**Direction Cosine Matrix IMU: Theory**

William Premerlani and Paul Bizard

This is the first of a pair of papers on the theory and implementation of a direction-cosine-matrix (DCM) based inertial measurement unit for application in model planes and helicopters. Actually, at this point, it is still a draft, there is still a lot more work to be done. Several reviewers, especially Louis LeGrand and UFO-man, have made good suggestions on additions and revisions that we should make and prepared some figures that we have not included yet. We will eventually incorporate their suggestions, but it may take a long time to get there. In the meantime, we think there is an audience who can benefit from what we have so far.

The motivation for DCM was to take the next step in stabilization and control functions from an inherently stable aircraft with elevator and rudder control, to an aerobatic aircraft with ailerons and elevator. One of the authors (Premerlani) built a two axes board several years ago, and developed rudimentary firmware to provide stabilization and return-to-launch (RTL) functions for a Gentle Lady sailplane. The firmware worked well enough, and the author came to rely on the RTL feature, but it never seemed to work as well as the author would like. In particular, satisfactory solutions to the following two issues were never found:

• Mixing. It was recognized that in a banked turn, there were two problems arising from the bank angle. First, the yaw rotation of the aircraft around the turn generated a nuisance signal in the pitch gyro, because of the banking. Second, in order to make a level turn, the elevator needed some “up” deflection. The amount of deflection depends on the bank angle, which could not be directly measured. Both issues were opposite sides of the same coin. • Acceleration. An accelerometer measures gravity minus acceleration. The acceleration is equal to the total of all of the aerodynamic forces (lift, thrust, drag, etc.) on the plane, plus the gravity force, divided by the mass. Therefore, the accelerometer measures the negative of the total of all of the aerodynamic forces. The measurement of gravity is what is needed to level the plane but that is not what you get out of an accelerometer during accelerated motion. Acceleration is a confounding variable. In particular, when the aircraft pitches up or down, for a short while it accelerates in such a way that the output of an accelerometer does not change. There is a similar effect that the NASA astronauts experience when they are in training planes. A ballistic path can produce zero net forces and therefore fool accelerometers temporarily. The combination of this issue and the previous one prevented really tight pitch control, and this issue prevented the use of pitch stabilization during a hand launch.

It was realized that part of the problem was not having a six degree of freedom inertial measurement unit (IMU), so it was decided to design a new board. The UAV DevBoard from SparkFun was the result.

Coincidentally, one of us (Premerlani) decided he wanted to step up to an aircraft with ailerons, and found that he just did not have the needed flying skills. He crashed 5 times in one summer, and had to completely replace his plane 3 times. So, he decided to use his new board for stabilization, shown below, attached to his Goldberg Endurance with Velcro.



The question was, how best to do that? Working together, we came to the same conclusion of Mahoney [1]. What is needed is a method that “fully respects the nonlinearity of the rotation group.” Paul and I decided that we should represent the rotation with a direction cosine matrix, that we could maintain the elements of the matrix using gyro, accelerometer, and GPS information, and that we could use the matrix for control and navigation. At a high level, here is how DCM works:

1. 陀螺仪是运动方向的主要来源，对当前飞机旋转时的时间变化率进行非线性微分运动学方程求积分。这个动作的频率较高（40Hz到50Hz），通常要给伺服系统足够的时间发送执行用的PWM信号。
2. 鉴于在积分时产生的微小误差的累计会渐渐破坏DCM必须满足的正交性约束，我们必须定期对其做微小的矫正以使其满足约束条件。
3. 鉴于数值的误差，在DCM元素中，陀螺漂移和补偿将会逐渐累积误差，我们使用参考向量来监测误差，在误差和角速度输入（步骤1）之间使用比例积分(PI)负反馈控制器，以此来消除误差。GPS被用来监测YAW轴的误，加速度计被用来监测PITCH 和ROLL的误差

过程见图Figure 1.

Error

Adjustment

PI Controller

RMatrix

]

[

Orientation

W

Rmatrix

Kinematics

&Normalization

[

XYZGyros

]

Gyros

[

Accelerometers

]

Gravity

DriftAdjustment

Rmatrix

Yaw

Pitch Roll

Error

Drift Detection

]

GPS

[

Course

Figure 1 DCM 框图

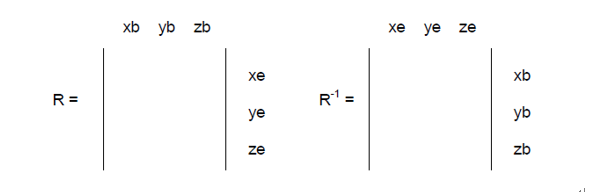
No doubt you are wondering what a rotation group is, and why it should be respected. You also might be wondering how you can use DCM for control and navigation. You also might have the same questions that UFO-MAN asked on the subject after he read Mahoney’s paper, so we will start with those questions:

* What is a quaternion and why do we use that instead of vector notation? 什么是四元数法，我们为什么要使用它的矢量符号？
* What is meant by a rotation group? 旋转群是什么？
* What is a rotation matrix? 什么事旋转矩阵？
* What does it mean to maintain orthogonality of the rotation matrix?

