

姿态解算

知识库

名称解释

姿态解算

目标机体通过配备惯性检测元件（IMU）以及其它传感器（如磁力计、GPS）采集得到的数据再经过某些算法，从而得到机体的姿态信息（如 Pitch, Roll, Yaw）的过程就叫姿态解算

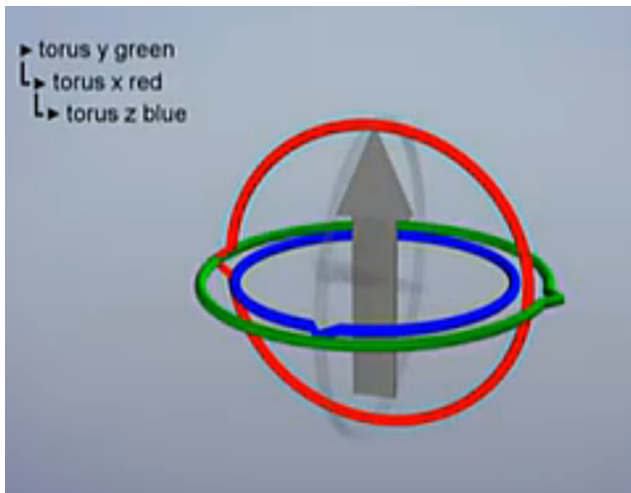
欧拉角

机体从一个坐标系转动到另一个坐标系可以通过依次围绕不同的坐标轴的3次连续转动得到。这三次绕三个轴转动的三个角度就是欧拉角。对于同一个姿态，用不同的欧拉角顺序描述时，对应的欧拉角是不同的。只有将欧拉角应用于惯性导航领域时才有偏航角（heading or yaw）、俯仰角(elevation or pitch)和横滚角(bank or roll)的概念.欧拉角有万向节死锁

万向节死锁（Gimbal Lock）

欧拉角作为姿态解析的致命缺点就是万向节死锁。

如图，当两个轴重合之后，图中灰色阴影的方向就无法通过一次旋转得到



视频

举个例子说明：

假如我们有一个望远镜和一个用来放望远镜的三脚架，（我们将）三脚架放在地面上，使支撑望远镜的三脚架的顶部是平行于地平面（参考平面）的，以便使得竖向的旋转轴（记为x轴）是完全地垂直于地平面的。现在，我们就可以将望远镜绕x轴旋转360度，从而观察（以望远镜为中心的）水平包围圈的所有方向。通常将正北朝向方位角度记为0度方位角。第二个坐标轴，即平行于地平面的横向的坐标轴（记为y轴）使得望远镜可以绕着它上下旋转，通常将地平面朝向的仰角记为0度，这样，望远镜可以向上仰+90度指向天顶，或者向下-90度指向脚底。好了，万事俱备。现在，天空中（包括地面上）的每个点只需要唯一的一对x和y度数就可以确定。比如x=90度,y=45度指向的点是位于正东方向的半天空上。现在，看看万向节死锁是怎么发生的。一次，我们探测到有一个飞行器贴地飞行，位于望远镜的正东方向（x=90度，y=10度），朝着我们直飞过来，我们跟踪它。飞行器飞行方向是保持x轴角度90度不变，而y向的角度在慢慢增大。随着飞行器的临近，y轴角增长的越来越快且当y向的角度达到90度时（即将超越），突然它急转弯朝南飞去。这时，我们发现我们不能将望远镜朝向南方，因为此时y向已经是90度，造成我们失去跟踪目标。这就是万向节死锁！

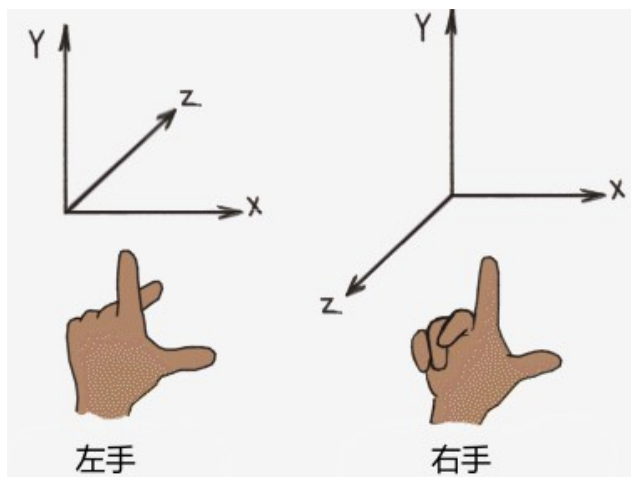
为什么说不能将望远镜朝向南方呢，让我们看看坐标变化，从开始的（x=90度，y=10度）到（x=90度，y=90度），这个过程没有问题，望远镜慢慢转动跟踪飞行器。当飞行器到达（x=90度，y=90度）后，坐标突然变成（x=180度，y=90度）（因为朝南），x由90突变成180度，所以望远镜需要绕垂直轴向x轴旋转180-90=90度以便追上飞行器，但此时，望远镜已经是平行于x轴，我们知道绕平行于自身的中轴线的旋转改变不了朝向，就象拧螺丝一样，螺丝头的指向不变。所以望远镜的指向还是天顶。而后由于飞行器飞远，坐标变成(x=180度，y < 90度)时，y向角减小，望远镜只能又转回到正东指向，望'器'兴叹。这说明用x,y旋转角（又称欧拉角）来定向物体有时并不能按照你想像的那样工作，象上面的例子中从（x=90度，y=10度）到（x=90度，y=90度），按照欧拉角旋转确实可以正确地定向，但从（x=90度，y=90度）到（x=180度，y=90度），再到（x=180度，y < 90度），按照欧拉角旋转后的定向并非正确。我的理解是坐

