目錄

简介

本站是对 `William Premerlani 和 Paul Bizard` 的Direction Cosine Matrix IMU: Theory文章的翻译。

> 访问地址：dcm.nephen.com

文章用处

ArduPilot的DCM库算法实现参考文章。

译文下载

该翻译文章会生成pdf文档，请转至下载地址。

编辑语法

Markdown语法，数学公式采用MathJax1/MathJax2。

翻译进度

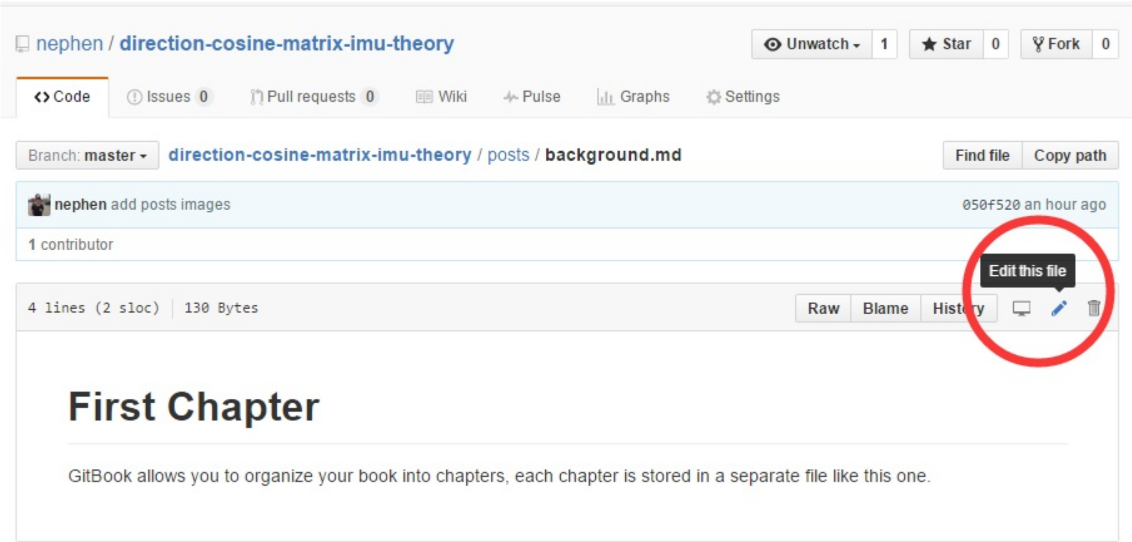
- 2016.5.8 开始翻译
- 2016.6.8 预计初版完成

参与维护

1. 网页端编辑（推荐方法：非常简单，只需三步就可以完成你的贡献）
    - 找到入口：浏览在线页面时编辑本页，这样，看到不妥的地方可以立马修改，如下图

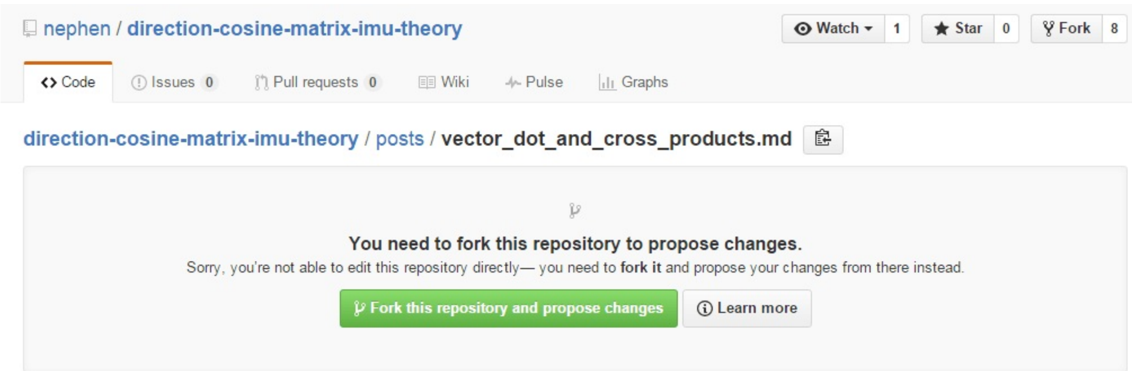
      

      或者也可以打开网页端进入要进行编辑的文件，如下图



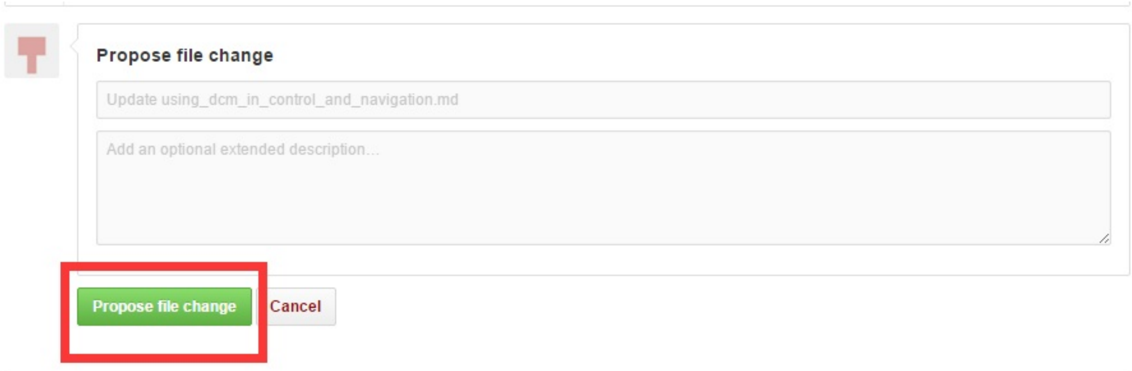
，

- 开始编辑：进入编辑页面后，对文件进行修改，会出现如下图问题

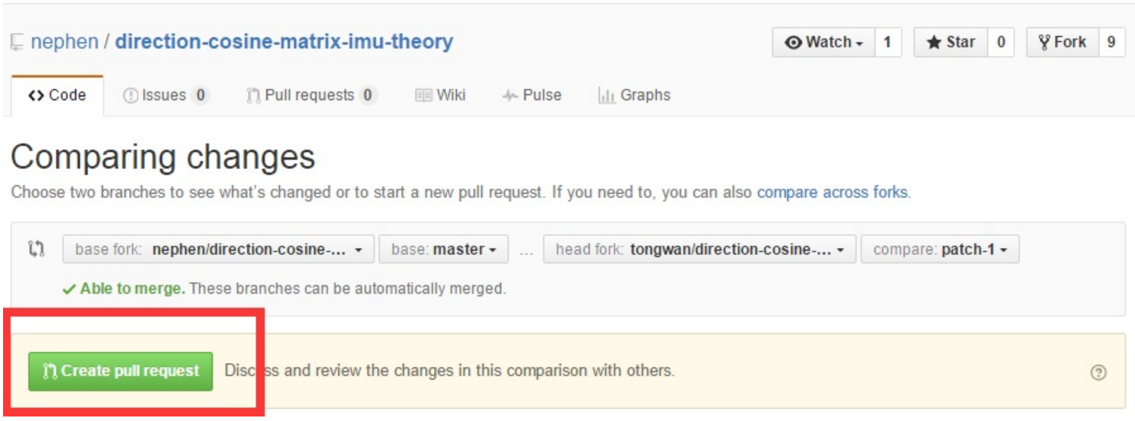


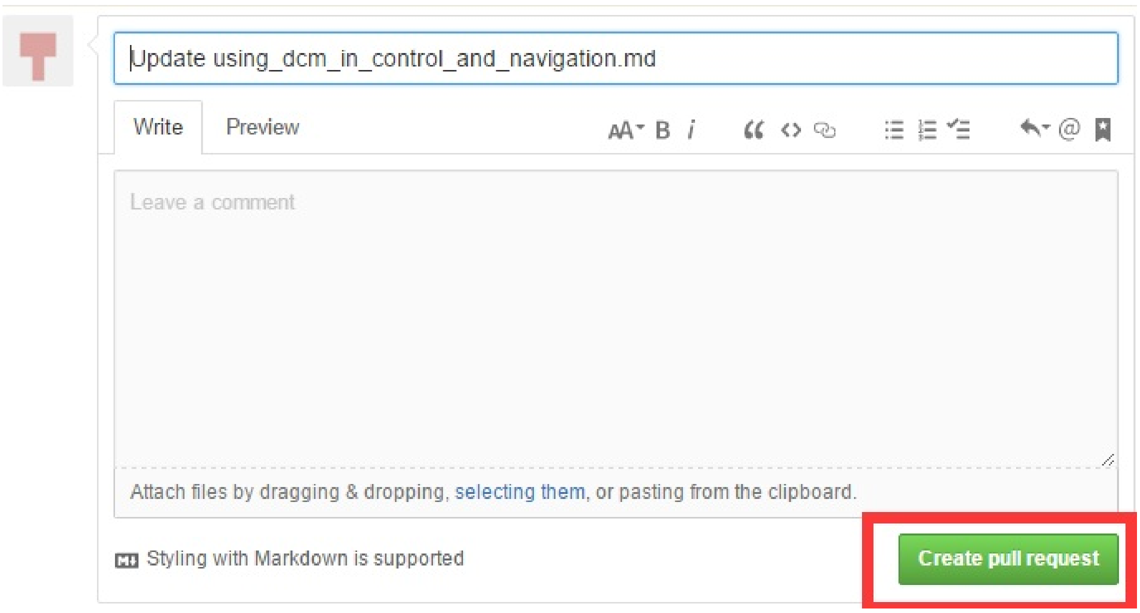
按照提示，点击 `Fork this repository and propose chanes` ，进入即可编辑，如下图