“保持旋转矩阵正交性”是什么意思？

* What is an anti symmetric matrix?反对称矩阵是什么？
* Can you briefly explain kinematics（运动学） in this rotation matrix context?
* Can you briefly explain dynamics（动力学） in this rotation matrix context?

做的所有事都针对于旋转，我们要做的其实就是飞机参考系相对于地球参考系的旋转。有几种方式可以做到这一点。Mahony的论文讨论的两种类似的方法，旋转矩阵和四元数，两种的处理方式都很相似，都用无近似和无奇点的方式表示旋转。四元数的优点在于仅仅只有4个值，而旋转矩阵有9个。旋转矩阵的优点在于很适合控制和导航。我们选择旋转矩阵，因其微弱优势以及我们更加熟悉。



旋转矩阵描述的是一个参考系与另一个参考系的关系。在其它系统看来，某个系统的矩阵的一列表示单位向量。一个系统通过乘上旋转矩阵转换为另一个系统。反向的选择通过旋转矩阵的逆，这等于他的转置（交换行列）。将数据变成单位向量，可以通过他们的点乘和叉乘获得各种角度的正弦和余弦。

z’

y’ = R.y y’

y = R

-1

.y’

x

y

z

x’

R

-

1

=

t

R

R.R

-

1

= I

t

R.R = I

As your plane flies along, it is possible to describe its motion with a translation (movement of its center of gravity) and a rotation (change in orientation around its center of gravity). Its orientation with respect to the earth can be described by specifying a rotation about an axis. By starting with the plane level and pointing in a standard direction and applying the rotation, you will place the plane in its actual orientation. Any orientation can be described as a rotation from the “standard” position.

A rotation group is the group of all possible rotations. It is called a group because any two rotations in the group can be composed to produce another rotation in the group, every rotation has an inverse rotation, and there is an identity rotation. That is the definition. However, the way that we like to think about it as being a group is that you can wind up going around a complete circle and arriving back where you started. The rotation group is closed.

The reason that the rotation group should be respected is that by doing that, you make the fewest approximations and are able to perform control and navigation with the plane in any orientation, including upside down and pointing vertical. You can do aerobatics without making any approximations.

The basic idea is that the rotation matrix that defines the orientation of your aircraft can be maintained by integrating the nonlinear differential equation that describes the kinematics of the rotation. (We will present the nonlinear differential equation shortly, and explain why it is nonlinear.) Kinematics is concerned with the geometry of the rotation of a rigid body, and how the rotation transforms one rigid configuration into another configuration. This is done by recognizing that the integration can be accomplished via a series of matrix compositions.

By matrix composition we simply mean multiplying two rotation matrices together. It can be shown that the resulting matrix represents the net rotation that results from applying the two rotations in sequence that each of the matrices represents.

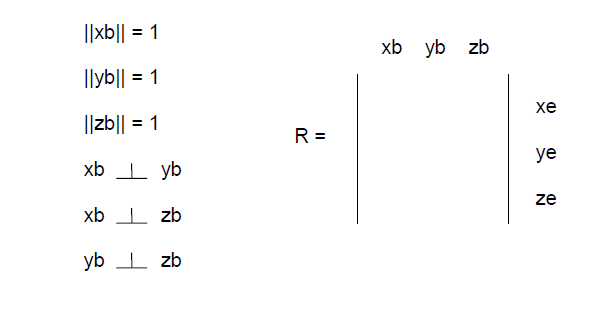
However, numerical integration introduces numerical errors, and does not produce the same result that symbolic integration. An exact symbolic integration of the exact gyro signals will produce the exactly correct rotation matrix. Numerical integration, even if we had the exact gyro signals, will introduce two sorts of numerical errors:

* Integration error. Numerical integration uses a finite time step and data that is sampled at a finite sampling rate. Depending on the method that you use to do the integration, you are making certain assumptions about what is happening between data samples. The method that we use in our implementation assumes that the rotation rate is constant over the time step. This introduces an error that is proportional to the rotational acceleration.
* Quantization error. No matter what representation you use for the values, the digital representation is finite, so there is a quantization error, starting at the analog to digital converter, and building whenever you perform a calculation that does not preserve all of the bits of the result.

One of the key properties of the rotation matrix is its orthogonality, which means that if two vectors are perpendicular in one frame of reference, they are perpendicular in every frame of reference. Also, that the length of a vector is the same in every frame of reference. Numerical errors can violate this property. For example, since the rows and columns are supposed to represent unit vectors, their magnitude should be equal to one, but numerical errors could cause them to get smaller or larger. Eventually they could shrink to zero, or go to infinity. The rows and columns are supposed to be perpendicular to each other, numerical errors could cause them to "lean" into each other, as shown below:

The rotation matrix has 9 elements. Actually, only 3 of them are independent. The orthogonality property of the rotation matrix in mathematical terms means that any pair of columns (or rows) of the matrix are perpendicular, and that the sum of the squares of the elements in each column (or row) is equal to 1. So, there are 6 constraints on the 9 elements.

旋转矩阵有9个元素，但只有三个是独立的，其余六个是使其正交的约束条件



An antisymmetric matrix is a matrix in which each element in the matrix is equal to the negative of the element with swapped row and column index. So, for example, if the element in the first row, third column is 0.5, then the element in the third row, first column must be -0.5. Also, the elements on the diagonal of an antisymmetric matrix must be zero.

