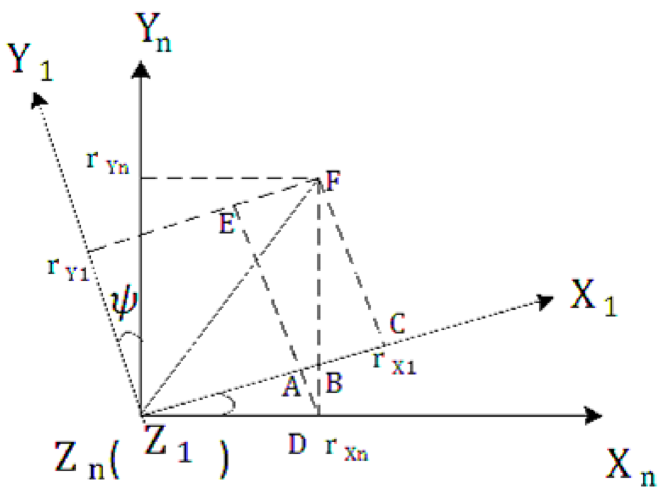
姿态解算原理

目前，描述两个坐标系之间关系的常用方法主要有欧拉角法、方向余弦矩阵法和四元数法。因此要弄懂这三种方法的定义及关系，我们必须先从坐标系转化开始了解。下面以四旋翼为例，定义两个坐标系。导航坐标系(参考坐标系)n，选取东北天右手直角坐标系作为导航坐标系 n。载体坐标系(机体坐标系)b，选取右手直角坐标系定义: 四轴向右为 X 正方向，向前为 Y 轴正方向，向上为 Z 轴正方向。

欧拉角：欧拉角是一种常用的描述方位的方法，是由欧拉提出的。基本思想就是将两个坐标系的变换分解为绕三个不同的坐标轴的三次连续转动组成的序列。欧拉角的旋转规定为连续两次旋转，必须绕着不同的转动轴旋转，所以一共有 12 种旋转顺规。这里我们选用 Z-Y-X 的旋转顺规来描述 b 系与 n 系的关系。Z-Y-X 顺规就是指。绕 Z 轴旋转偏航角(yaw)ψ（习惯以北偏东为正)，绕 Y 轴旋转横滚角(roll)Φ，绕 X 轴旋转俯仰角(pitch)θ。

先由简单的 2 维 XY 坐标系计算，坐标原点为 O（图中未标出）载体坐标系 b 和导航坐标系 n 初始重合，导航坐标系 n 绕 Z 轴旋转ψ角得坐标系 1，坐标系 1 绕 Y 轴旋转Φ角得坐标系 2，坐标系 2 绕 X 轴旋转θ角得载体坐标系 b。向量 OF 在导航坐标系中的坐标为(r_{Xn}, r_{Yn})，在载体坐标系中的坐标为(r_{Xb}, r_{Yb})。经过三次旋转之后，可以得出这两个坐标的关系，也就是从 n 系到 b 系的旋转矩阵R_n^b，这种方法被称为由方向余弦表示的旋转矩阵。下面分别来讲解。

坐标系 n 绕 Z 轴旋转ψ角（yaw 角）得坐标系 1:



$\mathrm{r_{X1}} = \mathrm{OA} + \mathrm{AC} = \mathrm{OD}\cos\psi + \mathrm{DF}\sin\psi = \mathrm{r_{Xn}}\cos\psi + \mathrm{r_{Yn}}\sin\psi$

$\mathrm{r_{Y1}} = \mathrm{DE} - \mathrm{AD} = \mathrm{DF}\cos\psi - \mathrm{OD}\sin\psi = -\mathrm{r_{Xn}}\sin\psi + \mathrm{r_{Yn}}\cos\psi$

因为 Z 轴坐标未变，所以$r_{Z1} = r_{Zn}$

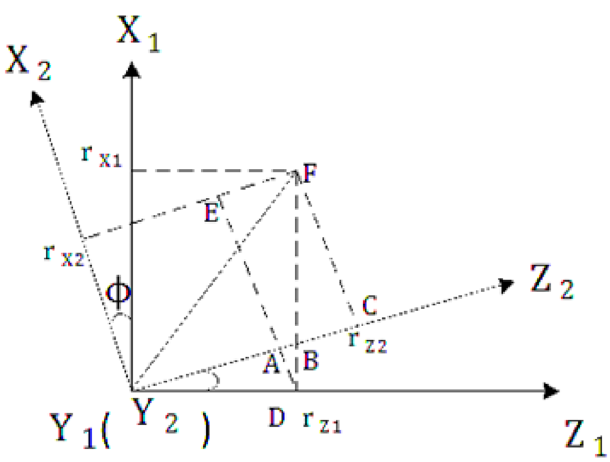
转换成矩阵形式表示为：

$$\begin{bmatrix} \mathrm{r_{X1}} \\ \mathrm{r_{Y1}} \\ r_{Z1} \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{r_{Xn}} \\ \mathrm{r_{Yn}} \\ r_{Zn} \end{bmatrix}$$