- 提交贡献：首先提名文件更改，如下图



检查是否与原有版本有冲突，如果有，解决冲突再提交，没有则提交，如下图

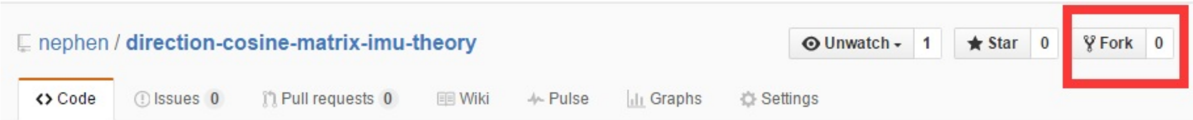




剩下来就是版主的事了，如果没有太大的问题，版主就可以合并分支了，到这你的对本文档的贡献就完成了。

2. 本地编辑（git高级用户推荐）

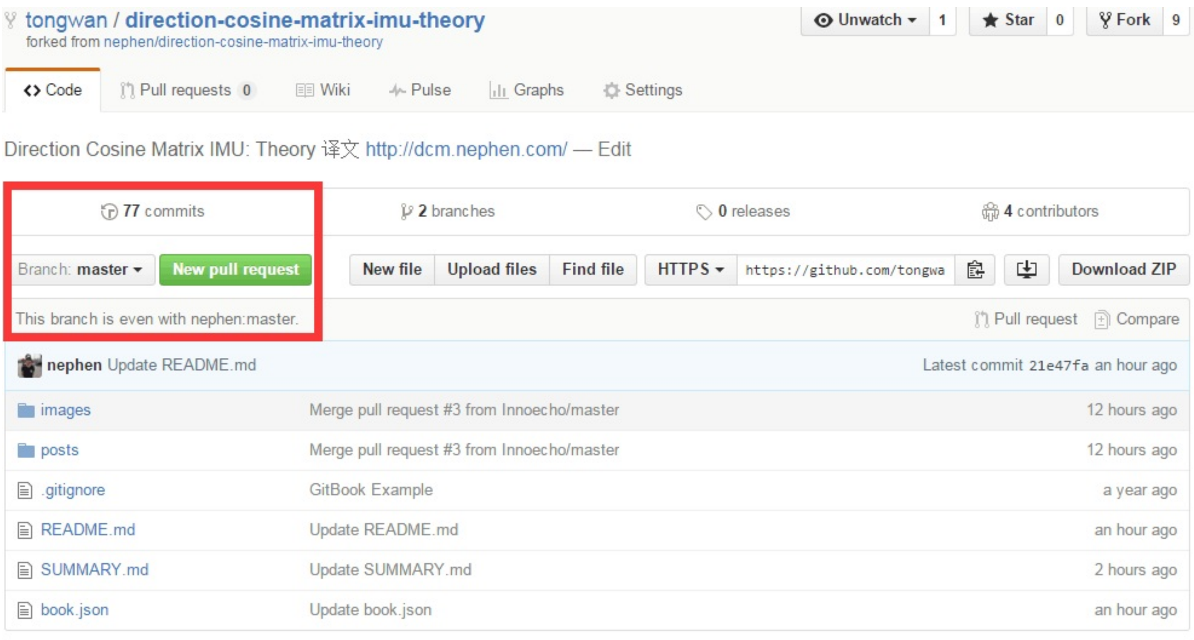
相对于网页端编辑，本地编辑只是编辑在本地，后期的提交分支还是得在网页端进行，不过在此之前你得fork本项目到你的仓库。



然后进行如下操作

```
#下载你的项目到本地
git clone https://github.com/your_github_username/direction-cosine-matrix-imu-the
ory.git

#进入文件夹进行编辑即可，完成后如下操作

git add .
#这里可以看到你的更改状况

git status
#添加你的更改备注，让别人知道你干了什么

git commit -m "your comment"
#提交更改

git pull https://github.com/nephen/direction-cosine-matrix-imu-theory.git master
#检查是否与nephen云端产生冲突，如果有，解决冲突后重新git commit -m "your comment"

git push origin master
#推送到个人云端
```

到这里为止，还只对你自己的仓库进行了修改，你需要 `new pull request` 提交分支到nephen的仓库，如下图，可以看出，如果只是少量的更改，建议使用网页端编辑。



参考资料

1. 关于gitbook，可查看www.gitbook.com。
2. gitbook的官方使用，可查看https://help.gitbook.com。

License

> 最后感谢Innoecho、小羊、小飞、小段、一字并肩王、大牛、小孟、luoshi006、落落的主导翻译！

这是基于惯性测量单元（IMU）的方向余弦矩阵（DCM）的理论与实现的系列文章的第一篇，DCM能够应用在模型飞机以及直升机上。事实上，这篇文章到目前还只是一个草案，依然还有很多工作要做。许多评论者对于本篇文章的增改提出了很棒的建议，尤其是Louis LeGrand 和一些匿名来信的人，我们应该制作并准备一些文章中还没有包含的图表。我们后面会采纳他们的建议来进行改善，但是这可能还需要花上很长一段时间。在此期间，我们认为我们迄今得到的成果依旧能令很多读者受益。

研究DCM的动机是想要将一个带有升降舵和方向舵控制的固有稳定的飞机在稳定和控制功能上更进一步，以使一个带有副翼和升降舵的特技飞机得到稳定控制。本文的其中一个作者（Premerlan）几年前制作了一个适用于两轴的飞控板，并为一个Gentle Lady滑翔机开发了初步的固件来提供稳定性以及返航功能。固件运行得很好，作者也慢慢的信任其返航功能了，但是这个功能似乎总不像作者想要的那么好使。尤其是下面两个问题始终找不到满意的解决方案：

- 混合控制。意识到当飞机倾斜转弯的时候，由倾斜角带来了两个问题。第一，由于倾斜，飞机转弯时的偏航旋转对陀螺仪的偏航产生了一个干扰信号。第二，为了完成一个水平的转动，升降舵需要需要一些向上的偏量。这个偏量的大小取决于倾角的大小，而且这个倾角不能直接测量得到。这两个问题好比一个硬币的两面。

- 加速度。加速度计测量的是重力加速度与实际加速度相减所得到的差值。加速度的值等于作用在飞机上总的气动力（升力，推力，阻力等）加上重力 的和再除以飞机的质量最后计算所得到的结果。因此，加速度测量的是作用在飞机上的总的空气动力的负值。重力的测量是在使飞机水平的过程中所需要的，但这不是在飞机加速运动中从从加速度计中获取的值。加速度是一个混杂变量。特别是当飞机上升或者下降的时候，在很短的一段时间内，加速度计的输出没有改变，但是飞机实际上加速了。美国国家航空和航天管理局的宇航员在教练机上也遇到过类似的情况。弹道路径可以产生零净力，因此能够暂时地欺骗加速度计。将加速度问题同混合控制问题结合起来可以避免小角度的俯仰控制，而加速度问题则避免了我们在手动启动飞机的时候使用俯仰自稳。

意识到问题出现的部分原因是由于我们目前的飞控板没有一个六自由度的惯性测量单元（IMU），因此我们决定重新设计一个新的飞控板，于是乎便有了来自SparkFun的无人机开发板。

凑巧的是我们的一个研究人员（Premerlani）决定去制作一个带副翼的飞行器，但是发现自己始终掌握不好必要的飞行技能。一个夏天他就坠毁了5架飞机，其中有3次还必须将飞机整个替换掉。因此，为了更加稳定的飞行，他决定将新的飞控板安装在他的Goldberg Endurance飞机上，如下图所示。

凑巧的是我们的一个研究人员（Premerlani）决定去制作一个带副翼的飞行器，但是发现自己始终掌握不好必要的飞行技能。一个夏天他就坠毁了5架飞机，其中有3次还必须将飞机整个替换掉。因此，为了更加稳定的飞行，他决定将新的飞控板安装在他的Goldberg Endurance飞机上，如下图所示。

接下的问题是，怎样才能使其发挥出最好的效果？在我们的共同努力下，得到了与Mahony[1]相同的结论。由于我们所需要的是一个“完全遵守旋转群非线性原理”的方法。因此我和Paul觉得我们应该用一个方向余弦矩阵来表示旋转，可以用陀螺仪，加速度计以及GPS的信息来计算矩阵的元素，然后用来完成控制和导航。如果大家理解得足够深入，以下便是DCM的工作原理：

1. 陀螺仪是飞行器方位信息的主要来源。通过对飞行器的非线性微分动力学方程进行整合，将飞行器的方位对旋转速率的时间变化率以及飞行器当前方位联系起来。这个操作是在一个较高的频率下进行的，（40到50赫兹）常常足以为电机提供每一个输送过来的PWM脉冲的最新信息。

2. 意识到积分产生的数值误差会逐渐破坏DCM所必须满足的正交约束，我们对矩阵的元素做了一些常规的，小的调整，以使其重新满足约束。

3. 意识到数值误差，陀螺仪漂移，以及陀螺偏移会逐渐累积DCM元素的误差，我们使用参考向量来检测这些误差，在检测到的误差和第1步中陀螺仪的输入之间加了一个PI负反馈调节器，这样可以在误差增加之前更快的将其消除。在此过程中，GPS用于检测偏航误差，加速度用于检测俯仰和横滚的误差。