It turns out that a small rotation can be described with an antisymmetric matrix as shown below:

0 a b

1. 0 c

1. -c 0

在我们的例子中,运动学涉及刚体转动的影响。它导致一个非线性微分方程描述身体的时间演化的方向的矢量旋转速度。方向余弦矩阵都是关于运动学。

动力学在我们的案例中是牛顿定律来描述的应用程序的时间变化率旋转速度矢量的应用扭矩。

顺便说一下,马赫尼的动力学的论文是不准确的飞机,他们主要考虑的直升机和垂直起飞。马赫尼的论文描述了如何实现一个组合定位测量和控制算法。保罗和我所做的只涉及运动学。现在我们已经完全无视动力学。运动学(旋转矩阵)本身是非常有用的提供一个基础控制和导航的飞机模型。

你可能还不知道如何使用DCM。控制和导航与DCM完全可以实现使用矢量叉乘和笛卡尔坐标点的产品。举个例子,在一个较高的水平,这是如何完成这四个控制和导航计算。

1. 要控制飞机的俯仰，就需要了解飞机的俯仰姿态，可以通过将飞机的滚动轴与地面垂直向量的点积求得。
2. 要控制飞机的滚动，就需要了解飞机的姿态，可以通过将飞机的俯仰轴与地面垂直向量的点积求得。
3. 要航行，你需要知道飞机的偏航姿态，你要走的方向，你可以通过把飞机的横滚轴与目标方向做叉乘。这样即使飞机是上下颠倒都可以。要知道飞机是否朝着目标相反的方向，将横滚轴和期望的方向进行点乘，如果是负的话，这架飞机航向偏离90以上
4. 要找出这架飞机是否倒过来，检查飞机偏航轴与垂直方向的点积的值。如果小于零，飞机便是倒.过来的
5. 要找出飞机绕垂直轴旋转速度，将陀螺旋转矢量转换为参考坐标系下的矢量，并与垂直轴作点积。

We now get deeper into the details of the theory.

# Axis conventions

要描述飞机的运动必须定义适当的参考系. 对于大多数处理飞机运动的问题，采用了双坐标系。一个坐标系是固定在地球上的，可以被认为是飞机运动分析的目的是一个惯性坐标系(下图的e)。另一个坐标系是固定在飞机上的，被称为机体坐标系（下图的b）

xe

ye

ze

xb

yb

zb

φ

θ

φ

θ

ψ

ψ

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。角旋转被称为欧拉角。对固定接地框架的车身框架的方向可以用以下方式来确定。想象一下飞机的位置，使身体轴系统是平行的固定框架，然后应用下面的旋转：

机身绕ZB轴旋转是 yaw angle ψ

机身绕YB旋转是 pitch angle θ

机身绕XB旋转是 roll angle φ

yb

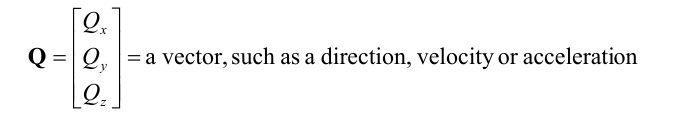
xb

zb

Figure 3 Body axes coordinate system

# Direction cosine matrices

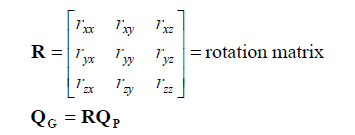
Certain types of vectors, such as directions, velocities, accelerations, and translations, (movements) can be transformed between rotated reference frames with a 3X3 matrix. We are interested in the plane frame of reference and the ground frame of reference. It is possible to rotate vectors by multiplying them by a matrix of direction cosines:



矢量，如方向，速度或加速度

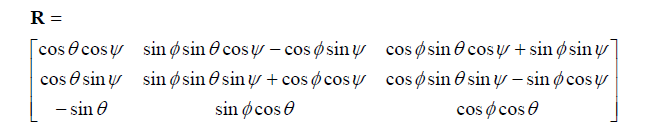
**QP** = a vector Q measured in the frame of reference of the plane

**QG** = a vector Q measured in the frame of reference of the ground



Eqn.1

方向余弦矩阵与欧拉角的关系:



Eqn.2

Equation 1 and equation 2 expresse how to rotate a vector measured in the frame of reference of the plane to the frame of reference of the ground. Equation 1 is expressed in terms of direction cosines. Equation 2 is expressed in terms of Euler angles.

如公式1，每个地面向量等于旋转矩阵和机体坐标系下的相应行的点积。Nine multiplies and six additions are required to compute the rotation. Equation 3 is a restatement of equation 1, with the matrix multiplication expanded in terms of the elements of the vectors and the matrix.