从 n 系到 1 系的旋转矩阵$R_n^1 = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$

从 1 系到 n 系的旋转矩阵$R_1^n = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$

坐标系 1 绕 Y 轴旋转Φ角（roll 角）得坐标系 2：



$$\mathrm{r_{Z2}} = \mathrm{OA} + \mathrm{AC} = \mathrm{OD}\cos\emptyset + \mathrm{DF}\sin\emptyset = \mathrm{r_{Z1}}\cos\emptyset + \mathrm{r_{X1}}\sin\emptyset$$

$$\mathrm{r_{X2}} = \mathrm{DE} - \mathrm{AD} = \mathrm{DF}\cos\emptyset - \mathrm{OD}\sin\emptyset = -\mathrm{r_{Z1}}\sin\emptyset + \mathrm{r_{X1}}\cos\emptyset$$

因为 Y 轴坐标未变，所以$r_{Y2} = r_{Y1}$

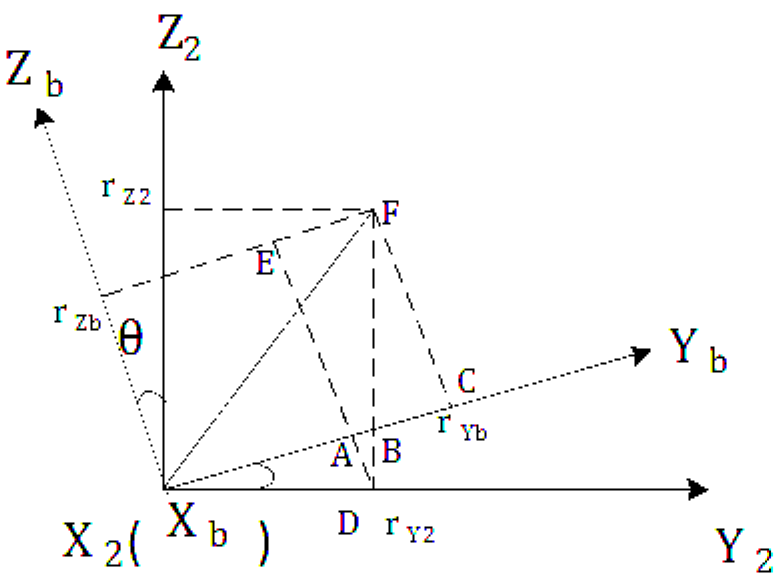
转换成矩阵形式表示为：

$$\begin{bmatrix} \mathrm{r_{X2}} \\ \mathrm{r_{Y2}} \\ r_{Z2} \end{bmatrix} = \begin{bmatrix} \cos\emptyset & 0 & -\sin\emptyset \\ 0 & 1 & 0 \\ \sin\emptyset & 0 & \cos\emptyset \end{bmatrix} \begin{bmatrix} \mathrm{r_{X1}} \\ \mathrm{r_{Y1}} \\ r_{Z1} \end{bmatrix}$$

从 1 系到 2 系的旋转矩阵$R_1^2 = \begin{bmatrix} \cos\emptyset & 0 & -\sin\emptyset \\ 0 & 1 & 0 \\ \sin\emptyset & 0 & \cos\emptyset \end{bmatrix}$

从 2 系到 1 系的旋转矩阵$R_2^1 = \begin{bmatrix} \cos\emptyset & 0 & \sin\emptyset \\ 0 & 1 & 0 \\ -\sin\emptyset & 0 & \cos\emptyset \end{bmatrix}$

坐标系 2 绕 X 轴旋转θ角（pitch 角）得坐标系 b：



$r_{Yb} = OA + AC = OD\cos\theta + DF\sin\theta = r_{Y2}\cos\theta + r_{Z2}\sin\theta$

$r_{Zb} = DE - AD = DF\cos\theta - OD\sin\theta = -r_{Y2}\sin\theta + r_{Z2}\cos\theta$

因为X轴坐标未变，所以$r_{Xb} = r_{X2}$

转换成矩阵形式表示为：

$$\begin{bmatrix} r_{Xb} \\ r_{Yb} \\ r_{Zb} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} r_{X2} \\ r_{Y2} \\ r_{Z2} \end{bmatrix}$$