标值的变化和飞行器空间的位置变化一一对应，但是从（ $x=90^\circ$ ， $y=90^\circ$ ）到（ $x=180^\circ$ ， $y=90^\circ$ ），再到（ $x=180^\circ$ ， $y<90^\circ$ ）这个变化，飞行器位置是连续的变化，但坐标值的变化却不是连续的（从 90° 突变到 180° ），其原因在于（ $x=90^\circ$ ， $y=90^\circ$ ）和（ $x=180^\circ$ ， $y=90^\circ$ ）甚至和（ $x=\text{任意度}$ ， $y=90^\circ$ ）这些不同的坐标值对应空间同一个位置，这种多个坐标值对应同一个位置的不一致性是造成死锁的根源。

上面是2维坐标系中的例子，同样，对于3维的也一样。比如有一个平行于 x 轴的向量，我们先将它绕 y 旋转直到它平行于 z 轴，这时，我们会发现任何绕 z 的旋转都改变不了向量的方向，即万向节死锁，所以说传统的欧拉角是不能做到全姿态解析的

右手坐标系

在空间直角坐标系中，让右手拇指指向 x 轴的正方向，食指指向 y 轴的正方向，如果中指能指向 z 轴的正方向，则称这个坐标系为右手直角坐标系。同理左手直角坐标系



方向余弦矩阵

一个向量在坐标系中的位置也可以用方向余弦表示，也就是这个向量分别到三个坐标轴的夹角余弦值，实际上就是这个向量到各个坐标轴的投影。推广到载体坐标系和参考坐标系当中，我们就有了机体坐标轴 xyz 分别与参考轴 XYZ 的方向余弦，这里就是所说的方向余弦矩阵了，它是由两组不同的标准正交基的基底向量之间的方向余弦所形成的 3×3 矩阵。方向余弦矩阵可以用来表达一组标准正交基与另一组标准正交基之间的关系。余弦矩阵的列表示载体坐标系中的单位矢量在参考坐标系中的投影

机体坐标系(R系) $R=[X_r, Y_r, Z_r]^T$

参考坐标系(O系) $O=[X_o, Y_o, Z_o]^T$

$$C_0^r = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \quad C_r^0 = \begin{bmatrix} C_{11} & C_{21} & C_{31} \\ C_{12} & C_{22} & C_{32} \\ C_{13} & C_{23} & C_{33} \end{bmatrix}$$

方向余弦阵列表示

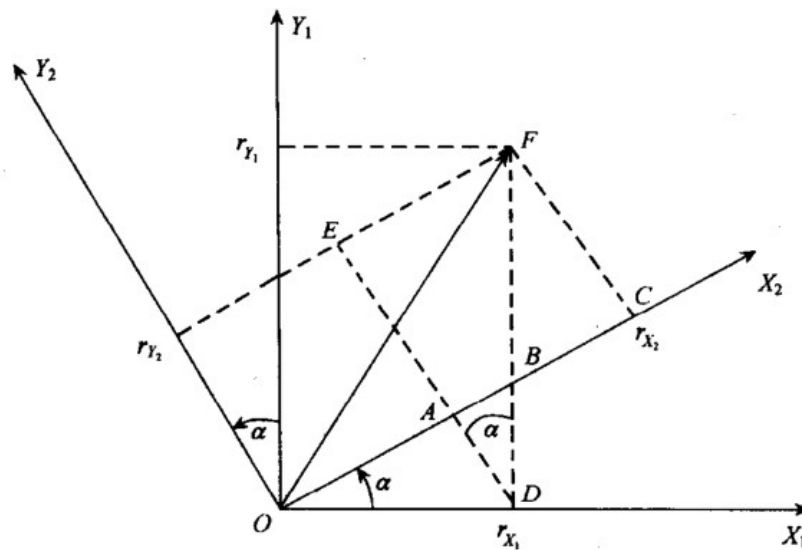
其中 C_0^r 表示坐标系O变化到坐标系R的余弦矩阵，也叫变化矩阵

其中 C_r^0 表示坐标系R变化到坐标系O的余弦矩阵

$$R = C_0^r \cdot O$$

接下来推导一下绕三个轴旋转的方向余弦矩阵

设坐标系 X_1 、 Y_1 、 Z_1 绕 OZ_1 轴旋转 α 角后得到坐标系 X_2 、 Y_2 、 Z_2 ，空间矢量 r 在原坐标系内的投影为 $[r_{x1}, r_{y1}, r_{z1}]^T$ ，在新坐标系内的投影为 $[r_{x2}, r_{y2}, r_{z2}]^T$ ，要求得到两组坐标值之间的关系。由于旋转绕 Z 轴进行，所以 Z 坐标未变， $r_{z1} = r_{z2}$ 。作图：



推倒绕Z旋转矩阵作图

$$\begin{aligned} r_{x2} &= OA + AB + BC \\ &= OD \cos \alpha + BD \sin \alpha + BF \sin \alpha \\ \text{因为 } BD + BF &= r_{y1} \\ &= r_{x1} \cos \alpha + r_{y1} \sin \alpha \end{aligned}$$

$$\begin{aligned} r_{y2} &= DE - AD \\ \text{因为相似三角形可得角ADB等于}\alpha, \text{所以 } DE &= \cos \alpha DF = \cos r_{y1}, \text{在}\triangle AOD \text{中 } AD = \sin \alpha OD = \sin r_{x1} \\ &= \cos r_{y1} - \sin r_{x1} \end{aligned}$$

$$r_{z2} = r_{z1}$$

将上面三个方程写成矩阵形式

$$\begin{bmatrix} r_{x_2} \\ r_{y_2} \\ r_{z_2} \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{x_1} \\ r_{y_1} \\ r_{z_1} \end{bmatrix}$$

绕Z旋转的方向余弦矩阵等式

$$C_1^2 = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

记

C_1^2 称为从坐标系1至坐标系2的变换矩阵。诸元是坐标系1各轴上的单位1在坐标系2各轴上的投影。

$$C_1^2 = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

\longrightarrow X_2 轴上的投影
 \longrightarrow Y_2 轴上的投影
 \longrightarrow Z_2 轴上的投影

\longleftarrow Z_1 轴上的单位 1
 \longleftarrow Y_1 轴上的单位 1
 \longleftarrow X_1 轴上的单位 1