*QGx* = *rxxQ Px*+*rxyQ Py*+*rxzQPz*

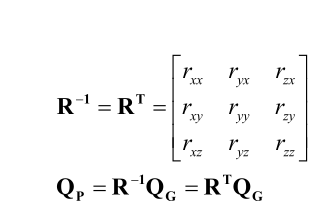
*QGy* = *ryxQ Px*+*ryyQ Py*+*ryzQ Pz*  Eqn. 3

*QGz* = *rzxQ Px*+*rzyQ Py*+*rzzQPz*

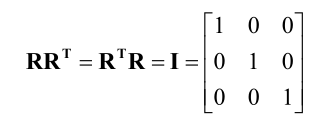
R旋转矩阵不一定是对称的.矩阵R中的三列来自于机体相对于对大地的三轴的相对量.矩阵R中的三行对应大地坐标系相对于机体坐标系的三轴的相对量. R矩阵包含所有飞机关于底面参考系的定向信息。R矩阵也被称为方向余弦矩阵, 因为每个 entry 都是机体轴和地面轴的角度余弦. 尽管矩阵R中似乎有9个独立的轴，但实际上只用3个轴, 因为其它6个轴因为要遵循所谓的矩阵的正交性 (也被成为常态化或归一化):三列向量互相垂直并且为单位向量.

矩阵的转置用这个表示 **RT** , 变换是交换了行和列. 总之是一个相反的矩阵,

矩阵的逆等于其转置，具体看下公式. (The identity matrix has all ones on the diagonal, and all zeros everywhere else. Multiplying any matrix by the identity matrix leaves it unchanged. In the case of rotation matrices, it turns out that the transpose of the R matrix is equal to its inverse:

 Eqn.4

对旋转矩阵的逆矩阵等于其转置的原因是由于对称性的情况。旋转矩阵的元素是对轴之间的余弦，一个飞机参考系，和一个地参考系。即交换行和列相同，转置相同。而且，事实上，逆矩阵等于其转置矩阵与正交条件一致，这可以用矩阵表示为：

 Eqn.5

Epan.5用来证明R的逆与R的转置相等， R的逆乘以方程，或由R逆转置乘以方程。

旋转矩阵的一个非常有用的特性是，我们可以组合旋转. We can multiply several rotations matrices together, and get a rotation matrix that is equivalent to applying all of the rotations in succession. We have to be careful to apply the rotations in succession on the left side of what we already have. For example, if we have three rotation matrices, from orientation A to orientation B, from B to C, and from C to D, we can compute the rotation matrix that will go from orientation A to orientation D according to:

**RDA** = **RDCRCBRBA**

**RBA** = rotation matrix from A to B

**RCB** = rotation matrix from B to C

**RDC** = rotation matrix from C to D

**RDA** = rotation matrix from A to D

Eqn.6

我们必须要小心的是，乘法旋转矩阵的运算是没有交换律的. That is, the order of matrix multiplication matters very much.. This is consistent with rotations, which are not commutative either. For example, consider what happens if a plane pitches around its own pitch and roll axes by 90 degrees each. The order very much matters. Suppose that is pitches up by 90 degrees, followed by a roll of 90 degrees. At that point the plane will be traveling vertically. However, if it rolls first, and then banks, it will be traveling in the horizontal plane.

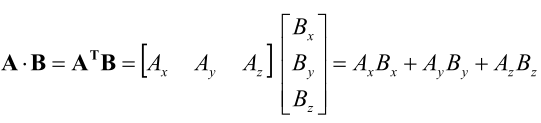
Finally, there is a useful identity that applies to matrices in general, and to rotation matrices in particular. The transpose of the product of two matrices is equal to the product of the transposes of the matrices, with the two matrices swapped:

(AB)T = B AT T

Eqn.7

**A,B** are matrices

**向量的向量积（叉积、外积、叉乘）和点积（点乘）**

Two very useful vector products that we will use in computing DCM and in using its elements for navigation and control are the dot product and the cross product. The dot product of two vectors A and B, is a scalar computed by performing a matrix multiplication of a A as a row vector with B as a column vector producing:  Eqn.8

It turns out that the vector dot product produces a result that is equal to the product of the magnitudes of the two vectors, times the cosine of the angle between them:

 Eqn.9

We note that the dot product is commutative（交换律）: **A** ⋅ **B= B** ⋅ **A**

The cross product of two vectors A and B, is a vector whose components are computed by:

(**A** × **B**)*x* = *Ay B z* − *Az By*

(**A** × **B**)*y* = *Az B x* − *Ax B z* Eqn.10

(**A** × **B**)*z* = *Ax B y* − *Ay Bx*

The cross product is perpendicular to both of its vector factors and its magnitude is proportional to the magnitudes of the vectors times the sine of the angle between them:

u

v

w

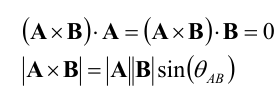
k

θ

w = u v = ||u||.||v||.sin(θ).k

||w|| = ||u||.||v||.sin(θ) k : unit vector orthogonal to the plane defined by u and v

Stated another way:

 Eqn.11

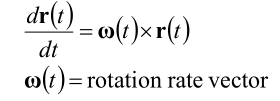
We note that the cross product is anti-commutative（反交换律）. **A B**× =−**B A**×

# 计算陀螺仪的方向余弦

在了解以上或更多的基本原理后, 我们现在讲DCM算法的核心概念：与陀螺仪信号的方向余弦改变的时间变化率相关的非线性微分方程。我们的目标是准确的计算方向余弦. 我们暂时假设陀螺仪没有误差. 之后我们再处理陀螺仪的漂移