从2系到b系的旋转矩阵$R_2^b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix}$

从b系到2系的旋转矩阵$R_b^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$

所以，可以得到从n系到b系的矩阵表示形式为：

$$\begin{bmatrix} r_{Xb} \\ r_{Yb} \\ r_{Zb} \end{bmatrix} = R_2^b R_1^2 R_n^1 \begin{bmatrix} r_{Xn} \\ r_{Yn} \\ r_{Zn} \end{bmatrix}$$

$$\begin{bmatrix} r_{Xb} \\ r_{Yb} \\ r_{Zb} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\emptyset & 0 & -\sin\emptyset \\ 0 & 1 & 0 \\ \sin\emptyset & 0 & \cos\emptyset \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{Xn} \\ r_{Yn} \\ r_{Zn} \end{bmatrix}$$

为方便，将cos和sin分别用C和S表示，则不同角度下的表示为C_θ、$C_\emptyset$、C_ψ、S_θ、$S_\emptyset$、S_ψ。

$$\begin{bmatrix} r_{Xb} \\ r_{Yb} \\ r_{Zb} \end{bmatrix} = \begin{bmatrix} C_\psi C_\emptyset & C_\emptyset S_\psi & -S_\emptyset \\ S_\theta S_\emptyset C_\psi - S_\psi C_\theta & S_\theta S_\emptyset S_\psi + C_\theta C_\psi & C_\emptyset S_\theta \\ S_\psi S_\theta + C_\theta C_\psi S_\emptyset & C_\theta S_\emptyset S_\psi - S_\theta C_\psi & C_\theta C_\emptyset \end{bmatrix} \begin{bmatrix} r_{Xn} \\ r_{Yn} \\ r_{Zn} \end{bmatrix}$$

从n系到b系的旋转矩阵$R_n^b = \begin{bmatrix} C_\psi C_\emptyset & C_\emptyset S_\psi & -S_\emptyset \\ S_\theta S_\emptyset C_\psi - S_\psi C_\theta & S_\theta S_\emptyset S_\psi + C_\theta C_\psi & C_\emptyset S_\theta \\ S_\psi S_\theta + C_\theta C_\psi S_\emptyset & C_\theta S_\emptyset S_\psi - S_\theta C_\psi & C_\theta C_\emptyset \end{bmatrix}$

从 b 系到 n 系的旋转矩阵$R_b^n = \begin{bmatrix} C_\psi C_\emptyset & S_\theta S_\emptyset C_\psi - S_\psi C_\theta & S_\psi S_\theta + C_\theta C_\psi S_\emptyset \\ C_\emptyset S_\psi & S_\theta S_\emptyset S_\psi + C_\theta C_\psi & C_\theta S_\emptyset S_\psi - S_\theta C_\psi \\ -S_\emptyset & C_\emptyset S_\theta & C_\theta C_\emptyset \end{bmatrix}$

所以有$(R_n^b)^T = R_b^n$

到这里我们已经得到了一个表示旋转的方向余弦矩阵。不过要想用欧拉角解算姿态，我们套用欧拉角微分方程就行了：

(欧拉角微分方程推导 http://blog.sina.com.cn/s/blog_40edfdc90102wazm.html)

$$\begin{bmatrix} \dot{\theta} \\ \dot{\gamma} \\ \dot{\psi} \end{bmatrix} = \frac{1}{C_\theta} \begin{bmatrix} C_\gamma C_\theta & 0 & C_\theta S_\gamma \\ S_\gamma S_\theta & 1 & -S_\theta C_\gamma \\ -S_\gamma & 0 & C_\gamma \end{bmatrix} \begin{bmatrix} \omega_{nbx}^b \\ \omega_{nby}^b \\ \omega_{nbz}^b \end{bmatrix}$$

上式中左侧，是本次更新后的欧拉角的角速度，对应 row、pit、yaw。右侧，是上个周期测算出来的角速度。三个角速度由直接安装在四轴飞行器的三轴陀螺仪在这个周期转动的角速度（单位为弧度每秒）获得，并通过欧拉微分方程得到更新后的角速度值，最后乘以间隔时间 T 得到陀螺仪角度，比如 0.02 秒 0.01 弧度/秒=0.0002 弧度。因此求解这个微分方程就能解算出当前的欧拉角。

欧拉角的缺点，一方面是因为欧拉角微分方程中包含了大量的三角运算，这给实时解算带来了一定的困难。而且当俯仰角为 90 度时方程式会出现神奇的“GimbalLock（万向节死锁）”。所以欧拉角方法只适用于水平姿态变化不大的情况，而不适用于全姿态飞行器的姿态确定。所以要引入四元数法。