矩阵元素投影关系

两坐标系间任何复杂的角位置关系都可以看做有限次基本旋转的复合，变换矩阵等于基本旋转确定的变换矩阵的连乘，连乘顺序依基本旋转的先后次序由右向左排列。例如机体的空间姿态可看作依次绕航向轴、俯仰轴、横滚轴作基本旋转后的复合结果。

$$O-X_n Y_n Z_n \xrightarrow[\text{旋转 } \Psi]{\text{绕 } -Z_n \text{ 轴}} O-X_1 Y_1 Z_1 \xrightarrow[\text{旋转 } \theta]{\text{绕 } X_1 \text{ 轴}} O-X_2 Y_2 Z_2 \xrightarrow[\text{旋转 } \gamma]{\text{绕 } Y_2 \text{ 轴}} O-X_b Y_b Z_b$$

旋转次序

各次基本旋转对应的变换矩阵:

$$\begin{aligned}
 C_n^1 &= \begin{bmatrix} \cos\Psi & -\sin\Psi & 0 \\ \sin\Psi & \cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 C_1^2 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \\
 C_2^b &= \begin{bmatrix} \cos\gamma & 0 & -\sin\gamma \\ 0 & 1 & 0 \\ \sin\gamma & 0 & \cos\gamma \end{bmatrix}
 \end{aligned}$$

基本旋转矩阵

符号与上面的公式不符合是因为对其进行了转置，也就是改变了系的变化方向

姿态矩阵为:

$$\begin{aligned}
 C_n^b &= C_2^b C_1^2 C_n^1 = \begin{bmatrix} \cos\gamma & 0 & -\sin\gamma \\ 0 & 1 & 0 \\ \sin\gamma & 0 & \cos\gamma \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\Psi & -\sin\Psi & 0 \\ \sin\Psi & \cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos\gamma\cos\Psi + \sin\gamma\sin\Psi\sin\theta & -\cos\gamma\sin\Psi + \sin\gamma\cos\Psi\sin\theta & -\sin\gamma\cos\theta \\ \sin\Psi\cos\theta & \cos\Psi\cos\theta & \sin\theta \\ \sin\gamma\cos\Psi - \cos\gamma\sin\Psi\sin\theta & -\sin\gamma\sin\Psi - \cos\gamma\cos\Psi\sin\theta & \cos\gamma\cos\theta \end{bmatrix}
 \end{aligned}$$

依次绕Z X Y轴旋转

四元素

四元数 (Quaternions) 是由威廉·卢云·哈密顿在1843年发现的数学概念。从明确的角度而言，四元数是复数的不可交换延伸。如把四元数的集合考虑成多维实数空间的话，四元数就代表着一个四维空间，相对于复数为二维空间。

实数域扩充到复数域，并用复数来表示平面向量，用复数的加、乘运算表示平面向量的合成、伸缩和旋，这就是我们熟知的复数的二维空间含义，所以人们会继续猜想，利用三维复数不就可以表达三维空间的变换了吗，历史上有很多数学家试图寻找过三维的复数，但后来证明这样的三维复数是不存在的。有关这个结论的证明，我没有查到更明确的版本，据《古今数学思想》中的一个理由，三维空间中的伸缩旋转变换需要四个变量来决定：两个变量决定轴的方向，一个变量决定旋转角度，一个变量决定伸缩比例。这样，只有三个变量的三维复数无法满足这样的要求。但是历史上得到的应该是比这个更强的结论，即使不考虑空间旋转，只从代数角度来说，三维的复数域作为普通复数域的扩张域是不存在的。并且，据《古今数学思想》叙述，即使像哈密顿后来引入四元数那样，牺牲乘法交换律，这样的三维复数也得不到。经过一些年的努力之后，Hamilton 发现自己被迫应作两个让步，第一个是他的新数包含四个分量，而第二个是他必须牺牲乘法交换律。

(《古今数学思想》第三册177页)但是四元数用作旋转的作用明显，简化了运算，而且避免了万向锁 (Gimbal Lock)，四元数是最简单的超复数，我们不能把四元数简单的理解为3D空间的矢量，它是4维空间中的的矢量，也是非常不容易想像的。

四元数都是由实数加上三个元素 i、j、k 组成，他们有下列关系

$$i^2=j^2=k^2=ijk=-1$$

每个四元数都是 i、j 和 k 的线性组合，即是四元数一般可表示为

$$a+bi+cj+dk$$

我们知道在平面(x,y)中的旋转可以用复数来表示，同样的三维中的旋转可以用单位四元数来描述。我们来定义一个四元数：

$$\mathbf{q} = a + \vec{u} = q_0 + q_1 i + q_2 j + q_3 k$$

四元素的表示

上式可以写作 $[\mathbf{W}, \mathbf{V}]$ ，其中 $\mathbf{V} = q_1 i + q_2 j + q_3 k$ ， $w = a = q_0$ 。

\mathbf{v} 是矢量，表示三维空间中的旋转轴。 w 是标量，表示旋转角度。那么 $[\mathbf{W}, \mathbf{V}]$ 就是绕轴 \mathbf{V} 旋转 w 度，所以四元素可以表示一个完整的旋转。并且只有单位四元素才可以表示旋转，这是四元素表示旋转的约束条件。

如何求单位四元素呢，如下：

四元数可以理解为一个实数和一个向量的组合，也可以理解为四维的向量。这里用一个圈表示 \mathbf{q} 是一个四元数

$$\mathring{\mathbf{q}} = \{w, \vec{v}\} = [w \ x \ y \ z]^T$$

四元数的长度（模）与普通向量相似。

$$|\mathring{\mathbf{q}}| = \sqrt{w^2 + x^2 + y^2 + z^2}$$

下面是对四元数的单位化，单位化的四元数可以表示一个旋转。

$$\hat{\mathring{\mathbf{q}}} = \frac{\mathring{\mathbf{q}}}{|\mathring{\mathbf{q}}|} = \left[\frac{w}{|\mathring{\mathbf{q}}|} \ \frac{x}{|\mathring{\mathbf{q}}|} \ \frac{y}{|\mathring{\mathbf{q}}|} \ \frac{z}{|\mathring{\mathbf{q}}|} \right]^T$$