整个进程的示意图如图1所示。

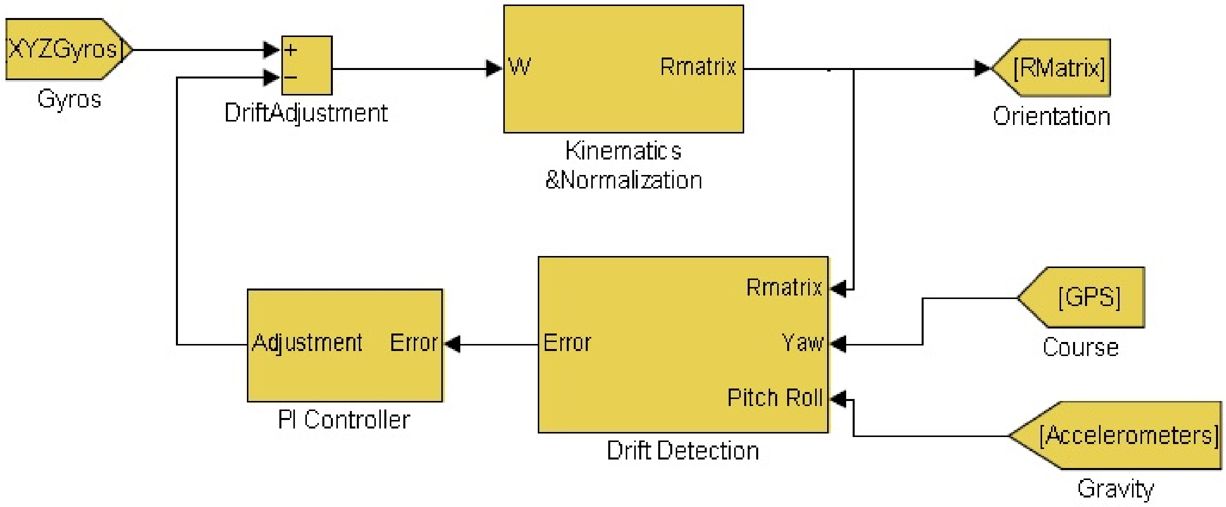


图1 DCM控制框图

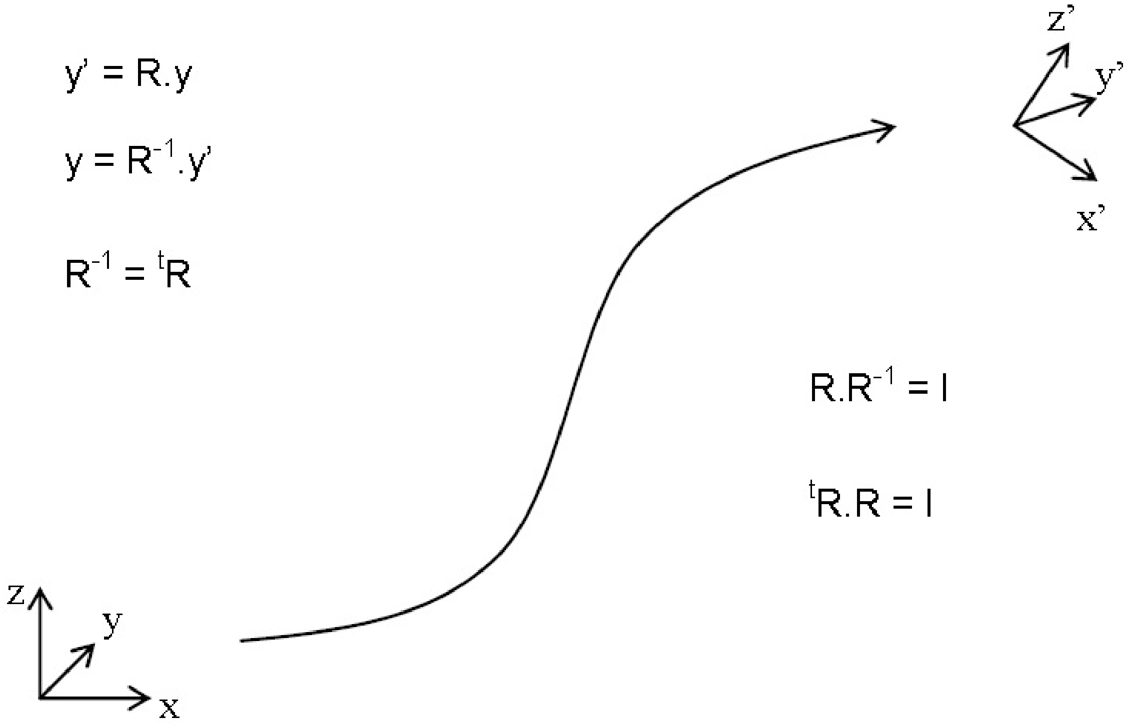
毫无疑问，你们应该都想知道什么是旋转群，为什么要使用它。你可能也想知道该如何去使用DCM进行控制和导航。当你也看过Mahony的论文后，你可能跟一些匿名来信的人有着相同的困惑，所以我们将从下面这些问题开始：

- 四元数是什么以及我们为什么用四元数而不是向量符号表示？
- 旋转群的含义是什么？
- 旋转矩阵是什么？
- 保持旋转矩阵的正交性有什么作用？
- 反对称矩阵是什么？
- 你能简要说明旋转矩阵中的运动学原理吗？
- 你能简要说明旋转矩阵中的动力学原理吗？

所有这一切都与旋转有关。事实上，我们要做的就是将飞机相对于地球的方位表示成一个旋转。有许多方法可以做到这一点。Mahony的论文讨论了两种不同的方法，旋转矩阵还有四元数。这两种表示方法的动机非常相似，它们都能够在不加近似值以及不产生奇点的情况下表示旋转。四元数的优势是其只需要四个变量就能表示一个旋转，而旋转矩阵需要9个。旋转矩阵则有一个天然的适合控制和导航的优势。这里我们选择用旋转矩阵来表示旋转是因为它有着微弱的优势，而且其表示方法与人们习惯的方式相似。

$$R = \begin{array}{cc} \begin{array}{ccc} xb & yb & zb \end{array} & \\ \left| \begin{array}{ccc} \quad & \quad & \quad \\ & & \\ & & \end{array} \right| & \begin{array}{c} xe \\ ye \\ ze \end{array} \end{array} \qquad R^{-1} = \begin{array}{cc} \begin{array}{ccc} xe & ye & ze \end{array} & \\ \left| \begin{array}{ccc} \quad & \quad & \quad \\ & & \\ & & \end{array} \right| & \begin{array}{c} xb \\ yb \\ zb \end{array} \end{array}$$

旋转矩阵描述的是一个坐标系相对于另一个坐标系的方位。旋转矩阵的列是其中一个坐标系的单位向量在另一个坐标系的中的投影。一个坐标系中的向量可以通过左乘一个旋转矩阵转移到另一个坐标系。相反的过程可以通过将一个向量左乘这个旋转矩阵的逆矩阵实现，已证明旋转矩阵的逆矩阵与旋转矩阵的转置矩阵是相等的。（矩阵的转置就是将矩阵的行与列交换。）单位向量在控制和导航中非常有用，因为其模长固定等于1。因此通常将它们用来作点乘（·）和叉乘（×）以得到各个角的正弦值或者余弦值。



一般使用平动（重心的移动）和转动（围绕重心的朝向改变）来描述飞机的运动。定义飞机水平放置，机头指向规定方向为飞机的基准朝向，从基准朝向开始，绕特定轴转动飞机，将其转动到实际朝向。依照此方法，飞机的任意朝向都可以用从基准朝向的转动来描述。

旋转群是由所有可能旋转的集合以及定义在其上的复合操作构成的群。它之所以构成群是因为集合中的任意两个旋转复合而成的旋转仍属于这个集合，每个旋转都存在逆旋转，并且存在单位旋转，这是旋转群的定义。然而，我们更愿意用这种方式来理解这个群：旋转一整圈还将回到起始位置，即旋转群是封闭的。

旋转群之所以被重视是因为：利用旋转群，在包括上下翻转和垂直向上等的任意朝向下，都可以使用最少的近似去实施控制和导航。也就是说，可以不采用任何近似去做特技飞行。

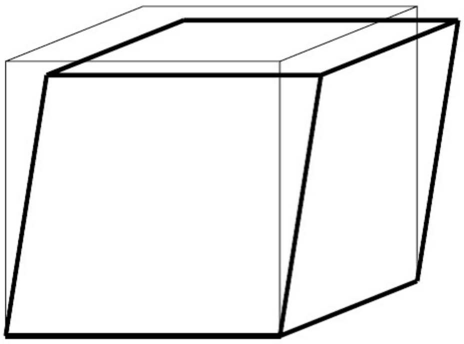
最基本的思路是：定义飞机朝向的旋转矩阵可以由描述旋转运动学的非线性微分方程积分得到（我们将在随后介绍这个非线性微分方程并解释它为什么是非线性的）。运动学关注刚体旋转的几何关系以及旋转矩阵如何对刚体施加旋转操作。通常认为总的旋转操作由一系列旋转矩阵复合而成。

旋转矩阵复合是指两个旋转矩阵相乘，根据相乘顺序依次施加两个旋转矩阵所代表的旋转，所得到的最终旋转效果与这两个矩阵相乘所得到的矩阵所代表的旋转是一致的。

然而，数值积分会引入数值误差，无法得到与符号积分相同的结果。由准确的陀螺信号进行准确的符号积分将会得到准确的、正确的旋转矩阵。即使采用准确的陀螺信号，数值积分仍会引入两类数值误差：

1. 积分误差。数值积分采用有限时间步长和具有有限采样率的数据。根据所使用的数值积分方法，对采样的数据做特定的假设。我们所使用的方法假设在每个时间步长内旋转速度恒定不变。这将引入正比于旋转加速度的误差。
2. 量化误差。无论使用哪种方法表示量值，这些表达都是有限的，所以会存在量化误差。从模数转换开始，到执行任何无法保留计算结果所有位数的计算，量化误差都将不断累积。

旋转矩阵的一个基本特征是它的正交性，即如果两个向量在某个参考系中是垂直的，那么它们在任意的参考系中都是垂直的。同样，不同参考系下的同一向量长度相同。然而数值误差会破坏这一特征，例如，旋转矩阵的行列都是单位向量，它们的长度应该等于1，但是数值误差会导致它们的长度变小或者变大。不断累积下去，它们的长度将会变为0或者无穷大。另外，旋转矩阵的行列应该是相互垂直的，数值误差会导致它们相互倾斜，如下图所示：



旋转矩阵有9个元素，但实际上其中只有3个是独立的。在数学意义上，旋转矩阵的正交特征意味着它的任意一对行（列）是垂直的，并且每行（列）元素的平方和等于1。所以，9个元素之间存在6个约束关系。

$$||xb|| = 1$$

$$||yb|| = 1$$

$$||zb|| = 1$$

$$xb \perp yb$$

$$xb \perp zb$$

$$yb \perp zb$$

$$R = \begin{array}{cc} \begin{array}{ccc} xb & yb & zb \end{array} & \\ \left| \begin{array}{ccc} \quad & \quad & \quad \\ \quad & \quad & \quad \\ \quad & \quad & \quad \end{array} \right| & \begin{array}{c} xe \\ ye \\ ze \end{array} \end{array}$$

如果一个矩阵中的每一个元素都等于它的对角元素（行列标号互换）的相反数，则称该矩阵为反对称矩阵。例如，对于一个反对称矩阵，如果它的第一行第三列的元素是0.5，那么它的第三行第一列的元素就必须是-0.5。因此，反对称矩阵的对角元素必须为0。

可以证明，一个小的旋转可以用如下的反对称矩阵描述：

$$\begin{vmatrix} 0 & a & b \\ -a & 0 & c \\ -b & -c & 0 \end{vmatrix}$$

本文研究的运动学仅涉及刚体的转动。由此导出描述飞机机体朝向与其旋转速度关系的非线性微分方程。这种运动学问题可以用方向余弦矩阵来解决。

本文研究的动力学是应用牛顿定理描述飞机机体旋转速度随时间的变化率与作用在机体上的力矩的关系。

另外，Mahony文章中的动力学对飞机而言是不准确的，它们主要涉及直升机以及垂直起降飞行器。Mahony的文章给出了一种包括朝向测量和控制的算法的应用，而Paul和我所做的只涉及运动学。我们暂时完全忽略动力学。运动学（旋转矩阵）本身非常有用，它是模型飞机导航和控制的基础。