关于万向节死锁：

http://www.cnblogs.com/soroman/archive/2008/03/24/1118996.html

四元数：

四元数定义：四元数是由四个元构成的数：

$$Q(q_0,q_1,q_2,q_3)=q_0+q_1i+q_2j+q_3k$$

其中，q_0是实数，i,j,k既是互相正交的单位向量，又是虚单位$\sqrt{-1}$。四元数有四个分量，其中只有一个是普通数值。因此可以将四元数看成由两部分组成：数量部分和向量部分。一个四元数可以表示成一个标量（数值）和一个向量的和。四元数的范数（模长的平方）也正好等于数量分量的平方与向量部分模长平方的和。

四元数的大小——范数：

四元数的大小用四元数的范数来表示：

$$\|Q\|=q_0^2+q_1^2+q_2^2+q_3^2$$

若$\|Q\|=1$，则称 Q 为规范化四元数。

四元数乘法：

$$\mathrm{i}\otimes\mathrm{i}=-1,\qquad \mathrm{j}\otimes\mathrm{j}=-1,\qquad \mathrm{k}\otimes\mathrm{k}=-1$$

$$\mathrm{i}\otimes\mathrm{j}=\mathrm{k},\qquad \mathrm{j}\otimes\mathrm{k}=\mathrm{i},\qquad \mathrm{k}\otimes\mathrm{i}=\mathrm{j}$$

$$\mathrm{j}\otimes\mathrm{i}=-\mathrm{k},\qquad \mathrm{k}\otimes\mathrm{j}=-\mathrm{i},\qquad \mathrm{i}\otimes\mathrm{k}=-\mathrm{j}$$

$\otimes$ 表示四元数乘法。

上述关系可以叙述为：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。所以四元数可以看成四维空间中的一个向量，也可以看成是超复数。

两个四元数乘法运算：

$Q(q_0,q_1,q_2,q_3)\otimes P(p_0,p_1,p_2,p_3) = (q_0+q_1i+q_2j+q_3k)\otimes(p_0+p_1i+p_2j+$

$p_3k) = (q_0p_0 - q_1p_1 - q_2p_2 - q_3p_3) + (q_0p_1 + q_1p_0 + q_2p_3 - q_3p_2)i + (q_0p_2 + q_2p_0 + q_3p_1 - q_1p_3)j + (q_0p_3 + q_3p_0 + q_1p_2 - q_2p_1)\mathrm{k} = r_0 + r_1 i + r_2 j + r_3 k$

上式写成矩阵形式:

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

或者

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

其中，$\mathrm{Matrix}(Q)$的构成形式是：第一列是四元数 Q 本身，第一行是 Q 的共轭四元数Q^*的转置，划去第一行和第一列余下的部分为:

$$\mathrm{V}(Q) = \begin{bmatrix} q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix}$$

称为$\mathrm{Matrix}(Q)$的核，是由四元数 Q 的元构成的反对称矩阵。$Matrix'(P)$的核为:

$$V'(P) = \begin{bmatrix} p_0 & -p_3 & p_2 \\ p_3 & p_0 & -p_1 \\ -p_2 & p_1 & p_0 \end{bmatrix}$$

四元数除法（求逆）:

如果 P⊗R=1，则称 R 为 P 的逆，$\mathrm{R} = P^{-1}$。

$$P^{-1} = P^*/\|P\|$$

当 P 的 2-范数为 1 时，称 P 为单位四元数，有 P 的逆等于 P的共轭（数量部分相等向量部分相反）。即:

$$P^{-1} = P^*/\|P\| = P^* = \bar{P}$$

由四元数表示的旋转矩阵：

定义四元数q_b^n用来表示导航坐标系 n 到载体坐标系 b 的旋转变换，那么有：

$$r^n = q_b^n \otimes r^b \otimes \left(q_b^n\right)^{-1} = q_b^n \otimes r^b \otimes \left(q_b^n\right)^* = q_b^n \otimes (r^b \otimes \left(q_b^n\right)^*) = q_b^n \otimes (M(q_b^{n*})r^b)$$
$$= M(q_b^n)M(q_b^{n*})r^b$$

$$M\left(q_b^n\right)M\left(q_b^{n*}\right) = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \\ -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ r_x^n \\ r_y^n \\ r_z^n \end{bmatrix}$$
$$= \begin{bmatrix} q_0^2+q_1^2+q_2^2+q_3^2 & 0 & 0 & 0 \\ 0 & q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_0q_2+q_1q_3) \\ 0 & 2(q_0q_3+q_1q_2) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 0 & 2(q_1q_3-q_0q_2) & 2(q_0q_1+q_2q_3) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \begin{bmatrix} 0 \\ r_x^b \\ r_y^b \\ r_z^b \end{bmatrix}$$