四元素相乘可以用来组合旋转

$$\mathring{\mathbf{q}}_0 = \mathring{\mathbf{q}}_1 \cdot \mathring{\mathbf{q}}_2$$

$$\begin{cases} w_0 = w_1 w_2 - x_1 x_2 - y_1 y_2 - z_1 z_2 \\ x_0 = w_1 x_2 + x_1 w_2 + y_1 z_2 - z_1 y_2 \\ y_0 = w_1 y_2 - x_1 z_2 + y_1 w_2 + z_1 x_2 \\ z_0 = w_1 z_2 + x_1 y_2 - y_1 x_2 + z_1 w_2 \end{cases}$$

四元数大量用于电脑绘图（及相关的图像分析）上表示三维物件的旋转及方位。四元数亦见于控制论、信号处理、姿态控制、物理和轨道力学，都是用来表示旋转和方位。

相对于另几种旋转表示法（矩阵，欧拉角，轴角），四元数具有某些方面的优势，如速度更快、提供平滑插值、有效避免万向锁问题、存储空间较小等等。

陀螺仪

陀螺仪，测量角速度，具有高动态特性，它是一个间接测量角度的器件。它测量的是角度的导数，即角速度，要将角速度对时间积分才能得到角度。

陀螺仪就是内部有一个陀螺，它的轴由于陀螺效应始终与初始方向平行，这样就可以通过与初始方向的偏差计算出旋转方向和角度。传感器MPU6050实际上是一个结构非常精密的芯片，内部包含超微小的陀螺。

如果这个世界是理想的，美好的，那我们的问题到此就解决了，从理论上讲只用陀螺仪是可以完成姿态导航的任务的。只需要对3个轴的陀螺仪角速度进行积分，得到3个方向上的旋转角度，也就是姿态数据。这也就是说的快速融合。

不过很遗憾，现实是残酷的，由于误差噪声等的存在，对陀螺仪积分并不能够得到完全准确的姿态，尤其是运转一段时间以后，积分误差的累加会让得到的姿态和实际的相差甚远。

那么哪些原因会使陀螺仪得到的姿态结果不准确呢？下面列举除常见的几种：

- 零点漂移
假设陀螺仪固定不动，理想角速度值是0dps(degree per second)，但是存在零点漂移，例如有一个偏置0.1dps加在上面，于是测量出来是0.1dps，积分一秒之后，得到的角度是0.1度，1分钟之后是6度，还能忍受，一小时之后是360度，转了一圈，也就是说，陀螺仪在短时间有很大的参考价值。
- 白噪声
电信号的测量中，一定会带有白噪声，陀螺仪数据的测量也不例外。所以获得的陀螺仪数据中也会带有白噪声，而且这种白噪声会随着积分而累加。
- 温度/加速度影响
陀螺仪是一个温度和加速度敏感的元器件。例如对于加速度，多轴飞行器中的马达一般会带来较强烈的振动，一旦减震控制不好，就会在飞行过程中产生很大的加速度，必会带来陀螺输出的变化，引入误差。这就是在陀螺仪数据手册上常见的deg/sec/g指标。
- 积分误差
对陀螺仪角速度的积分是离散的，长时间的积分会出现漂移的情况。所以要考虑积分误差的问题。
这是由于陀螺仪测量姿态存在这么多的误差，所以我们必须要使用其它传感器辅助校正，其中最重要的就是下面的加速度传感器。

加速度计

加速度计的低频特性好，可以测量低速的静态加速度。在我们的飞行器上，就是对重力加速度 g （也就是前面说的静态加速度）的测量和分析，其它瞬间加速度可以忽略。记住这一点对姿态解算融合理解非常重要。

当我们把加速度计拿在手上随意转动时，我们看的是重力加速度在三个轴上的分量值。加速度计在自由落体时，其输出为0。为什么会这样呢？这里涉及到加速度计的设计原理：加速度计测量加速度是通过比力来测量，而不是通过加速度。

加速度计仅仅测量的是重力加速度，3轴加速度计输出重力加速度在加速度计所在机体坐标系3个轴上的分量大小。重力加速度的方向和大小是固定的。通过这种关系，可以得到加速度计所在平面与地面的角度关系。

加速度计若是绕着重力加速度的轴转动，则测量值不会改变，也就是说加速度计无法感知这种水平旋转。

有关陀螺仪和加速度计和关系，姿态解算融合的原理，再把下面这个比喻放到这里一遍。

机体好似一条船，姿态就是航向（船头的方位），重力是灯塔，陀螺（角速度积分）是舵手，加速度计是瞭望手。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。

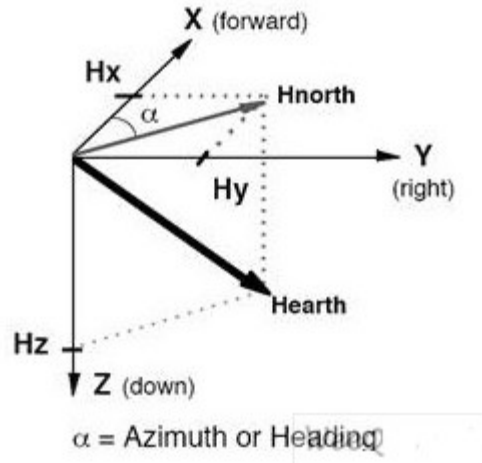
磁力计

基础知识

磁力计（Magnetic、M-Sensor）也叫地磁、磁感器，可用于测试磁场强度和方向，定位设备的方位，磁力计的原理跟指南针原理类似，可以测量出当前设备与东南西北四个方向上的夹角。所以，陀螺仪知道“我们转了个身”，加速计知道“我们又向前走了几米”，而磁力计则知道“我们是向西方向”的。