不像旋转机械陀螺, 当飞机绕着一个固定的东西转动, 飞机的电子陀螺仪旋转和参数的信号的旋转速度成正比. 由于旋转不可以commute, 还有陀螺仪的旋转顺序问题, 我们不能简单的吧陀螺仪信号变化为角度, that will not work. 让我们来看看需要做什么来得到正确结果

一个总所周知的运动学结论：旋转矢量的旋转矢量变化率由自身旋转引起如下：



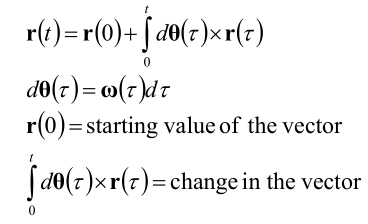
r为旋转矢量，即Eqn.2中的R

Eqn.12

我们可以看出:

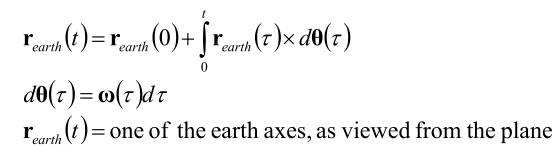
1. 微分方程是非线性的. 将旋转向量的输入叉乘我们试图整合的变量. 因此任何线性方法得到的仅仅是一个近似值.
2. 两个向量必须在同一参考系下测量.
3. 因为叉乘满足反交换律, 我们可以颠倒顺序和改变符号

If we know the 输入参数and 旋转矢量的时间变化, we can numerically integrate equation 11 to跟踪旋转矢量:

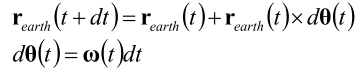
 Eqn.13

我们的策略是将方程13应用到矩阵R的行或列，将它们作为旋转向量处理。

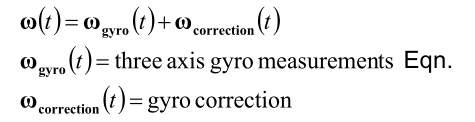
我们碰到的第一个问题就是，我们要跟踪的矢量和旋转矢量，不在同一个坐标测量。理想的情况下，我们想在地球参考系中跟踪飞机的轴线，但陀螺仪测量是在飞机的坐标系下进行的。有一个简单的解决方法，通过识别旋转中的对称性。在飞机参考系中，地球参考系与地球参考系中的飞机的旋转相等且相反。所以我们可以通过翻转角速度的信号来跟踪飞机参考系中的地球坐标轴。方便起见，我们可以改变符号，和交换叉乘系数：

 Eqn.14

方程14中的向量就是是矩阵R（Eqn.1）中的行. 下一个问题是如何方便的实现方程14.我们采用和 Mahoney 相同的矩阵方法. 让我们回到微分方程14:

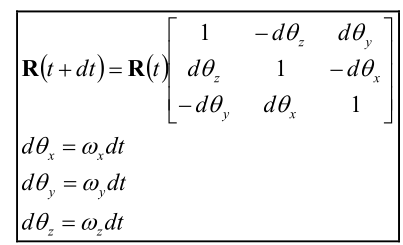
 Eqn.15

还有就是, 我们将在之后消除预期的漂移. 我们需要添加一个比例积分漂移补偿反馈控制器来对陀螺仪的旋转速率进行校正，参数真正的旋转速率估计值:

 Eqn.16

稍后我们将详细解释计算的陀螺仪校正向量。基本上，我们的GPS和加速度计的参考向量来计算旋转误差，并通过反馈控制器的输入计算，然后更新矩阵方程（Eqn.16）

When we repeat equation 14 for each of the earth axes, we can put the result into a convenient matrix form:

 Eqn.17

Equation 17 is a recipe for updating the direction cosine matrix from gyro signals. It is equivalent to Manhoney’s result. The values of 1 on the diagonal of the matrix in equation 17 represent the first term in equation 15. The smaller, off-diagonal elements represent the second term in equation 15. Equation 17 is implemented numerically by repeated matrix multiplications, with short time steps. Each matrix multiplication requires 27 multiplications and 18 additions. It maps well to the dsPIC30F4011, which has hardware resources to perform matrix multiplication efficiently. It can be performed on CPUs that do not have matrix support, in which case it is recommended to use integer arithmetic.

The only approximation that equation 17 makes is that the time step is short enough so that the R matrix does not change much from step to step. A typical time step is around 0.020 seconds, during which an aircraft rotating at around 60 degrees per second rotates approximately 0.020 radians, which translates to a maximum change in any of the R matrix coefficients of around 2%. Thus, the second order terms that are being ignored are on the order of 0.02%.