所以，从 b 系到 n 系的旋转矩阵为：

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

又由方向余弦表示的旋转矩阵为：

$$R_b^n = \begin{bmatrix} C_\psi C_\emptyset & S_\theta S_\emptyset C_\psi - S_\psi C_\theta & S_\psi S_\theta + C_\theta C_\psi S_\emptyset \\ C_\emptyset S_\psi & S_\theta S_\emptyset S_\psi + C_\theta C_\psi & C_\theta S_\emptyset S_\psi - S_\theta C_\psi \\ -S_\emptyset & C_\emptyset S_\theta & C_\theta C_\emptyset \end{bmatrix}$$

由R_b^n的第一列和第三行对应相等可以列出五方程，如下：

$$\begin{cases} C_\psi C_\emptyset = q_0^2 + q_1^2 - q_2^2 - q_3^2 \\ C_\emptyset S_\psi = 2(q_0 q_3 + q_1 q_2) \\ -S_\emptyset = 2(q_1 q_3 - q_0 q_2) \\ C_\emptyset S_\theta = 2(q_0 q_1 + q_2 q_3) \\ C_\theta C_\emptyset = q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{cases}$$

解得姿态欧拉角如下：

$$\psi = \tan^{-1}[\left(2(q_0 q_3 + q_1 q_2)\right)/(q_0^2 + q_1^2 - q_2^2 - q_3^2)]$$

$$\emptyset = -\sin^{-1}(2(q_1 q_3 - q_0 q_2))$$

$$\theta = \tan^{-1}[\,2(q_0 q_1 + q_2 q_3)/q_0^2 - q_1^2 - q_2^2 + q_3^2]$$

$\tan^{-1}$和$\sin^{-1}$的结果是[-π/2,π/2]，这并不能覆盖所有朝向，因此需要用$\tan^{-1}2$来代替$\tan^{-1}$，从而产生所有可能的朝向，得到新的公式如下面所示：

$$\psi = \tan^{-1} 2[\left(2(q_0 q_3 + q_1 q_2)\right)/(q_0^2 + q_1^2 - q_2^2 - q_3^2)]$$

$$\emptyset = -\sin^{-1}(2(q_1 q_3 - q_0 q_2))$$

$$\theta = \tan^{-1} 2[\,2(q_0 q_1 + q_2 q_3)/q_0^2 - q_1^2 - q_2^2 + q_3^2]$$

一般 pitch 角为±90°，roll 和 yaw 角是±180°。

四元数当中的四个元q_0, q_1, q_2, q_3与一个单位向量$\omega = \omega_x i + \omega_y j + \omega_z k$（其中$\|\omega\| = 1$，$\omega_x \omega_y \omega_z$为投影到三维轴上的方向余弦）绕任意轴旋转α角的对应关系是怎样的？

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

关系：

$$q_0 = \cos\frac{\alpha}{2}$$

$$q_1 = \omega_x \sin\frac{\alpha}{2}$$
$$q_2 = \omega_y \sin\frac{\alpha}{2}$$
$$q_3 = \omega_z \sin\frac{\alpha}{2}$$

推导过程：具体请看秦永元《惯性导航》p292-297。单位向量ω绕任意轴旋转α角得到的前后两个坐标系的变换矩阵 R（罗德里格斯变换矩阵）：

$$R =$$

$$\begin{bmatrix} \cos\alpha + \omega_x^2(1-\cos\alpha) & \omega_y\omega_z(1-\cos\alpha) - \omega_z\sin\alpha & \omega_y\sin\alpha + \omega_x\omega_z(1-\cos\alpha) \\ \omega_z\sin\alpha + \omega_x\omega_y(1-\cos\alpha) & \cos\alpha + \omega_y^2(1-\cos\alpha) & \omega_y\omega_z(1-\cos\alpha) - \omega_x\sin\alpha \\ \omega_x\omega_z(1-\cos\alpha) - \omega_y\sin\alpha & \omega_x\sin\alpha + \omega_y\omega_z(1-\cos\alpha) & \cos\alpha + \omega_z^2(1-\cos\alpha) \end{bmatrix}$$

旋转矩阵 R 与四元数矩阵对应，斜对角线相加之和为：

$$4q_0^2 = 2 + 2\cos\alpha$$

$$q_0 = \sqrt{(1+\cos\alpha)/2} = \cos\frac{\alpha}{2}$$

由四元数变换矩阵第二行第一个与第一行第二个元素相减，得：

$$4q_0q_3 = 2\omega_z\sin\alpha$$
$$q_3 = \omega_z \sin\frac{\alpha}{2}$$

同理得

$$q_1 = \omega_x \sin\frac{\alpha}{2}$$
$$q_2 = \omega_y \sin\frac{\alpha}{2}$$

四元数$Q = \cos\frac{\alpha}{2} + \omega\sin\frac{\alpha}{2}$，ω分成各个方向上是它的方向余弦$\sin\beta_x$、$\sin\beta_y$、$\sin\beta_z$。