你可能仍然想知道如何使用DCM。在笛卡尔坐标系中，使用向量叉乘和点乘运算可以完全由DCM实现控制和导航。例如，以下是五种这些导航和控制算法的应用情形：

1. 控制飞行器的俯仰，需要知道飞行器的俯仰角，而俯仰角可以由飞行器滚转轴和地垂线的点积得到。
2. 控制飞行器的滚转，需要知道飞行器的滚转角，而滚转角可以由飞行器俯仰轴和地垂线的点积得到。
3. 导航需要知道飞行器相对于目标方向的偏航角，而偏航角可以由飞行器滚转轴与目标方向的方向向量做叉乘得到。这种算法同样适用于上下翻转情况。为了判断飞行器飞行方向是否与目标方向相反，可以对滚转轴和目标方向向量做点乘，如果结果为负，则飞行器相对于目标航线偏航角度超过了90度。
4. 为了判断飞行器是否上下翻转，可以对飞行器滚转轴和地垂线做点乘，如果结果为负，则飞行器上下翻转。
5. 为了得到飞行器绕地垂线的旋转速度，可以将陀螺旋转向量变换到地球参考系，并取其与地垂线的点积。

现在我们将深入讨论DCM理论的细节。

轴公约

为了描述飞行器的运动，需要定义一个合适的坐标系统。对于大多数处理飞机运动的问题，采用了双坐标系。一种是基于地面的坐标系统，其目的是为了分析飞行器运动而建立的惯性系统。另一种是基于飞行器本身的，称为机体坐标系统。Figure 2显示的是这两种右手坐标系统。

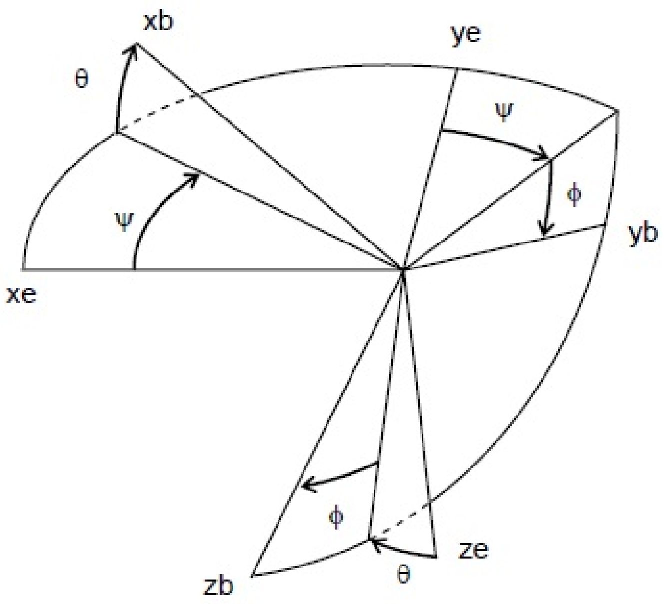


Figure 2 Body fixed frame and earth fixed frame

飞机的方向经常被描述为三个连续的旋转，其顺序是很重要的。该旋转角被称为欧拉角。机体坐标系相对于地理坐标系的方位可以通过接下来的操作确定。想象一下飞机的位置，使机体坐标系与固定机架平行，然后做以下旋转：

1. 绕zb轴旋转机体得到偏航角\(\psi\)
2. 绕yb轴旋转机体得到俯仰角\(\theta\)
3. 绕xb轴旋转机体得到横滚角\(\Phi\)

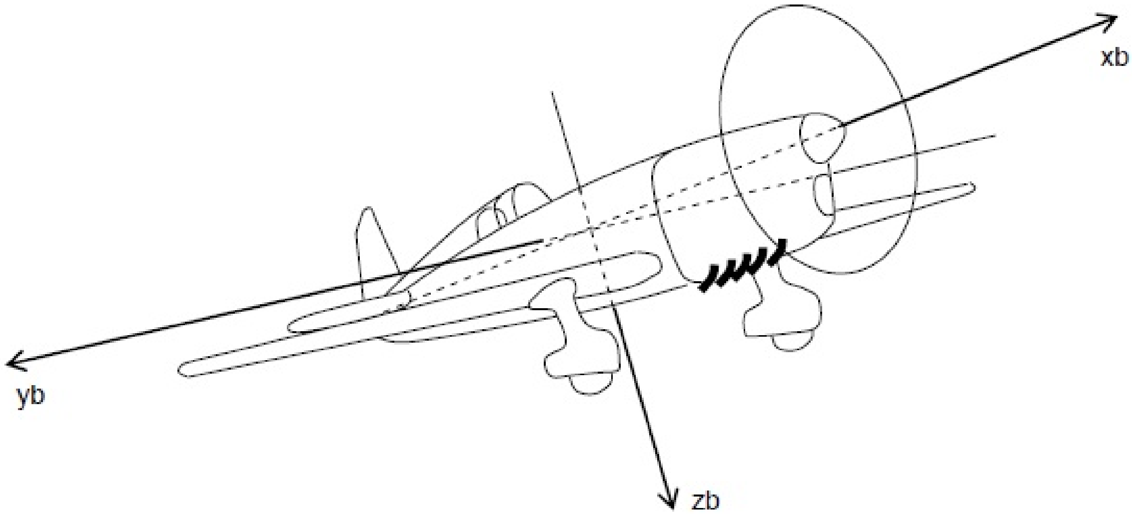


Figure 3 Body axes coordinate system

方向余弦矩阵

某些类型的向量，比如：方向，速度，加速度，位移，（运动）可以通过旋转参考系和3*3的矩阵来进行转换。通过机体坐标系和地球坐标系，在方向余弦矩阵里对旋转向量进行相乘从而将这些向量进行转换。

$$\mathbf{Q} = \begin{bmatrix} Q_x \\ Q_y \\ Q_z \end{bmatrix} = \text{a vector, such as a direction, velocity or acceleration}$$

$$\mathbf{Q_P} = \text{a vector } \mathbf{Q} \text{ measured in the frame of reference of the plane}$$

$$\mathbf{Q_G} = \text{a vector } \mathbf{Q} \text{ measured in the frame of reference of the ground} \qquad \text{Eqn. 1}$$

$$\mathbf{R} = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{bmatrix} = \text{rotation matrix}$$

$$\mathbf{Q_G} = \mathbf{R}\mathbf{Q_P}$$

方向余弦矩阵和欧拉角之间的关系是：

$$\mathbf{R} = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \qquad \text{Eqn. 2}$$

方程1和方程2表达了怎样把机体参考系里的标准向量旋转（转换？）到地球参考系。方程1可由方向余弦确定，方程2可由欧拉角确定。

在方程1中，地球坐标系中的每个向量的分量与相应的旋转向量的行和机体坐标系的向量的点积相等。计算旋转需要九次相乘六次相加。方程3通过对向量和矩阵的扩展相乘来对方程1进行复述。

$$\begin{aligned} Q_{Gx} &= r_{xx}Q_{Px} + r_{xy}Q_{Py} + r_{xz}Q_{Pz} \\ Q_{Gy} &= r_{yx}Q_{Px} + r_{yy}Q_{Py} + r_{yz}Q_{Pz} \\ Q_{Gz} &= r_{zx}Q_{Px} + r_{zy}Q_{Py} + r_{zz}Q_{Pz} \end{aligned} \qquad \text{Eqn. 3}$$

注意矩阵R并不一定是对称的。矩阵R的三列是用来将机体的三个向量转换到地球参考系。矩阵R的三行是用来将地球坐标系的三个向量转换到机体参考系。矩阵R包含的所有信息都需要表达出机体相对于地面的方向。矩阵R也叫作方向余弦矩阵，因为每一个条目都是机体轴和地

球坐标系的轴的夹角的余弦。虽然九个独立参数会在矩阵R中出现，实际上只有三个参数是真正独立的，因为六个所谓正交（也称规范化）的条件是：三个纵列上的向量是互相垂直的，而三个横向的向量等于1。

任何矩阵，特别是旋转矩阵的变换，都可表示为RT，通过互换行和列来完成。通常情况下，如果存在，一个方矩阵的逆可以表示为R-1。矩阵的逆与矩阵相乘等于单位矩阵。单位矩阵的对角线上的值都是1，其他全部为0。任意矩阵乘以单位矩阵都保持不变。在旋转矩阵情况下，对R矩阵的转换等于它的逆矩阵。

$$\mathbf{R}^{-1} = \mathbf{R}^{\mathrm{T}} = \begin{bmatrix} r_{xx} & r_{yx} & r_{zx} \\ r_{xy} & r_{yy} & r_{zy} \\ r_{xz} & r_{yz} & r_{zz} \end{bmatrix} \qquad \text{Eqn. 4}$$

$$\mathbf{Q}_{\mathrm{P}} = \mathbf{R}^{-1}\mathbf{Q}_{\mathrm{G}} = \mathbf{R}^{\mathrm{T}}\mathbf{Q}_{\mathrm{G}}$$

旋转矩阵的逆矩阵和它的转置之后相等是因为其结构对称。旋转矩阵的元素是机体参考系和地球参考系各轴夹角的余弦。矩阵的逆变换就相当于互换了地球参考系与机体参考系的角色，与交换矩阵的行和列是一样的，与对矩阵进行转置也是一样的。

不仅如此，事实上逆和转置相等是和正交性条件一致的，它可以用矩阵符号表示如下：

$$\mathbf{R}\mathbf{R}^{\mathrm{T}} = \mathbf{R}^{\mathrm{T}}\mathbf{R} = \mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{Eqn. 5}$$

方程5可以用来证明R矩阵的逆就是R矩阵的转置，通过方式与R的逆矩阵相乘，或者R转置之后的逆。

旋转矩阵非常有用的一个特点就是：我们可以调整旋转。我们可以将多个旋转矩阵一起相乘，并计算出一个旋转矩阵，相当于把许多的旋转连续起来。在我们已经有的左侧，我们必须小心应用连续旋转。比如我们有三个旋转矩阵：从方位A到方位B，从B到C，从C到D，我们可以通过下面的方程计算出从A到D的旋转矩阵

$$\begin{aligned} \mathbf{R}_{\mathbf{DA}} &= \mathbf{R}_{\mathbf{DC}}\mathbf{R}_{\mathbf{CB}}\mathbf{R}_{\mathbf{BA}} \\ \mathbf{R}_{\mathbf{BA}} &= \text{rotation matrix from A to B} \\ \mathbf{R}_{\mathbf{CB}} &= \text{rotation matrix from B to C} \\ \mathbf{R}_{\mathbf{DC}} &= \text{rotation matrix from C to D} \\ \mathbf{R}_{\mathbf{DA}} &= \text{rotation matrix from A to D} \end{aligned} \qquad \text{Eqn. 6}$$