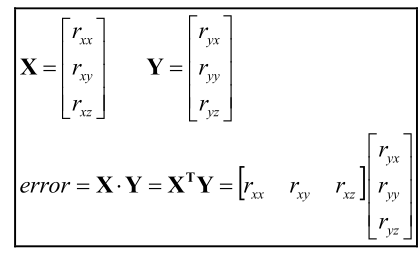
# 测试和仿真结果表明，方程15本身的实现，在具有适度的性能的陀螺上，得到非常准确的结果，非常低的漂移，在几度每分钟的顺序。漂移是如此之低，这是一个简单的问题，调整，而不影响性能。然而，通过它自己，方程15将最终积累数值圆形和陀螺仪漂移，偏移和增益误差。在接下来的两节中，我们将解释如何取消错误。

# 归一化（重整）

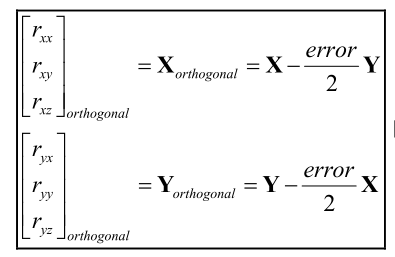
数值误差的累计将逐渐破坏Eqn.5中的正交性条件. In effect, the axes in the two frames of reference no longer describe a rigid body. Fortunately, 数值误差累积得很慢, 所以走在他前面是一件很简单的事.

我们称正交化这个过程为重整（归一化）. We devised several ways that it could be done. Simulations showed they all worked quite well, so in the end, we settled on the simplest approach. It works as follows.

首先我们计算矩阵X行和Y行的点积,他应该等于零 , 其结果是X Y行彼此旋转多少的量(产生的误差大小):

 Eqn.18

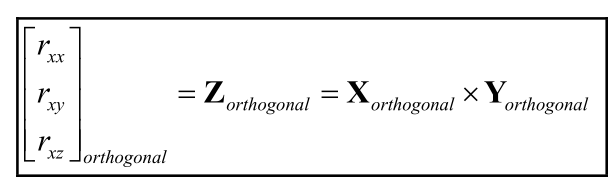
我们分别分配误差的一半给X Y行, and approximately rotate the X and Y rows in the opposite direction by cross coupling:



Eqn.19

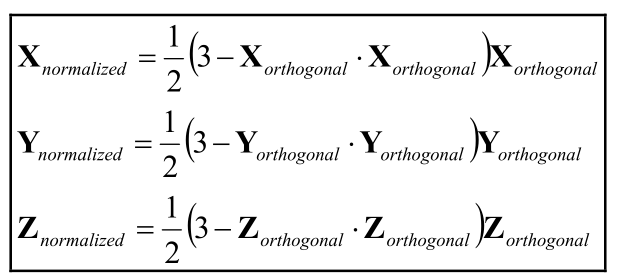
你可以将方程18带入到19里面验证方程的正交性, 记住R矩阵每行每列的大小大约等于1。将误差平均分配到 每个向量上矫正误差的效果比只分配到一个向量上的效果好。

接下里矫正垂直于X和Y轴的行Z。Z等于X行与Y行的叉积:



Eqn.20

最后要做的是检测矩阵R的行，保证每行等于1。 其中方法之一就是求每行的模（每行每个元素的平方和开平方根）。还有一个简单的方法，by recognizing that the magnitudes will never be much different than one,于是利用泰勒展开式计算。调整行向量的结果大小:



Eqn21

What equation 21 says to do to adjust the magnitude of each row vector to one, is to subtract the dot product of the vector with itself (the square of the magnitude), subtract from three, multiply by ½, and multiply each element of the vector by the result.

There are not that many multiplies and additions in the normalization process. There are no divisions or square roots. We perform the calculation for each step of the integration, every 0.020 seconds.

# 消除漂移

虽然陀螺仪表现良好，但是没修正就会用每秒几度的漂移, 最终我们不得不解决漂移 . 解决方法是用其它的参考方向来检测陀螺仪的误差并且提供一个负反馈循环以补偿误差，用了一个经典的检测反馈循环，见Figure 1. 步骤:

1. 使用方向参考向量来检测定位误差是通过计算一个旋转向量，将实测值与参考向量的计算值的对齐。
2. Feed the rotation error vector back through a proportional plus integral (PI) feedback controller to produce a rotation rate adjustment for the gyros. (A PI regulator is a special case of a commonly used feedback regulator called a PID regulator. The D stands for derivative. In our case, we do not need the derivative term.)
3. 增加 (或者减去，根据你的旋转误差的符号) 输出一个PI控制量来得到真正的陀螺仪信号.

参考向量的主要要求是它不会漂移. 他的瞬态性并不重要 因为是陀螺仪提供瞬时的精确定向估计.

我们的两个参考向量由GPS和加速度计提供. 磁力计也很好用, 特别是空中悬停应用的偏航控制, 但飞机的一般飞行，一个GPS就够了.如果要使用磁力计, to provide a vector reference you should use a three axis magnetometer. Low cost three axis magnetometers are commercially available.

我们使用加速度计提供飞机Z轴的参考向量. Details will be given in a separate section. 我们使用GPS来提供飞机x轴(roll axis)的参考. 两个参考向量恰巧是相互垂直的. That is convenient, but not absolutely necessary.

对于这两个参考向量,通过测量的向量与方向余弦矩阵估计的向量叉乘来监测方向误差. 叉积特别适合的原因有两个，其大小与两个向量之间角的正弦成正比，并且其方向垂直于两者. 所以他代表一个旋转轴和一个旋转量, that would be needed to rotate the measured vector to become parallel to the estimated vector. In other words, 它等于负的旋转误差. By feeding it back to the gyros through the PI controller, the estimated orientation is gradually forced to track the reference vectors, and gyro drift is cancelled.