在旋转中，描述刚体旋转的四元数是规范化四元数，其范数为 1。

基于四元数互补滤波的姿态解算

```
#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 IMUupdate(T_int16_xyz *gyr, T_int16_xyz *acc, T_float_angle *angle)
{
     float ax = acc->X,ay = acc->Y,az = acc->Z;
     float gx = gyr->X,gy = gyr->Y,gz = gyr->Z;
   float norm;//归一化
   float vx, vy, vz;
   float ex, ey, ez;

   float q0q0 = q0*q0;
   float q0q1 = q0*q1;
   float q0q2 = q0*q2;
   float q1q1 = q1*q1;
   float q1q3 = q1*q3;
   float q2q2 = q2*q2;
   float q2q3 = q2*q3;
   float q3q3 = q3*q3;

     if(ax*ay*az==0)
```

基于四元数互补滤波的姿态解算

```
#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 IMUupdate(T_int16_xyz *gyr, T_int16_xyz *acc, T_float_angle *angle)

{

    float ax = acc->X,ay = acc->Y,az = acc->Z;

    float gx = gyr->X,gy = gyr->Y,gz = gyr->Z;

  float norm;//归一化

  float vx, vy, vz;

  float ex, ey, ez;


  float q0q0 = q0*q0;

  float q0q1 = q0*q1;

  float q0q2 = q0*q2;

  float q1q1 = q1*q1;

  float q1q3 = q1*q3;

  float q2q2 = q2*q2;

  float q2q3 = q2*q3;

  float q3q3 = q3*q3;


    if(ax*ay*az==0)
```

```
        return;

    gx *= Gyro_Gr;
    gy *= Gyro_Gr;
    gz *= Gyro_Gr;
//1.重力加速度归一化
   norm = sqrt(ax*ax + ay*ay + az*az);
   ax = ax /norm;//取 ax、ay、az 的模
   ay = ay / norm;
   az = az / norm;
//2.提取四元数的等效余弦矩阵中的重力分量
   vx = 2*(q1q3 - q0q2);
   vy = 2*(q0q1 + q2q3);
   vz = q0q0 - q1q1 - q2q2 + q3q3 ;
//3.向量叉积得出姿态误差
   ex = (ay*vz - az*vy) ;
   ey = (az*vx - ax*vz) ;
   ez = (ax*vy - ay*vx) ;
//4.对误差进行积分
   exInt = exInt + ex * Ki;
   eyInt = eyInt + ey * Ki;
```

```
  ezInt = ezInt + ez * 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
  gx = gx + Kp*ex + exInt;
  gy = gy + Kp*ey + eyInt;
  gz = gz + Kp*ez + ezInt;
// 6.一阶龙格库塔求解四元数微分方程
  q0 = q0 + (-q1*gx - q2*gy - q3*gz)*halfT;
  q1 = q1 + (q0*gx + q2*gz - q3*gy)*halfT;
  q2 = q2 + (q0*gy - q1*gz + q3*gx)*halfT;
  q3 = q3 + (q0*gz + q1*gy - q2*gx)*halfT;
//7.四元数归一化:
  norm = sqrt(q0*q0 + q1*q1 + q2*q2 + q3*q3);
  q0 = q0 / norm;
  q1 = q1 / norm;
  q2 = q2 / norm;
  q3 = q3 / norm;

angle->yaw   = atan2(2 * q1 * q2 + 2 * q0 * q3, -2 * q2*q2 - 2 * q3* q3 + 1)* RtA;
// yaw
  angle->pit  = asin(-2 * q1 * q3 + 2 * q0* q2)* RtA ; //pitch
  angle->rol  = atan2(2 * q2 * q3 + 2 * q0 * q1, -2 * q1 * q1 - 2 * q2* q2 + 1)* RtA;
```

```
//roll

    if(angle->rol>90||angle->rol<-90)
    {
        if(angle->pit>0)angle->pit=180-angle->pit;
        if(angle->pit<0)angle->pit=-(180+angle->pit);
    }
}
```

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？可以通过加速度计的输出稳定性来修正陀螺仪的积分漂移误差。利用导航坐标系 n 当中的标准重力加速度 g，把它转换到载体坐标系 b 上，$\mathbf{g}^{\mathbf{b}}$与载体坐标系加速度计输出$\mathbf{a}^{\mathbf{b}}$进行比较，将比较的误差信号经过积分和比例$\boldsymbol{k_p}$放大后与陀螺仪输出的角速度信号叠加，修正角速度值。对于加速度计给定的值，不会存在积累误差，所以最终将积累误差消除了。加速度计所产生的