地球的磁场象一个条形磁体一样由磁南极指向磁北极。在磁极点处磁场和当地的水平面垂直，在赤道磁场和当地的水平面平行，所以在北半球磁场方向倾斜指向地面。用来衡量磁感应强度大小的单位是Tesla或者Gauss（1Tesla=10000Gauss）。随着地理位置的不同，通常地磁场的强度是0.4-0.6 Gauss。需要注意的是，磁北极和地理上的北极并不重合，通常他们之间有11度左右的夹角。

地磁场是一个矢量，对于一个固定的地点来说，这个矢量可以被分解为两个与当地水平面平行的分量和一个与当地水平面垂直的分量。如果保持电子罗盘和当地的水平面平行，那么罗盘中磁力计的三个轴就和这三个分量对应起来



磁力计分量.jpg

实际上对水平方向的两个分量来说，他们的矢量和总是指向磁北的。罗盘中的航向角（Azimuth）就是当前方向和磁北的夹角。由于罗盘保持水平，只需要用磁力计水平方向两轴（通常为X轴和Y轴）的检测数据就可以计算出航向角。当罗盘水平旋转的时候，航向角在0- 360之间变化

磁场干扰

电子指南针主要是通过感知地球磁场的存在来计算磁北极的方向。然而由于地球磁场在一般情况下只有微弱的0.5高斯，而一个普通的手机喇叭当相距2厘米时仍会有大约4高斯的磁场，一个手机马达在相距2厘米时会有大约6高斯的磁场，这一特点使得针对电子设备表面地球磁场的测量很容易受到电子设备本身的干扰

校准方式

这里之谈笔者使用的校准方式

首先需要收集磁力计在各个轴上的最大最小值，之后找出反应最大的一轴数据

将其比例值设为1，其余的轴的比例值由反应最大轴的最大最小值之差除以该轴的最大最小值只差来求得。之后对其偏置补偿，就是找到一元一次函数中的那个常数。如果设其中一个轴的比例系数为 X_s ，那么偏置数 X_b 的值为

$$X_b = X_s [1/2(X_{\max} - X_{\min}) - X_{\max}]$$

最后对原始值进行修正：

$$X_{\text{out}} = X_{\text{in}} * X_s + X_b$$

MPU6050

MPU-60x0是全球首例6轴运动处理传感器。它集成了3轴MEMS陀螺仪，3轴MEMS加速度计，以及一个可扩展的数字运动处理器DMP（Digital Motion Processor），可用I2C接口连接一个第三方的数字传感器，比如磁力计。扩展之后就可以通过其I2C或SPI接口输出一个6轴的信号（SPI接口仅在MPU-6000可用）。MPU-60x0也可以通过其I2C接口连接非惯性的数字传感器，比如压力传感器。

MPU-60x0对陀螺仪和加速度计分别用了三个16位的ADC，将其测量的模拟量转化为可输出的数字量。为了精确跟踪快速和慢速的运动，传感器的测量范围都是用户可控的，陀螺仪可测范围为 ± 250 ， ± 500 ， ± 1000 ， $\pm 2000^\circ/\text{秒}$ （dps），加速度计可测范围为 ± 2 ， ± 4 ， ± 8 ， $\pm 16g$ 。一个片上1024字节的FIFO，有助于降低系统功耗。和所有设备寄存器之间的通信采用400kHz的I2C接口或1MHz的SPI接口（SPI仅MPU-6000可用）。对于需要高速传输的应用，对寄存器的读取和中断可用20MHz的SPI。另外，片上还内嵌了一个温度传感器和在工作环境下仅有 $\pm 1\%$ 变动的振荡器。

比如在开源飞控crazepony上，MPU6050/HMC5883/MS5611传感器之间的连接

MPU9150

MPU-9150整合了MPU6050及AK8975电子罗盘功能(三轴陀螺仪 + 三轴加速度+三轴磁场)，IIC通信，芯片内置16bit AD转换器,16位数据输出（磁场13位）

陀螺仪范围： ± 250 500 1000 2000 °/s

加速度范围： $\pm 2 \pm 4 \pm 8 \pm 16g$

磁场范围： $\pm 1200 \mu T$

MPU9250

MPU9150是只支持I2C的，MPU9250是支持SPI/I2C两种方式。里面的传感器也是不同的，MPU9150里面是MPU6050+AK8975，而MPU9250里面是MPU6500+AK8963，这两个传感器组合不同，前者性能上要高一些，后者主打低功耗方面的，各种参数要略低一些，比如唤醒速度等。

辅助理解

四元素与姿态阵间的关系

关系推导详细请看<<惯性导航>>292页 9.2.2章节。

用四元素来表示的变化矩阵公式为：

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

1468994043757.jpg

其中R是参考坐标系，对于方向余弦矩阵中的导航坐标系n，b系就是机体坐标系。

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，所以矩阵为正交矩阵也就是 $C_n^b = (C_b^n)^T$ 。

下面是绕三轴旋转的方向余弦矩阵

$$\begin{aligned} C_n^b &= C_2^b C_1^n C_n^1 = \begin{bmatrix} \cos\gamma & 0 & -\sin\gamma \\ 0 & 1 & 0 \\ \sin\gamma & 0 & \cos\gamma \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\Psi & -\sin\Psi & 0 \\ \sin\Psi & \cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos\gamma\cos\Psi + \sin\gamma\sin\Psi\sin\theta & -\cos\gamma\sin\Psi + \sin\gamma\cos\Psi\sin\theta & -\sin\gamma\cos\theta \\ \sin\Psi\cos\theta & \cos\Psi\cos\theta & \sin\theta \\ \sin\gamma\cos\Psi - \cos\gamma\sin\Psi\sin\theta & -\sin\gamma\sin\Psi - \cos\gamma\cos\Psi\sin\theta & \cos\gamma\cos\theta \end{bmatrix} \end{aligned}$$