而当对旋转矩阵进行相乘时，我们需要对这一系列操作非常小心的原因是：矩阵的相乘时不可互换的（互换位置）。也就是说矩阵相乘的顺序异常重要。对于旋转也是这样，不可互换。例如，当机体的俯仰和滚转角都为90度时。顺序非常重要。假设俯仰到90度，接着横滚也为90度，机体将与地面垂直（头朝上垂直）。反之，如果先横滚90度，后俯仰90度，机体为侧躺在地面之上。

最后，对于一般矩阵，特别是旋转矩阵，还有一个非常有用的特性：两个矩阵乘积的转置等于两个矩阵转置的乘积（对A和B转置之后的位置互换）

$$(\mathbf{AB})^{\mathrm{T}} = \mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}$$
$$\mathbf{A},\mathbf{B}\ \text{are matrices} \qquad \text{Eqn. 7}$$

向量点叉乘

在计算DCM、使用它的元素进行导航和控制的时候，我们将用到两个非常有用的向量乘法：点乘和叉乘。两个向量A和B的点乘结果是一个标量，通过执行矩阵的乘法——A作为行向量、B作为列向量而被计算出来：

$$\mathbf{A}\cdot\mathbf{B}=\mathbf{A}^{\mathrm{T}}\mathbf{B}=\begin{bmatrix}A_x & A_y & A_z\end{bmatrix}\begin{bmatrix}B_x\\ B_y\\ B_z\end{bmatrix}=A_xB_x+A_yB_y+A_zB_z \qquad \text{Eqn. 8}$$

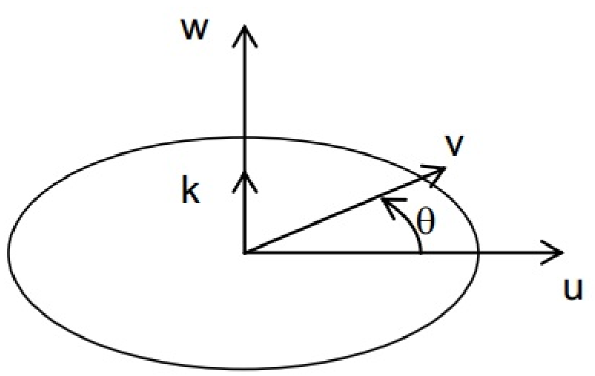
其实两个向量的点乘产生的结果等于两向量模的乘积，乘以两向量夹角的余弦值：

$$\mathbf{A}\cdot\mathbf{B}=|\mathbf{A}||\mathbf{B}|\cdot\cos(\theta_{AB}) \qquad \text{Eqn. 9}$$

注意点乘是可交换的：**A**\(\cdot\)**B**=**B**\(\cdot\)**A**
两个向量A、B的叉乘结果是一个向量，这个向量的各个组成部分由下面公式计算出来：

$$\begin{aligned}(\mathbf{A}\times\mathbf{B})_x&=A_yB_z-A_zB_y\\ (\mathbf{A}\times\mathbf{B})_y&=A_zB_x-A_xB_z\\ (\mathbf{A}\times\mathbf{B})_z&=A_xB_y-A_yB_x\end{aligned} \qquad \text{Eqn. 10}$$

这个向量垂直于A、B向量，它的模等于两向量模的乘积，乘以夹角的正弦值：



$w = u \wedge v = ||u||.||v||.\sin(\theta).k$

$||w|| = ||u||.||v||.\sin(\theta)$

k : unit vector orthogonal to the plane defined by u and v

另一方面：

$$
\begin{aligned}
(\mathbf{A}\times\mathbf{B})\cdot\mathbf{A} &= (\mathbf{A}\times\mathbf{B})\cdot\mathbf{B} = 0 \\
|\mathbf{A}\times\mathbf{B}| &= |\mathbf{A}||\mathbf{B}|\sin(\theta_{AB})
\end{aligned}
\qquad \text{Eqn. 11}
$$

注意叉乘是反交换的：**A**\(\times\)**B**=-**B**\(\times\)**A**

陀螺仪信号计算方向余弦

在前文基础上，我们现在关注DCM算法的核心概念：描述陀螺信号和方向余弦矩阵时间变化率之间关系的非线性微分方程。我们的目的是，在不采用任何可能破坏方程非线性的近似条件的前提下来计算方向余弦矩阵。为方便讨论，暂时假设陀螺信号没有任何误差，随后我们将会专门讨论陀螺误差。

与物理陀螺固定在惯性空间而飞行器围绕它旋转不同，电子陀螺随着飞行器一起旋转，产生正比于旋转角速度的电信号。因为旋转的不可交换，所以旋转顺序变得尤为重要，我们不能简单地积分角速度信息来得到角度信息。我们所要做的是从旋转的运动学方程入手，考虑需要做什么才能得到正确的角度信息。

众所周知，旋转运动学方程（由旋转带来的向量的变化率）可以表示为：

$$\frac{d\mathbf{r}(t)}{dt} = \boldsymbol{\omega}(t) \times \mathbf{r}(t)$$
$$\boldsymbol{\omega}(t) = \text{rotation rate vector} \qquad \text{Eqn. 12}$$

观察上式，可以看到：

1. 微分方程是非线性的。输入的角速度向量和我们想要积分得到的方向向量叉乘。因此，任何线性的方法都只是近似结果。
2. 式中所有的向量都在同一参考系下测量。
3. 因为叉乘是不可交换的，所以我们在交换顺序的同时需要改变符号才能保证结果不变。

如果我们知道初始条件和每一时刻的旋转角速度，那么我们可以通过对式(11)进行数值积分得到每一时刻向量的方向。

$$\mathbf{r}(t) = \mathbf{r}(0) + \int_0^t d\boldsymbol{\theta}(\tau) \times \mathbf{r}(\tau)$$
$$d\boldsymbol{\theta}(\tau) = \boldsymbol{\omega}(\tau) d\tau$$
$$\mathbf{r}(0) = \text{starting value of the vector} \qquad \text{Eqn. 13}$$
$$\int_0^t d\boldsymbol{\theta}(\tau) \times \mathbf{r}(\tau) = \text{change in the vector}$$

我们的策略是将R矩阵中的每行或每列视为一个方向向量，采用式(13)积分它们。

采用上述方法所遇到的第一个障碍是：我们想要积分得到的旋转向量和旋转角速度向量不在同一个参考系下测量。理论上，我们想要知道机体参考系在地球参考系下的方向，但是陀螺测量的是机体参考系下的角速度。有一个简单的解决方法是考虑旋转的对称性，即从机体参考系看地球参考系的旋转和从地球参考系看机体参考系的旋转是一致的，只不过方向相反。

所以我们可以转而讨论地球参考系在机体参考系下的朝向，这样做仅仅需要将陀螺信号反向即可，但是出于方便考虑，我们利用叉乘的特性，保持陀螺信号符号不变同时改变叉乘顺序。

$$\begin{aligned}&\mathbf{r}_{earth}(t)=\mathbf{r}_{earth}(0)+\int_0^t \mathbf{r}_{earth}(\tau)\times d\boldsymbol{\theta}(\tau)\\&d\boldsymbol{\theta}(\tau)=\boldsymbol{\omega}(\tau)d\tau\\&\mathbf{r}_{earth}(t)=\text{one of the earth axes, as viewed from the plane}\end{aligned} \qquad \text{Eqn. 14}$$

式(14)中的向量即为式(1)中R矩阵的行向量。接下来的问题是如何方便地应用式(14)。我们使用与Mahony相同的矩阵方法，从式(14)所表达的差分形式着手：

$$\begin{aligned}&\mathbf{r}_{earth}(t+dt)=\mathbf{r}_{earth}(t)+\mathbf{r}_{earth}(t)\times d\boldsymbol{\theta}(t)\\&d\boldsymbol{\theta}(t)=\boldsymbol{\omega}(t)dt\end{aligned} \qquad \text{Eqn. 15}$$

在此之前还需要做的一件事是陀螺漂移的消除，我们需要在陀螺测量角速度的基础上加上角速度修正量来得到我们对角速度的最好的估计。角速度修正量由陀螺漂移反馈补偿器得到，具体的内容将在之后的章节讨论。基本的思路是通过GPS以及加速度参考向量计算旋转误差，然后将之送入反馈补偿器，最后将补偿器输出通过式(16)送回旋转更新方程。

$$\begin{aligned}&\boldsymbol{\omega}(t)=\boldsymbol{\omega}_{\mathbf{gyro}}(t)+\boldsymbol{\omega}_{\mathbf{correction}}(t)\\&\boldsymbol{\omega}_{\mathbf{gyro}}(t)=\text{three axis gyro measurements}\\&\boldsymbol{\omega}_{\mathbf{correction}}(t)=\text{gyro correction}\end{aligned} \qquad \text{Eqn. 16}$$

对地球参考系的三个轴做式(14)的同样操作，可以将其写成更为方便的矩阵形式：

$$\boxed{\begin{aligned}&\mathbf{R}(t+dt)=\mathbf{R}(t)\begin{bmatrix}1 & -d\theta_z & d\theta_y\\ d\theta_z & 1 & -d\theta_x\\ -d\theta_y & d\theta_x & 1\end{bmatrix}\\&d\theta_x=\omega_x dt\\&d\theta_y=\omega_y dt\\&d\theta_z=\omega_z dt\end{aligned}} \qquad \text{Eqn. 17}$$

式(17)描述了由陀螺信号更新方向余弦矩阵的方法，这与Mahony的结果是相同的。式(17)中的矩阵对角线元素1代表式(15)中的第一项；非对角元素中的那些小量代表式(15)中的第二项。在很短的时间步长内通过不断地矩阵相乘式(17)，每次矩阵乘法需要进行27次乘法和18次加法，对于具有硬件资源来更有效率地进行矩阵乘法的dsPIC30F4011，这种算法表现地很出色。当然，这种算法也可以在不支持矩阵运算的CPU上执行，在这种情况下，推荐使用整数算法。

式(17)采用的唯一近似要求时间步长需要足够的短，这样R矩阵在两步之间不会变化的特别多。一个典型的时间步长是0.02s，角速度大约60度/秒的飞行器在每个步长内大概旋转了0.02弧度，此种情况下R矩阵中元素最大变化率不会超过2%。因此，被忽略的二阶项大概是0.02%。

测试和仿真结果表明，仅仅是应用式(15)，不需要特别好的陀螺，就能得到非常精确的结果：漂移量非常小，大概每分钟几度。对于非常小的漂移，调整它而不损失性能是很容易做到的。然而，式(15)自身运算将会累计数值截断误差，陀螺漂移误差，陀螺偏置误差以及增益误差。在接下来的两章我们将会讨论如何消除这些误差。

重规范化

由于数值误差的存在，方向余弦矩阵不再满足正交性，即式(5)右端不再严格等于单位矩阵。事实上，此时坐标系不再描述一个刚体。幸运的是，数值误差积累地非常缓慢，所以及时修正误差不是一件困难的事情。