The cross product of a measured reference vector with a corresponding vector computed from the direction cosine matrix is an indication of the error. It is approximately equal to the rotation that would have to be applied to the reference vector to bring it into alignment with the computed vector. We are interested in the amount of rotation correction that we need to apply to the direction cosine matrix, which is equal to the negative of the error rotation. So, it is convenient to compute the correction by interchanging the order in the cross product. The correctional rotation is equal to the cross product of the vector estimated by the direction cosines with the reference vector.

我们使用比例积分反馈控制来修正陀螺的漂移, 既因为他稳定，还因为它的积分项完全消除了陀螺仪的漂移 包括温漂和零漂.

The way that the reference vector errors map back onto the gyros is done via the direction cosine matrix, so that the mapping depends on the orientation of the IMU. For example, the GPS reference vector might correct either the X, Y, Z, or combinations of X, Y and Z axis gyro signals, 取决于地球参考系的轴向.

We will now get into more detail for the two references that we are using.

# GPS

GPS 提供了一个无漂移的飞机的偏航向量. 我们不用GPS直接作为偏航值是因为GPS的瞬态响应远比陀螺的慢，相反，如果我们将GPS作为一个参考向量，来取消陀螺仪的漂移，便能实现偏航的锁定.

GPS能够提供位置、速度大小和方向的信息.GPS通过从卫星接收到 位置和速度 并通过串口发送信息.对于大多数的GPS接收器来说，数据有两种格式, [NMEA](http://baike.baidu.com/link?url=xz-ZAX1Hhg28Xj_vq7OynXaIaVwRJuGhtmGyrFE1k0MJlS9s3rZKp8UTfg1Z0GSwLtXRcF6leRugNz4KnXRB5K) and 二进制. NMEA用逗号分隔，ASCLL码表示，可读性高。用二进制， 二进制传输即用0和1的序列传输，它相对于NMEA来说会提供一些额外的NMEA没有的信息。

GPS必须有序移动以提供方向信息. 否则, GPS的天线将无法定向. GPS速度矢量由GPS接收天线的移动速度测得，为每秒移动的距离。有很多种方式来计算，但是每种方法都需要GPS进行移动。

GPS提供的位置和速度 是两种不同的坐标 系统. 一种提供经度, 维度, 高度, 对地速度, 和对地航向.对地航向就是相对于北方测量的航向值。恰巧,正是机身坐标系下Z轴的逆时针方向（Z轴朝下）。垂直速度可以从二进制获取信息的方式中获得。

在另一种系统 ECEF (earth-centered, earth-fixed)系统下, X, Y, Z 的位置和速度信息是以地球为中心的右手坐标系 。

GPS会连续获得信息，通常1秒一次 (1 Hz), 虽然也有更高速率的, 现在5Hz更加普遍。但是, 高速率并不会带来更好的性能, 因为GPS内部处理信号的动态性的限制。

考虑到GPS的动态，有几个因素需要记住：

1. 报告延时. 在某些情况下，对于一些GPS，它可能需要花12秒来计算传输数据.
2. 滤波.所有的GPS都要进行滤波啦提高位置和速度的精度. 这将导致数据变得平滑，当GPS的速度位置改变时，不会立刻显示，而是逐渐变化。
3. 平滑滤波 and static navigation.很多GPS都提供滤波来防止速度或者位置信息的突变。 This is useful for automotive applications to prevent changes from being seen as the result in changes in the satellite signals, such as when collection of satellites that are being used changes. They also provide a “static navigation” option so that variation in the apparent location is suppressed when the velocity falls below a certain value. This is also useful for automotive applications.

It is not likely that you will every run into track smoothing or static navigation, because出厂默认关闭,但应值得注意。 然而,必须考虑到延迟和过滤

By reporting latency we mean a simple time delay between when the GPS measures position and velocity, and when it appears in the sequence of messages. Usually this delay is the reporting time period. For example, if your GPS is reporting at 5 Hz, the reporting latency is typically 0.2 seconds.

However, it could be much larger than that if you are not careful. One of us (Bill) had the bad luck of stumbling into a 12 second latency with a reporting rate of 1 Hz. It turned out that the 12 second delay was triggered by using a combination of 4800 baud and the binary interface. It was reduced to a 1 second latency by changing the baud rate to 19,200. Chances are that you will not run into this effect, but be aware that it exists. 如果你在使用二进制的接口, 你应该将波特率提高到19200以上.

In addition to a simple latency, you will generally also run into a delay caused by internal filtering done by the GPS. All GPS units perform some sort of filtering of the data by the very nature of how they do their computations. There is an inherent compromise in any system between accuracy and transient response. The more accurate you want to know something, the longer it will take to estimate it. In most units, the filtering shows up as a smoothing of the data. Typically, the dynamic response of many types of GPS is a simple exponential response with a 1 second time constant, so that it takes about 3 seconds to fully respond to a step change. If you ignore the GPS dynamics, there will be a small error introduced into your navigation calculations during a turn. One of us (Paul) saw that it is possible to compensate for this small error by introducing a filter between the direction cosine matrix and the input to the yaw drift correction. [Do we need a figure?].