依次绕Z X Y轴旋转

先统一变换方向为参考坐标系到机体坐标系

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

1468994919345.jpg

右方向余弦矩阵公式可得

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

1468994967054.jpg

将对应的余弦矩阵元素变为四元素矩阵的元素，就可以得到四元素与欧拉角的关系。也就是说，当参考坐标系到机体坐标系的四元素Q已经确定的时候，便可以计算出机体的航向角、俯仰角、横滚角。四元素包含了所有的姿态信息，捷联惯导中的姿态更新实质是如何计算四元素。

四元素的更新

<<惯性导航>>书中的四元素微分方程写的很详细，但这里就直接列出最后结果。利用龙格库塔更新四元数，只需要将传入上一次的四元数和新得到的角速度值再填上采样周期就可以更新四元数

最后的推倒公式为：

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

龙格库塔更新四元数

下面附上部分推导供参考

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

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

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

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

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

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

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

四元数的初值确定

四元数的初值Q (0) 由捷联惯导的初始对准确定。设初始对准确定的姿态阵为Cⁿ_b[T_{ij}] , 根据四元数矩阵方程有如下方程成立：

$$\left\{ \begin{array}{l} q_0^2 + q_1^2 - q_2^2 - q_3^2 = T_{11} \\ q_0^2 - q_1^2 + q_2^2 - q_3^2 = T_{22} \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 = T_{33} \\ q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \\ 2(q_1q_2 - q_0q_3) = T_{12} \\ 2(q_1q_3 + q_0q_2) = T_{13} \\ 2(q_1q_2 + q_0q_3) = T_{21} \\ 2(q_2q_3 - q_0q_1) = T_{23} \\ 2(q_1q_3 - q_0q_2) = T_{31} \\ 2(q_2q_3 + q_1q_0) = T_{32} \end{array} \right.$$

1469022062269.jpg

由上述方程可解的

$$\left\{ \begin{array}{l} |q_1| = \frac{1}{2} \sqrt{1 + T_{11} - T_{22} - T_{33}} \\ |q_2| = \frac{1}{2} \sqrt{1 - T_{11} + T_{22} - T_{33}} \\ |q_3| = \frac{1}{2} \sqrt{1 - T_{11} - T_{22} + T_{33}} \\ |q_0| = \frac{1}{2} \sqrt{1 + T_{11} + T_{22} + T_{33}} \end{array} \right.$$

$$\left\{ \begin{array}{l} 4q_1q_0 = T_{32} - T_{23} \\ 4q_2q_0 = T_{13} - T_{31} \\ 4q_3q_0 = T_{21} - T_{12} \end{array} \right.$$

1469022197666.jpg

所以 q_0 、 q_1 、 q_2 、 q_3 的符号可按下式确定

$$\begin{cases} \text{sign}(q_1) = \text{sign}(q_0)[\text{sign}(T_{32} - T_{23})] \\ \text{sign}(q_2) = \text{sign}(q_0)[\text{sign}(T_{13} - T_{31})] \\ \text{sign}(q_3) = \text{sign}(q_0)[\text{sign}(T_{21} - T_{12})] \end{cases}$$

1469022482706.jpg

上式中， $\text{sign}(q_0)$ 的符号可任选

四元数的规范化

表征旋转的四元数应该是规范化四元数，即 $\|Q\| = 1$ ，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征，因此若干次更新后，必须对四元数做规范化处理：

$$q_i = \frac{q_i}{\sqrt{\hat{q}_0^2 + \hat{q}_1^2 + \hat{q}_2^2 + \hat{q}_3^2}}, \quad i = 0, 1, 2, 3$$

其中， $\hat{q}_0, \hat{q}_1, \hat{q}_2, \hat{q}_3$ 是四元数更新所得的值。

1469022758001.jpg

相关数学知识

矩阵乘法

设 A 为 $n \times m$ 的矩阵， B 为 $m \times p$ 的矩阵，那么称 $n \times p$ 的矩阵 C 为矩阵 A 与 B 的乘积，记作

$C=AB$ ，只有在矩阵 A 的列数和矩阵 B 的行数相同时才有意义

A 矩阵从第一行开始，分别与矩阵 B 第一列的每个元素分别相乘求和，组成新矩阵的第一行第一列的元素，依次计算。

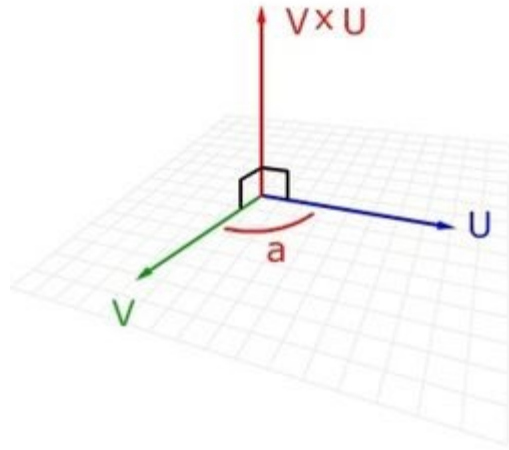
例如

$$C = AB = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix} = \begin{pmatrix} 1 \times 1 + 2 \times 2 + 3 \times 3 & 1 \times 4 + 2 \times 5 + 3 \times 6 \\ 4 \times 1 + 5 \times 2 + 6 \times 3 & 4 \times 4 + 5 \times 5 + 6 \times 6 \end{pmatrix} = \begin{pmatrix} 14 & 32 \\ 32 & 77 \end{pmatrix}$$

矩阵乘法举例

有上述方程可解得

向量积（外积、叉积、叉乘）



叉乘

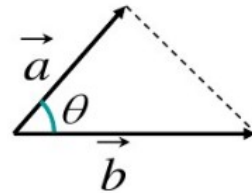
定义: 设 \vec{a}, \vec{b} 的夹角为 θ ,

$$\text{向量 } \vec{c} \begin{cases} \text{方向: } \vec{c} \perp \vec{a}, \vec{c} \perp \vec{b} \text{ 且符合右手规则} \\ \text{模: } |\vec{c}| = |\vec{a}| |\vec{b}| \sin \theta \end{cases}$$

