叉积等同于角度误差

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 q0~q3 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°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？呃，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

《推導_四元數.pdf》

——潇洒的石头