我们把使方向余弦矩阵满足正交性的操作称为重规范化。有很多种方法可以实现重规范化操作。仿眞结果显示它们的效果都比较好，这里给出最简单的一种方法，操作流程如下：

首先计算方向余弦矩阵X轴与Y轴的内积，如果矩阵严格正交，那么这个结果应该是0，所以这个结果实际上反映了X轴与Y轴相互旋转靠近的程度。

$$\mathbf{X}=\begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix} \qquad \mathbf{Y}=\begin{bmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{bmatrix}$$

$$error=\mathbf{X}\cdot\mathbf{Y}=\mathbf{X}^{\mathrm{T}}\mathbf{Y}=\begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \end{bmatrix}\begin{bmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{bmatrix} \qquad \text{Eqn. 18}$$

把误差均分给X轴与Y轴，并近似地将X轴与Y轴分别向相反的方向转动，在此X轴与Y轴互相修正，具体操作如下：

$$\begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix}_{orthogonal}=\mathbf{X}_{orthogonal}=\mathbf{X}-\frac{error}{2}\mathbf{Y}$$

$$\begin{bmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{bmatrix}_{orthogonal}=\mathbf{Y}_{orthogonal}=\mathbf{Y}-\frac{error}{2}\mathbf{X} \qquad \text{Eqn. 19}$$

可以验证，将式(19)代入式(18)后，正交性误差逐渐减小。验证过程中可能需要用到方向余弦矩阵行列均为单位向量这一条件。相比于将误差完全分给X轴或Y轴其中一个，将误差均为能够在修正后得到更小的误差。

下一步是调整方向余弦矩阵的Z轴，使得它与X轴和Y轴均正交。这里我们采用的方法是简单地重新计算Z轴，使其等于X轴与Y轴的外积。

$$\begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix}_{orthogonal} = \mathbf{Z}_{orthogonal} = \mathbf{X}_{orthogonal} \times \mathbf{Y}_{orthogonal} \qquad \text{Eqn. 20}$$

最后一步是调整方向余弦矩阵每一行的大小使其等于1。一种方法是对每一行的每个元素，除以该行元素平方和的二次方根。然而，有一种更容易的方法是，鉴于每一行的大小不会与1相差太远，可以采用泰勒展开。计算方法如下所示：

$$\begin{aligned} \mathbf{X}_{normalized} &= \frac{1}{2}\left(3 - \mathbf{X}_{orthogonal} \cdot \mathbf{X}_{orthogonal}\right)\mathbf{X}_{orthogonal} \\ \mathbf{Y}_{normalized} &= \frac{1}{2}\left(3 - \mathbf{Y}_{orthogonal} \cdot \mathbf{Y}_{orthogonal}\right)\mathbf{Y}_{orthogonal} \\ \mathbf{Z}_{normalized} &= \frac{1}{2}\left(3 - \mathbf{Z}_{orthogonal} \cdot \mathbf{Z}_{orthogonal}\right)\mathbf{Z}_{orthogonal} \end{aligned} \qquad \text{Eqn. 21}$$

式(21)所描述的是将每行向量归一化的操作：用3减去该行向量与自身的内积（该行元素平方和），然后乘以1/2，最后将结果乘到该行所有元素上。

采用此种方法进行归一化的好处是，既没有采用更多的乘法和加法，同时也完全移除了除法与平方根运算。在每一步积分都执行上述重规范化操作，执行周期为0.02s。

漂移消除

虽然陀螺表现相当好，在指令下面每秒的误差很小，但最终对陀螺仪的漂移我们不得不做一些事情。所要做的就是利用其它方向的参考来探测陀螺偏移，提供一个经典的PID负反馈检测回路给陀螺补偿误差。如图1所示。步骤如下：

1. 使用方向参考向量探测定向误差，通过计算一个旋转矢量，将测量值和计算值的参考矢量调整。
2. 通过一个比例积分（PI）反馈控制器来产生的陀螺旋转校准速度，将旋转矢量误差反馈。（PI调节器是常用的PID反馈调节器的一个特例，D代表微分。在我们的这个例子中，我们不需要用到微分项。）
3. 加上（或者减去，这依赖于你对旋转误差的符号约定）比例积分控制器的输出到实际的陀螺仪信号。

对方向参考向量的主要要求是：它不漂移。其瞬态性能并不那么重要是因为陀螺对方向估计有瞬态保真性。

GPS和加速度计为我们提供了两个参考向量提供。磁力计也是非常有用的，特别对偏航的控制，但是对飞行器飞行的指向，仅一个全球定位系统就做得很好。如果你使用磁力计，你应该使用一个三轴磁力计提供一个矢量参考。 低成本三轴磁力计在市场上可以很方便的买到。

我们使用加速度计为飞机的Z轴提供的参考矢量飞。将在一个单独的部分中给出细节。我们使用GPS作为飞机X-轴（滚转轴）的水平投影参考。我们的2个参考向量恰好是彼此垂直的。这很方便，但不是绝对必需的。

对于两个参考向量中的任一个，通过测得的向量的交叉乘积检测方向误差，这个测得的矢量用方向余弦矩阵估计。这个叉积特别合适有两个理由。它的大小与两个矢量夹角是正弦成正比，同时它的方向垂直这两个矢量。所以它代表一个旋转轴和旋转量，需要旋转这个测量的矢量使它变得与这个估计的矢量平行。换句话说，它等于方向旋转误差的负值。通过比例积分控制器把它反馈到陀螺仪，定位估计逐渐被迫跟踪参考向量，这样陀螺的漂移就被消除了。

测量从方向余弦矩阵计算出相应的参考矢量的叉积是一个错误的指示。它大约等于这个旋转，将不得不用这个参考矢量和这个计算矢量校准。我们感兴趣的是旋转校正量，这个量我们需要用在方向余弦矩阵中，它等于旋转误差的负值。通过交换叉积的秩序可以很方便的计算修正。矫正旋转等于通过参考矢量的方向余弦的矢量估计的叉积。

我们用比例积分控制器来矫正陀螺旋转，因为它稳定，同时这个积分项完全消除了陀螺仪的漂移，包括热漂移，零残余方向误差。

参考矢量误差是通过方向余弦矩阵的方式映射到陀螺的。所以这样的映射取决于惯性测量单元。例如，GPS参考矢量可能矫正X、Y、Z或者X、Y和Z轴陀螺信号，取决于地球坐标系轴的方向。

我们现在将获得更多关于我们所用的GPS和加速度计这两个参考的细节。

GPS

GPS提供无漂移时飞机偏航指向的参考矢量。我们不使用GPS本身来得到偏航信息的唯一理由是陀螺的瞬态响应比GPS快多了。相反，我们使用GPS作为参考矢量来消除陀螺漂移并获得航向“锁定”。

GPS无线电模块提供的两个主要服务是汇报位置和速度的大小与方向。GPS根据它从在轨卫星收到的信号决定它的位置和速度，并且通过串口发送出信息。对于大部分GPS接收机，都有两种数据格式，NMEA和二进制。NMEA是用逗号分隔、可以直接阅读的标准ASCII码格式。在二进制接口中，二进制值以其内部含义代表的二进制的1和0组成的序列发送。二进制接口提供了一些NMEA接口不能获得的额外信息。

为了给出方向信息，GPS必须要移动。否则就确定不了GPS天线的指向。GPS报告的速度矢量是天线在1秒内的位置变化量。有一些GPS或许可以用来计算方向的方法，但对于所有方法，GPS都必须要移动。

对于GPS模组有两种不同的坐标系来报告位置和速度。一个坐标系报告、经度、纬度、距地面高度偏航角（偏航角是测量出的航线偏离北方的顺时针角度）。相当有趣的是，从数学意义上，当机体固连参考系的Z轴指向正下方时，这个偏航角和绕该轴顺时针旋转测量得到的角是同一个角。在该参考系中，竖直速度可以通过二进制接口获得。

另一参考系，ECEF（(earth-centered, earth-fixed）报告XYZ方向上的位置和速度，该参考系是原点在地心的右手XYZ坐标系。 GPS以报告的连续序列发送其信息，通常是每秒一次（1HZ）。虽然有提高报告频率到更为普遍的每秒5次（5HZ）的趋势，但更高的报告频率并不一定带来更好的性能，因为GPS内部信号处理的动力学产生了限制。

在考虑GPS动力学时，应该留心以下几种因素：

1. 报告延迟。在某些环境下，对于一些GPS模组，发送计算数据会花费长达12秒的时间；
2. 滤波。所有的GPS模组都进行了某类滤波来提高位置和速度的精度。当GPS改变速度或位置时，这种滤波会对位置和速度的估计产生平滑作用，以至于新的信息不会立刻被看到，但会逐渐显露出来。
3. 跟踪平滑和静态导航。很多种GPS无线电模块提供“跟踪平滑”选项以便忽略位置或速度的急速改变。为了防止卫星信号的变化被当成结果的变化，例如当正在使用的卫星集合改变了，这个功能对于汽车应用是有用的。GPS无线电模块也提供了“静态导航”选项，以便当速度落于一定值之下时，明显的位置变动被压制住。这对于汽车应用也是有用的。

你不可能每个模块都使用了跟踪平滑或静态导航功能，因为厂家会默认关闭这些选项，但你应该意识到它们。可是，报告延迟和滤波一定要考虑。

报告延迟意味着GPS测量到位置和速度的时刻和消息序列出现时刻的一个简单的时间延迟。通常这个延迟是报告的时间周期。例如，如果你的GPS是以5HZ报告，报告延迟通常是0.2秒。但是，如果你不仔细的话，延迟也会大的多。我们中的一位（Bill）曾经不幸在1Hz的报告频率下遭遇到12秒的延迟。最终证明12秒的延迟是由于同时使用了4900的波特率和二进制接口。改变波特率到19200，延迟减小到1秒。幸运的是你不会经常碰到这个效果，但要意识到它存在。如果你使用二进制接口，你应该使用19200或更高的波特率。