叉乘定义

几何意义: 右图三角形面积

$$S = \frac{1}{2} |\vec{a} \times \vec{b}|$$



叉乘几何意义

举例:

向量: $u=(u_1, u_2, u_3)$ $v=(v_1, v_2, v_3)$

叉积公式: $u \times v = \{ u_2v_3 - v_2u_3, u_3v_1 - v_3u_1, u_1v_2 - u_2v_1 \}$

点积公式: $u \cdot v = u_1v_1 + u_2v_2 + u_3v_3 = |u||v|\cos(\theta)$

解算中用到叉乘主要是用来求误差

四元素基本运算

定义两个四元数:

$$q = a + \vec{u} = a + bi + cj + dk$$

$$p = t + \vec{v} = t + xi + yj + zk$$

1468990296793.jpg

加法：

四元数加法: $p + q$

跟复数、向量和矩阵一样，两个四元数之和需要将不同的元素加起来：

$$p + q = a + t + \vec{u} + \vec{v} = (a + t) + (b + x)i + (c + y)j + (d + z)k$$

加法遵循实数和复数的所有交换律和结合律。

四元素加法

乘法

列为乘数，行为被乘数

×	1	i	j	k
1	1	i	j	k
i	i	-1	k	-j
j	j	-k	-1	i
k	k	j	-i	-1

乘法表

两个四元数之间的非可换乘积通常被称为格拉斯曼积

$$pq = at - \vec{u} \cdot \vec{v} + a\vec{v} + t\vec{u} + \vec{v} \times \vec{u}$$

$$pq = (at - bx - cy - dz) + (bt + ax + dy - cz)i + (ct + ay + bz - dx)j + (dt + za + cx - by)k$$

模|p|

四元数的绝对值是四元数到原点的距离

$$|p| = \sqrt{p \cdot p} = \sqrt{p^* p} = \sqrt{a^2 + b^2 + c^2 + d^2}$$

四元素模的公式

例子

假设：

$$x = 3 + i$$

$$y = 5i + j - 2k$$

那么：

$$x + y = 3 + 6i + j - 2k$$

$$xy = (3 + i)(5i + j - 2k) = 15i + 3j - 6k + 5i^2 + ij - 2ik$$

$$= 15i + 3j - 6k - 5 + k + 2j = -5 + 15i + 5j - 5k$$

加法和乘法举例

均值、标准差、方差

均值描述的是样本集合的中间点，它告诉我们的信息是有限的，而标准差给我们描述的是样本集合的各个样本点到均值的距离之平均。方差是衡量源数据和期望值相差的度量值。

平均数： $M = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n}$ （ n 表示这组数据个数， x_1 、 x_2 、 x_3 x_n 表示这组数据具体数值）

方差公式： $s^2 = \frac{(x_1 - M)^2 + (x_2 - M)^2 + (x_3 - M)^2 + \dots + (x_n - M)^2}{n}$

方差公式

也可以写作

均值： $\bar{X} = \frac{\sum_{i=1}^n X_i}{n}$

标准差： $S = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}}$

方差： $S^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}$

协方差

标准差和方差一般是用来描述一维数据的，但现实生活中我们常常会遇到含有多维数据的数据集，最简单的是大家上学时免不了要统计多个学科的考试成绩。面对这样的数据集，我们当然可以按照每一维独立的计算其方差，但是通常我们还想了解更多，比如，一个男孩子的猥琐程度跟他受女孩子的欢迎程度是否存在一些联系。协方差就是这样一种用来度量两个随机变量关系的统计量，我们可以仿照方差的定义：

$$\text{var}(X) = \frac{\sum_{i=1}^n (X_i - \bar{X})(X_i - \bar{X})}{n-1}$$

来度量各个维度偏离其均值的程度，协方差可以这样来定义：

$$\text{cov}(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{n-1}$$

协方差的结果有什么意义呢？如果结果为正值，则说明两者是正相关的（从协方差可以引出“相关系数”的定义），也就是说一个人越猥琐越受女孩欢迎。如果结果为负值，就说明两者是负相关，越猥琐女孩子越讨厌。如果为0，则两者之间没有关系，猥琐不猥琐和女孩子喜不喜欢之间没有关联，就是统计上说的“相互独立”。

协方差矩阵

前面提到的猥琐和受欢迎的问题是典型的二维问题，而协方差也只能处理二维问题，那维数多了自然就需要计算多个协方差，给出协方差矩阵的定义：

$$C_{n \times n} = (c_{i,j}, \quad c_{i,j} = \text{cov}(\text{Dim}_i, \text{Dim}_j))$$

我们可以举一个三维的例子，假设数据集有三个维度，则协方差矩阵为：

$$C = \begin{pmatrix} \text{cov}(x,x) & \text{cov}(x,y) & \text{cov}(x,z) \\ \text{cov}(y,x) & \text{cov}(y,y) & \text{cov}(y,z) \\ \text{cov}(z,x) & \text{cov}(z,y) & \text{cov}(z,z) \end{pmatrix}$$

可见，协方差矩阵是一个对称的矩阵，而且对角线是各个维度的方差。

弧度

在数学和物理中，弧度是角的度量单位。它是由国际单位制导出的单位，单位缩写是rad。根据定义，一周的弧度数为 $2\pi r/r=2\pi$ ， $360^\circ=2\pi$ 弧度，因此，1弧度约为 57.3° ，即 $57^\circ 17' 44.806''$ ， 1° 为 $\pi/180$ 弧度，近似值为0.01745弧度，周角为 2π 弧度，平角（即 180° 角）为 π 弧度，直角为 $\pi/2$ 弧度。