信息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$\boldsymbol{k_p}$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$\boldsymbol{k_p}$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$\boldsymbol{k_p}$的减小而延长。

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：

取导航坐标系中的标准重力加速度g，定义为$g^n = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$，那么将$g^n$转换至载体坐标系b中的表达式为：

$$g^b = R_n^b g^n = \begin{bmatrix} 2(q_1 q_3 - q_0 q_2) \\ 2(q_0 q_1 + q_2 q_3) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

载体坐标系b系中加速度计的输出为a，由于重力加速度为标准重力加速度，所以这里要对加速度计输出归一化后才能继续运算，因此归一化后的载体坐标系加速度计输出为：

$$a^b = \begin{bmatrix} a_x/\|a\| \\ a_y/\|a\| \\ a_z/\|a\| \end{bmatrix}$$

3.向量叉积得出姿态误差:

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，Vx、Vy、Vz 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，ex、ey、ez 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。

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。

再搞个直白一点的例子：机体好似一条船，地理就是那地图，姿态就是航向（船头在地图上的方位），重力和地磁是地图上的灯塔，陀螺/积分公式是舵手，加计和电子罗盘是瞭望手。舵手负责估计和把稳航向，他相信自己，本来船向北

开的，就一定会一直往北开，觉得转了 90 度弯，那就会往东开。当然如果舵手很牛逼，也许能估计很准确，维持很长时间。不过只信任舵手，肯定会迷路，所以一般都有地图和瞭望手来观察误差。瞭望手根据地图灯塔方位和船的当前航向，算出灯塔理论上应该在船的 X 方位。然而看到实际灯塔在船的 Y 方位，那肯定船的当前航向有偏差了，偏差就是 ERR=X-Y。舵手收到瞭望手给的 ERR 报告，觉得可靠，那就听个 90%*ERR，觉得天气不好、地图误差大，那就听个 10%*ERR，根据这个来纠正估算航向。

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。

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 ex、ey、ez 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。

6.一阶龙格库塔法求解四元数微分方程

一阶龙格库塔计算式

$$\mathrm{q}(\mathrm{t}+\mathrm{T}) = \mathrm{q}(\mathrm{t}) + T\Omega_b(t)\mathrm{q}(\mathrm{t})$$

在四元数中，$\mathrm{Q}(q_0, q_1, q_2, q_3)$将被更新如下所示:

$$(q_0)_{t+\mathrm{T}} = (q_0)_t + \frac{T}{2}[-\omega_x q_1 - \omega_y q_2 - \omega_z q_3]$$

$$(q_1)_{t+T} = (q_1)_t + \frac{T}{2}[\omega_x q_0 - \omega_y q_3 + \omega_z q_2]$$

$$(q_2)_{t+T} = (q_2)_t + \frac{T}{2}[\omega_x q_3 + \omega_y q_0 - \omega_z q_1]$$

$$(q_3)_{t+T} = (q_3)_t + \frac{T}{2}[-\omega_x q_2 + \omega_y q_1 + \omega_z q_0]$$

四元数微分

已知四元数$\mathrm{Q} = \cos\frac{\alpha}{2} + \mathrm{n}\sin\frac{\alpha}{2}$，对时间微分

$$\frac{dQ}{dt} = -\frac{1}{2}\sin\frac{\alpha}{2}\frac{d\alpha}{dt} + \frac{dn}{dt}\sin\frac{\alpha}{2} + n\frac{1}{2}\cos\frac{\alpha}{2}\frac{d\alpha}{dt}$$

已知刚体绕瞬时转轴转过α角，角速度为$\frac{d\alpha}{dt} = \omega^E$，$\mathrm{n}\cdot\mathrm{n} = -1$，$\frac{dn}{dt} = 0$。E为导航坐标系，b为载体坐标系。

$$\frac{dQ}{dt} = n\frac{1}{2}\omega^E\left(\cos\frac{\alpha}{2} + \mathrm{n}\sin\frac{\alpha}{2}\right) = \frac{1}{2}\omega^E Q$$

因为陀螺仪在载体坐标系上测得的角速度为ω^b，$\omega^b = \omega_x i + \omega_y j + \omega_z k$（$\omega_x$、$\omega_y$、$\omega_z$是陀螺仪实际所测得的角速度信息），所以需要将$\omega^E$转到$\omega^b$上去。

$$\omega^b = Q^*\omega^E Q$$

两边同乘以Q

$$Q\omega^b = QQ^*\omega^E Q = \omega^E Q$$

$$\frac{dQ}{dt} = n\frac{1}{2}\omega^E\left(\cos\frac{\alpha}{2} + \mathrm{n}\sin\frac{\alpha}{2}\right) = \frac{1}{2}\omega^E Q = \frac{1}{2}Q\omega^b$$