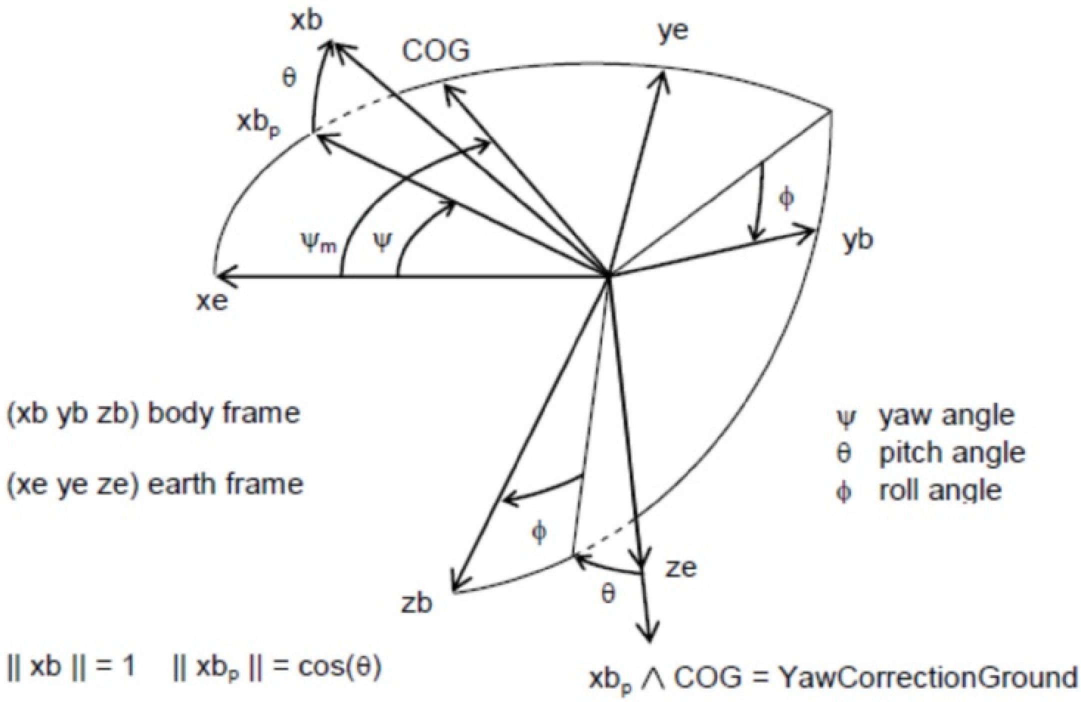
除了简单的延迟，你通常也会遇到由GPS内部滤波造成的延迟。由于GPS模组本身计算的本质所致，它们都进行了某种方式的数据滤波。对于任意系统在精度和瞬态响应之间本质上都有折中。你想知道某些事情越准确，估计它花的时间越长。在大部分模组中，滤波体现为对数据的平滑。通常，很多种GPS的动态响应是带有1秒时间常数的简单的指数响应，以至于要花费整整大约3秒来响应阶跃变化。如果你忽视了GPS的动力学，在转弯时将会有微小误差引入到你的导航解算中。我们中的一位（Paul）看到在方向余弦阵和偏航漂移修正的输入中引入滤波器有可能补偿微小误差。[我们需要一张图吗？]那样的话，两个用于估计偏航角误差的矢量动力学就匹配了。

经常假设有很高报告频率（例如5HZ）的GPS，将会比报告频率为1HZ的GPS提供更好的动态性能。但是，又没有确凿的案例证明更高的报告频率提供更好的动态响应。当然，它的延迟也会变小。但是，滤波器动力学仍旧存在问题，通常证明效果很有限。

对地航向的水平GPS分量经过长时间后会有零点漂移，而且这个水平过程可以用作参考矢量来获得对惯性测量单元的“偏航锁定”。我们也考虑包含GPS的垂直速度，但终究决定排除了它，而更倾向于加速度计来提供垂直信息。

假设飞机在指向的方向上运动。任何假设中的瞬态误差本质上都不会影响性能。但是，大风，特别是侧风会违背这个假设。你可以采用两种方法。一种方法是以某种方式从获得的信息中计算风矢量。我们一直致力于这个工作。另一种方法是使用适量的反馈增益。飞机的指向和运动方向的差别会以误差的形式输入到漂移修正反馈控制器中。结果是方向余弦阵会适应风，并把飞机旋转一定角度以保持飞机沿着地面系的目标航线运动。

下面的图片展示了偏航修正是如何计算的：



对地GPS航向矢量和IMU滚转轴（X）在水平面上的投影意味着漂移量。旋转修正是R矩阵X列和对地航向矢量叉积的Z分量。

首先，我们从规范化的水平速度矢量构建参考矢量。可以通过简单地用地球参考系的对地航向角的余弦和正弦来做

$$COGX = \cos(cog)$$
$$COGY = \sin(cog) \qquad \text{Eqn. 22}$$

接着我们计算偏航修正

$$YawCorrectionGround = r_{xx}COGY - r_{yx}COGX \qquad \text{Eqn. 23}$$

但是，方程23得出的是地球参考系中的偏航角修正。为了调整陀螺漂移，我们将需要知道机体系中的修正矢量。为了计算这个，我们必须对地面系中的偏航修正乘以R矩阵的Z行

$$\mathbf{YawCorrectionPlane} = YawCorrectionGround \begin{bmatrix} r_{zx} \\ r_{zy} \\ r_{zz} \end{bmatrix} \qquad \text{Eqn. 24}$$

方程24产生的偏航修正矢量将和从加速度计算得到的滚转-俯仰修正结合为整个的一个矢量用来补偿漂移。计算的细节将在我们讨论完加速度计如何使用后给出。

有三种相对于偏航补偿的情形支持大的偏航修正权重，使得快速响应偏航误差。

第一种情形是初始偏航锁定。当算法开始后，根本没办法知道板子的指向，即便知道，经过一段等待GPS锁定的时间，航向也一直在漂移。而且就算GPS锁定后，在飞机起飞前，GPS报告的对地航向也会是随机数。通过给偏航漂移修正一个大的权重，偏航锁定可以在起飞后立刻获得。

第二种情形是风。如果飞机在侧风中飞了很长时间，风会被当成陀螺补偿。如果飞机接着转弯180度，经过一段时间方向余弦阵算法会转动飞机到反向的角度，这个角度需要补偿风。

第三种情形是当飞机垂直运动时，飞机的X轴是垂直的，方程23为0。

出于这些原因，最好使用大的偏航漂移修正权重。

加速度计

由于水平姿态有零飘，所以采用加速度计对横滚角和俯仰角漂移进行修正。我们必须考虑离心加速度，其中能够被解释说明的（机体运动加速度）情况，将会接下来会进行讨论。

我们成员之一BILL在建立他的第一块板子的时候，他认为加速度计完全可以靠自己进行横滚俯仰控制，然而由于一系列原因并不是这样。主要是因为加速度计测量的是运动加速度和重力加速度之和。如果加速度计仅测量重力加速度，这样计算将没有任何问题；但是加速度计还测量运动加速度，所以会带来问题。BILL曾经尝试在手推出滑翔机的时候，仅基于加速度计输出稳住机体俯仰角，推出产生的运动加速度错误地估计出飞机俯仰朝上进行控制，控制反馈让机体笔直俯冲向下。

通常加速度计的工作机理就是测量弹簧拉一小质量块的形变。由于加速度计的动态固有频率较高，所以加速度计响应灵敏。弹簧形变大小取决于质量块上受到的所有力，这个力等于质量块质量乘以重力加速度矢量和运动加速度矢量之和。（常规迹象来说，加速度计测量的是重力加速度减去运动加速度）

所以除了重力加速度，加速度计还测量运动加速度，这并不惊讶，因为它就叫加速度计。 因此，加速度计只有在飞机不处于加速运动状态，才能准确估计横滚和俯仰角。问题是飞机经常加速运动，其中一些运动加速度，比如离心运动加速度，在没有飞机动态模型情况下，是很容易计算和补偿的。然而，却很难辨别计算出向前的运动加速度。

还有一线生机。一般来说，飞机不是在向前加速运动，而是时而加速时而减速的运动，因此它的运动加速度过程在相互抵消。一架飞机不可能一直加速向前运动，直到空气阻力使其速度不能更快；一架飞机也不可能一直减速运动，不然停止将从天上掉下。由于加速度计没有漂移，所以只要我们不依赖于加速度计实现高动态响应，我们还是可以用加速度计来修正陀螺漂移带来的横滚俯仰误差。

目前市场上有很多好的加速度计，其中大部分采用DCM算法将工作良好。加速度计并不像陀螺仪这么关键，因为加速度计零偏上的任何变化不会产生误差的累积，但是陀螺仪会。因为加速度计是直接测量角度，而陀螺仪是测量角度随时间变化的速率的。

加速度计有许多种接口类型，包括模拟电压、脉宽调制以及一些标准的通信接口。我们选择了一个加速度计，采用最为简单的接口方式即模拟电压输出。

采用方向余弦方法的最大好处是，在飞机任何方向角度时都能实现，计算没有任何奇异或者特别的逻辑。飞机任何角度方向可以采用方向余弦矩阵的9个元素充分描述。 由于我们需要校正漂移，来计算飞机的任何方向角度，所以我们将需要测量沿着飞机三个轴的加速度。这个可以用市面上三轴一体的加速度计实现，或者三个独立的加速度计实现。

在我们能够使用加速度计信息来补偿横滚和俯仰角漂移之前，我们必须考虑离心加速度和飞机向前运动的速度变化量。虽然飞机向前加速或者减速运动时间比较短，但是它可以无限的转动下去。

幸运的是，所需要来计算离心加速度的信息一应俱全，离心加速度等于转动角速度和线速度的叉乘积。我们不需要一个确切的答案，一般来说也只有一个确切的答案。一般来说，飞机在指定的方向运动，因此我们假设飞机的速度矢量方向平行于飞机的X轴，GPS给我们提供飞机对地的速度大小，由于大地是一个惯性参考系，我们可以计算出飞机机体系下的速度矢量，也就是在X方向的对地速度。