姿态解算（融合）参考算法

Mahony互补滤波算法

发现很多组织都使用这个人写的姿态解算，融合了三轴加速度计、三轴陀螺仪、三轴磁力计，我们就来分析一下解算的流程。

代码示例见[MahonyAHRS.h](#)

1.初始化四元素

2.用磁力计求出当前姿态与参考向量的偏差

- 三轴磁力计值归一化

参考代码

```
recipNorm = 1.0 / sqrt(mx * mx + my * my + mz * mz);  
mx *= recipNorm;  
my *= recipNorm;  
mz *= recipNorm;
```

- 将载体坐标系变为地面坐标系

公式来源于四元素变化矩阵

```
hx = 2.0f * (mx * (0.5f - q2q2 - q3q3) + my * (q1q2 - q0q3) + mz * (q1q3 + q0q2));  
hy = 2.0f * (mx * (q1q2 + q0q3) + my * (0.5f - q1q1 - q3q3) + mz * (q2q3 - q0q1));  
hz = 2.0f * mx * (q1q3 - q0q2) + 2.0f * my * (q2q3 + q0q1) + 2.0f * mz * (0.5f - q1q1 - q2q2);
```

- 算出北向向量作为一个参考向量

合成了X和Y轴，得到一个指向北方的向量

```
bx = sqrt(hx * hx + hy * hy);  
bz = hz;
```

- 将参考向量变为机体坐标系

```
halfwx = bx * (0.5f - q2q2 - q3q3) + bz * (q1q3 - q0q2);  
halfwy = bx * (q1q2 - q0q3) + bz * (q0q1 + q2q3);  
halfwz = bx * (q0q2 + q1q3) + bz * (0.5f - q1q1 - q2q2);
```

由于要在同一个坐标系下作对比，所以要将求得的参考向量转化为机体坐标系下的情况

- 在机体坐标系下参考向量和姿态向量叉乘求得误差

归一化之后使用叉乘求得的便是两个向量的偏差值，将这个偏差值先保存起来

```
halfex += (my * halfwz - mz * halfwy);
halfey += (mz * halfwx - mx * halfwz);
halfez += (mx * halfwy - my * halfwx);
```

3. 用加速度计求出这一次与上一次重力向量的偏差

- 三轴加速度值归一化

这里和磁力计一样，归一化是为了只考虑向量的方向

```
recipNorm = 1.0 / sqrt(ax * ax + ay * ay + az * az);

ax *= recipNorm;
ay *= recipNorm;
az *= recipNorm;
```

上一次四元素在机体坐标系下换算出来的重力的单位向量

这里通过四元数矩阵的第三列求得上一次的重力向量也就是Z轴，要注意的是机体坐标系下，所以矩阵要按照需求进行转置，当你发现有的时候是取得第三行，而有的时候是第三列的时候，就是这个原因了。

```
halfvx = q1q3 - q0q2;
halfvy = q0q1 + q2q3;
halfvz = q0q0 - 0.5f + q3q3;
```

- 当前的重力的单位向量与上一次的做叉乘
求出这一次和上一次的偏差值，并且累加偏差值

```
halfex += ay * halfvz - az * halfvy;
halfey += az * halfvx - ax * halfvz;
halfez += ax * halfvy - ay * halfvx;
```

- 将该误差累加到磁力计求得的误差中去
这里将磁力计算出来的偏差和加速度计算出来的偏差相加合成了一个总的偏差用来修正陀螺仪，也就是说所用的磁力计和角速度计修正陀螺仪了

4. 通过PI控制器使用上面两种偏差修正陀螺仪数据

到了这里需要的数据都已经凑齐，那么便使用PI控制器来对数据进行融合了。

```
gyro_bias[0] += twoKi * halfex * dt;
gyro_bias[1] += twoKi * halfey * dt;
gyro_bias[2] += twoKi * halfez * dt;
```

```
gx += gyro_bias[0];
gy += gyro_bias[1];
gz += gyro_bias[2];
```

积分环节

```
gx += twoKp * halfex;
gy += twoKp * halfey;
gz += twoKp * halfez;
```

比例环节

通过这一步就能得到修正后的陀螺仪数据，也就是足以信赖的数据了

5. 使用龙格库塔公式更新四元数

使用上面得到的可靠角速度便能更新四元数了，这里使用龙格库塔公式来更新

```
dq0 = 0.5f*(-q1 * gx - q2 * gy - q3 * gz);
dq1 = 0.5f*(q0 * gx + q2 * gz - q3 * gy);
dq2 = 0.5f*(q0 * gy - q1 * gz + q3 * gx);
dq3 = 0.5f*(q0 * gz + q1 * gy - q2 * gx);

q0 += dt*dq0;
q1 += dt*dq1;
q2 += dt*dq2;
q3 += dt*dq3;
```

得到的四元素便包含了机体的转动信息

6. 归一化四元数并更新相关数据

为啥要归一化四元素上面有讲，方法上面也已经列出，这里就直接看代码吧

```
recipNorm = 1.0 / sqrt(q0 * q0 + q1 * q1 + q2 * q2 + q3 * q3);
q0 *= recipNorm;
q1 *= recipNorm;
q2 *= recipNorm;
q3 *= recipNorm;
```

7. 通过四元数旋转矩阵得到姿态角

最后就是来求得我们期望的姿态角了，这里是通过方向余弦矩阵和四元数矩阵来求得的，详细可以看上面四元素和姿态阵间的关系

```
mAngle.x = asin(-2 * q1 * q3 + 2 * q0 * q2) * RtA; // pitch
mAngle.y = atan2(2 * q2 * q3 + 2 * q0 * q1, -2 * q1 * q1 - 2 * q2 * q2 + 1) * RtA; // roll
mAngle.z = atan2(2 * (q0 * q3 + q1 * q2), 1 - 2 * (q2 * q2 + q3 * q3)) * RtA; // yaw
```

注意的是三角函数需要把它换算成弧度制哦。

整理：[lisettecarlr](#) & [Neurack](#)