将$\frac{dQ}{dt} = \frac{1}{2}Q\omega^b$展开：

$$\frac{dQ}{dt}=\frac{1}{2}(q_0+q_1i+q_2j+q_3k)(\omega_xi+\omega_yj+\omega_zk)$$

$$=\frac{1}{2}\begin{pmatrix}0 & -\omega_x & -\omega_y & -\omega_z\\ \omega_x & 0 & \omega_z & -\omega_y\\ \omega_y & -\omega_z & 0 & \omega_x\\ \omega_z & \omega_y & -\omega_x & 0\end{pmatrix}\begin{pmatrix}q_0\\ q_1\\ q_2\\ q_3\end{pmatrix}$$

整理为:

$$\frac{dQ}{dt}=\Omega_bQ$$

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

更新四元数过程：假设有一阶微分方程

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

解为

$$X[t+\Delta\mathrm{t}]=X[t]+\Delta\mathrm{t}\,f[X[t],\omega[t]]$$

Δt为取样周期，进而，四元数为:

$$Q[t+\Delta\mathrm{t}]=Q[t]+\Delta\mathrm{t}\,\Omega_b(t)\mathrm{q(t)}$$

展开上式

$$(q_0)_{t+\Delta\mathrm{t}}=(q_0)_{\Delta\mathrm{t}}+\frac{T}{2}[-\omega_xq_1-\omega_yq_2-\omega_zq_3]$$

$$(q_1)_{t+\Delta\mathrm{t}}=(q_1)_{\Delta\mathrm{t}}+\frac{T}{2}[\omega_xq_0-\omega_yq_3+\omega_zq_2]$$

$$(q_2)_{t+\Delta\mathrm{t}}=(q_2)_{\Delta\mathrm{t}}+\frac{T}{2}[\omega_xq_3+\omega_yq_0-\omega_zq_1]$$

$$(q_3)_{t+\Delta t} = (q_3)_{\Delta t} + \frac{T}{2}[-\omega_x q_2 + \omega_y q_1 + \omega_z q_0]$$

所以知道角速度后就可以更新四元数。

程序当中的 half-T 设置，它的设置与你配置的寄存器有关，比如设置采样率 i2cWrite(devAddr,MPU6050_RA_SMPLRT_DIV,0x00); 和低通滤波 i2cWrite(devAddr,MPU6050_RA_CONFIG,0x03); 则依据采样率公式采样率 = 陀螺仪的输出率 / (1 + SMPLRT_DIV)。当数字低通滤波器没有使能的时候，陀螺仪的输出平路等于 8KHZ，反之等于 1KHZ。现在我们启用了低通滤波，则输出率为 1KHz，而分频系数为 0x00；所以得到的采样率为 1KHz/(1+0)=1Khz，则采样周期为 0.001 秒，则 half-T=0.0005。

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。

未完待续

http://wenku.baidu.com/linkurl=m5PL1Em4sFdsEcjsaUDwJrXLAY8705TMpIXYPP6dDCo0FcJBIdI3QJLNlvfPg25wiKK94Pq7lwO5BzVQPCHTb3sKi5SLC9PmkHjA4PQSCoW 四元数法及其应用

http://blog.csdn.net/wkdwl/article/details/52119163 互补滤波和梯度下降算法

http://blog.csdn.net/wkdwl/article/details/52106646 四元数，欧拉角，方向余弦定义

http://blog.csdn.net/sunnyxiaohu/article/details/50596671 有卡尔曼滤波的代码

http://blog.csdn.net/q123456789098/article/details/53420797 acc+gpro 融合地磁算法
http://blog.csdn.net/wubaobao1993/article/details/53130362 基于四元数的简单互补滤波姿态解算
推荐秦永元的两本书，一是《惯性导航》，目前已出到第二版了;二是《卡尔曼滤波与组合导航原理》。
http://wenku.baidu.com/link?url=dvAgzlkrSVzkrUBBksLzQHNuatEcz1Zaxa_TQp-qHVJ4yHMY_3-pwSilx6vjYqXwGruIt3jkR0inq7fZvxZ8VZs4scq7lDLhsXIBb-VJe2C 四元数解算姿态完全解析及资料汇总
捷联惯导算法心得: http://www.amobbs.com/forum.php?mod=viewthread&tid=5492189&highlight=
http://www.cnblogs.com/soroman/archive/2008/03/24/1118996.html 关于万向节死锁
《推導_四元數.pdf》