在机体参考坐标系中，我们计算离心加速度为陀螺矢量和速度矢量的叉乘积：

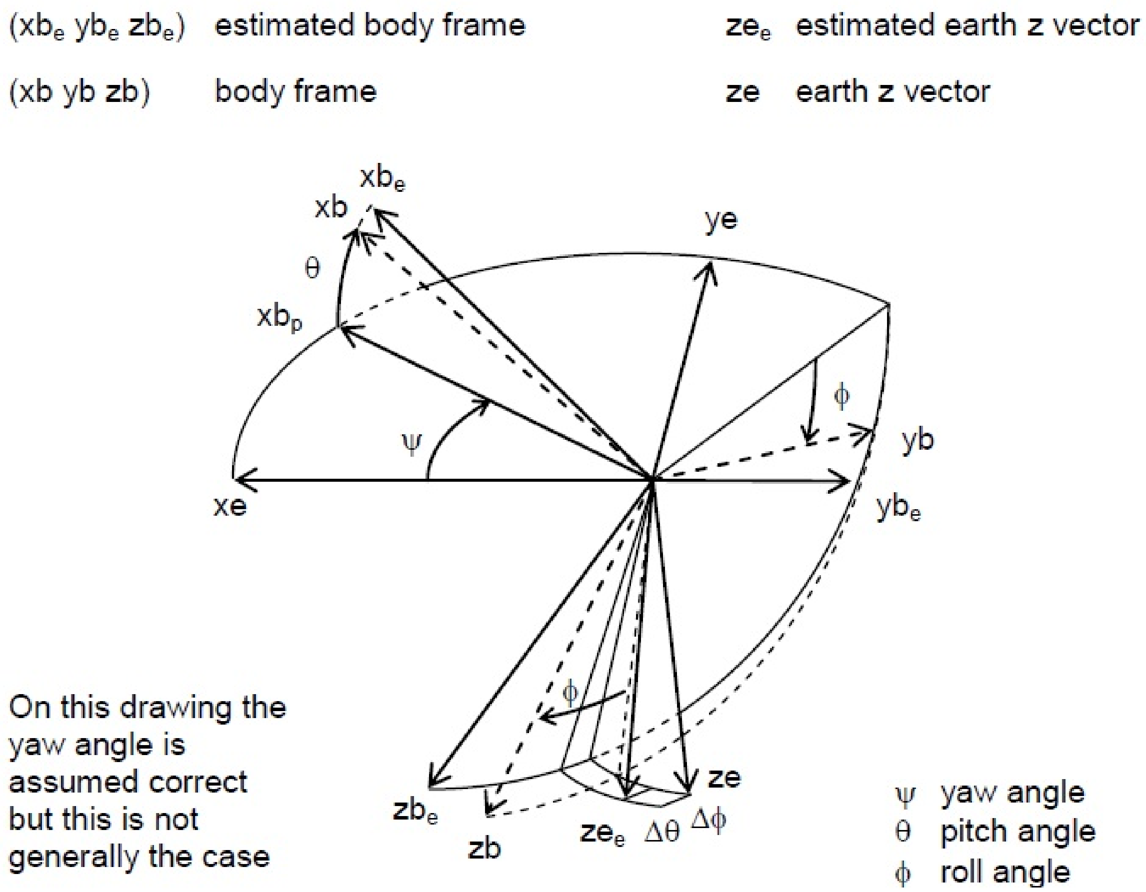
$$\mathbf{A}_{\mathbf{centrifugal}} = \boldsymbol{\omega}_{\mathbf{gyro}} \times \mathbf{V}$$
$$\mathbf{V} = \begin{bmatrix} velocity \\ 0 \\ 0 \end{bmatrix} \qquad \text{Eqn. 25}$$

注意式25，我们只需要执行两次乘法，因为机体系下速度矢量中两个元素为0.

常规地，市场上三轴加速度计中Z轴是朝下的，向下的重力加速度产生一个正输出。为了修复带有离心加速度的重力加速度值，我们需要添加离心加速度值得估计。因此，机体系下的重力矢量的参考测量值为：

$$\mathbf{g}_{\mathbf{reference}} = \mathbf{Accelerometer} + \boldsymbol{\omega}_{\mathbf{gyro}} \times \mathbf{V}$$
$$\mathbf{Accelerometer} = \begin{bmatrix} Accelerometer_x \\ Accelerometer_y \\ Accelerometer_z \end{bmatrix} \qquad \text{Eqn. 26}$$

除了重力矢量的基准测量值，我们还需要基于方向余弦矩阵的一个估计值，主要体现在方向余弦矩阵的最后一行，即地球参考坐标系垂直向下轴相对于机体坐标系的投影。



在机体系下，横滚俯仰转动的修正矢量，是通过计算方向余弦矩阵的第三行和重力参考矢量归一化后值得叉乘积。

$$\mathbf{RollPitchCorrectionPlane} = \begin{bmatrix} r_{zx} \\ r_{zy} \\ r_{zz} \end{bmatrix} \times \mathbf{g}_{\mathbf{reference}} \qquad \text{Eqn. 27}$$

在非常紧凑连续的转弯中，加速度计也许会饱和。换句话说，实际的运动加速度可能会超出加速度计的测量范围。在那样情况时，将会引入横滚俯仰角度方向的错误估计。控制应避免使加速度计饱和，类似地，陀螺仪也会在这个快速转弯时将会饱和。这个可以通过陀螺仪值反馈校正去限制转动速度。

反馈控制器

每一个旋转漂移矫正矢量（偏航、滚转和俯仰）乘以重量，输入到比例积分反馈控制器，增加陀螺仪矢量以产生一个矫正的陀螺仪矢量用作方程17的输入。（现在可以回到图1了。）计算过程如下。首先我们计算总共的旋转修正的平均加权。在我们的案例中，有两个修正，但是通常会更多：

$$\begin{aligned}&\mathbf{TotalCorrection} = \\ &\qquad W_{RP}\mathbf{RollPitchCorrectionPlane} \\ &\qquad + W_{Y}\mathbf{YawCorrectionPlane}\end{aligned} \qquad \text{Eqn. 28}$$

接下来，我们把总的修正传输到比例积分控制器：

$$\begin{aligned}\omega_{\mathrm{PCorrection}} &= K_P\mathbf{TotalCorrection} \\ \omega_{\mathrm{ICorrection}} &= \omega_{\mathrm{ICorrection}} + K_I dt\mathbf{TotalCorrection} \\ \omega_{\mathrm{correction}} &= \omega_{\mathrm{PCorrection}} + \omega_{\mathrm{ICorrection}}\end{aligned} \qquad \text{Eqn. 29}$$

然后我们把陀螺仪修正矢量以通过给陀螺信号增加修正矢量的方式反馈到旋转更新方程，如方程16所示。

就此，要通过一个完整的计算来完成。下一步我们重复这整个计算。

一些读者可能不知道为什么我们使用一个单一的反馈控制器的加权输入，而不是每一个向量一个单独的控制器。其实，我们可以这样，但长时间单独的积分器将积累等值反向的误差，最终可能导致积分器饱和或翻转。测试表明，这种情况的发生将需要很长的时间。然而，使用一个单独的控制器结果会更加正确。

权重和增益的选择需要在准确性和从干扰中恢复的速度方面进行折衷。使用重量和收益，有足够大的恢复在约10秒的风力和陀螺仪的实际情况。在风和陀螺仪饱和的实际情况下，用权重和增益比较好，最多10秒钟就可以恢复。在反馈环里，DCM算法是一种非线性积分器。因此，你可以为整个环选择线性化等效的动态模型的增益。

陀螺仪的特点

风

必须把风的影响考虑进来，主要是偏航。因为我们使用地面航线来实现锁定航向，方向余弦矩阵轴将最终与飞机的方向对齐，而不是指向它的方向。这是没有问题的，当飞机是在一条直线上 直接回到RTL，或在航点之间使用DCM实现自动驾驶。 当算法把风的因素当成是漂移的情况下，会发生什么事， 并逐步旋转飞机的角度，保持地面航线的偏转角。然而，当飞机转弯时，它会适应风向和地面航线间新的角度，用有限的DCM适应之间的新角度 风向和地面的过程。有两个解决问题的途径：

- 用足够大的反馈增益，以适应几秒钟内风或者航线的变化，我们目前使用的方法，在我们的固件中，目前是适用这种方案。
- 计算风矢量。理论上是可行的，考虑到陀螺仪偏移较小，假如飞机转弯。我们目前正在考虑这种方法，以看它是否比上面的方法效果更好。现在，在转弯后会继续适应风的因素，实际上这确实可更好的工作。

使用DCM控制和导航

在上一节中我们介绍了几个方向余弦用于控制和导航的应用示例。在这一节中，我们提供了更多的细节：

- 1、想要控制飞行器的俯仰角，就需要知道飞行器当前的俯仰姿态，俯仰姿态可以通过飞行器的横滚轴与铅垂向量（ground vertical）的点积求出。

飞行器横滚轴 (X) 与铅垂向量（ground vertical）(Z) 的点积，就是一个方向余弦，r\(_{zx}\)。它就等于横滚轴与地面参考坐标系水平面的夹角的正弦值。所以，矩阵元素直接指示出横滚轴是否与地面平行，且可以直接用于俯仰控制的反馈回路。当飞行器水平时，r\(_{zx}\) = 0 。

- 2、想要控制飞行器的横滚角，就需要知道飞行器当前的滚转姿态，滚转姿态可以通过飞行器的俯仰轴与铅垂向量（ground vertical）的点积求出。

飞行器俯仰轴 (Y) 与铅垂向量（ground vertical）(Z) 的点积，是一个方向余弦，r\(_{zy}\) 。它等于俯仰轴与地面参考坐标系水平面的夹角的正弦值。所以，矩阵元素直接指示出俯仰轴是否与地面平行，且可以直接用于横滚控制（原文为俯仰，疑似笔误。）的反馈回路。当飞行器水平时，r\(_{zy}\) = 0。

- 3、想要导航，就需要知道飞行器当前相对于目标点方向的偏航姿态，可以通过飞行器横滚轴与目标点方向的叉乘得到。即使飞行器上下颠倒，依然成立。想要知道飞行器是否飞向相反方向，只需对横滚轴和期望方向向量求点积。如果是负的，则飞行器偏离航向超过 90°。

飞行器的横滚轴向量是旋转矩阵（R）的第一列。我们只需要水平分量，所以，将 Z 轴分量设为 0 。所得向量为：$\begin{bmatrix} r_{xx} & r_{yx} & 0 \end{bmatrix}$。将该向量与期望方向向量求叉乘，得到偏航角度的正弦值；求点积得到余弦值。

- 4、想要知道飞行器是否上下颠倒，只需检查飞行器偏航轴与竖直向量点积的符号。如果是负的，则飞行器上下颠倒。

飞行器偏航轴与竖直向量的点积是矩阵元素 r\(_{zz}\) 。当飞行器接近水平飞行时，该元素约等于 0 。当飞行器上下颠倒时，该元素约等于 -1 。当飞行器（原文为固定翼）侧向倾斜 90° 时，该元素为 0 。

- 5、想要知道飞行器绕竖直轴的转向速度，将陀螺仪旋转向量转换到地面参考坐标系，对该向量和竖直轴做点积。

这就等于对旋转矩阵（R）的第三行与陀螺仪旋转向量的点积。所以，飞行器绕地垂轴的转向速度为：$\omega_x r_{zx} + \omega_y r_{zy} + \omega_z r_{zz}$ 。

09:41:52

设计实现

参考

Robert Mahony等人撰写如论文地址为 http://gentlenav.googlecode.com/files/MahonyPapers.zip。