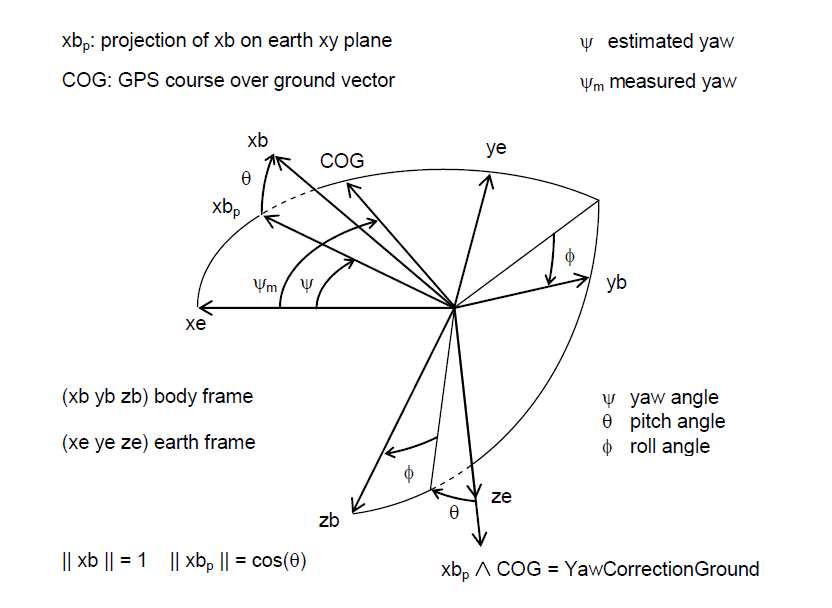
这样，用这两个动态的向量来估计偏航的误差

假定一个GPS有较高的输出率，如5Hz, 它将比通常的 1 Hz的GPS有更好的动态性. 然而没有必要有更高的输出率，但是能减少时延。 However, there is still the issue of the filter dynamics, which will generally turn out to be the limiting effect.

The GPS horizontal course over ground signal has zero drift over the long term, 可以被用来当做IMU的偏航锁.我们也考虑过GPS的垂直速度, 但是决定不用他,更加亲赖于使用加速度计的来获取垂直信息.

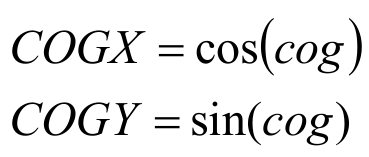
假设飞机正朝向他指向的方向移动.这一假设上的任何瞬态误差不影响性能. 然而强烈的风，特别是侧风, 会打破这个假设。 有两种方式供你选择. 一种方式是计算出风的矢量信息. We are continuing to work on that. 另一种方式是使用适当的反馈增益. The difference between the direction the aircraft is pointing and the direction that it is moving will show up as an error at the input to the drift correction feedback controller. The result will be that DCM will adapt to the wind, and rotate the plane the amount required to keep it moving along the desired course over ground.

下图显示了偏航矫正的计算方法:



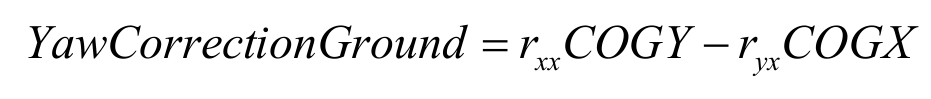
GPS航向（COG）与飞机航向（X轴（xb））在水平方向上的投影（xbp）的旋转误差代表航向的漂移错误。 所以航向校正值为旋转矩阵R的X向量与GPS测得的航向向量的叉积（上图右下角公式）。

首先,我们归一化水平速度向量作为参考向量. 在地球坐标系下，GPS航向（COG）的cosin值作为X轴的值和sin值作为Y轴的值:



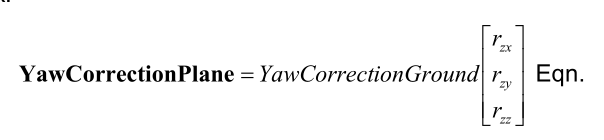
Eqn.22

We then compute yaw correction:



Eqn.23

然而方程23对偏航的纠正是对于地球参考系的. 为了调整偏航误差, 我们需要知道在飞机参考系的修正矢量我们必须乘上R矩阵的Z行:



Eqn.24

方程24所产生的偏航矫正矢量将结合加速度计计算的PITCH-ROLL矫正值来作为矫正误差的总的向量。 详细的计算我们将在加速度计那块讨论。

关于偏航的漂移矫正有三个限制that argue for a large weighting of the yaw correction, 能够对偏航误差进行快速反应.

首先是初始化偏航锁. 当算法开始的时候, 它无法知道我们所指的方向. 即使知道, 在等待GPS锁定期间, 他将会漂移, 甚至在被GPS锁定后, 在飞机启动之前GPS将会产生一个随机偏航. 通过这个将会产生一个很大的偏航误差, 在起飞后不久将会启动偏航锁

第二点是风，如果飞机在侧风中飞了很长时间,风将会被看做一个陀螺仪的补偿. 如果飞机进行了180°的大转弯, 一会儿后DCM算法需要在另一个角度对风进行补偿.

第三个条件是当飞机垂直行驶时。在这段时间内，飞机的×轴是垂直的，并且方程23得到零。

由于这些原因，最好对偏航矫正使用a large weight

# 加速度计

加速度计用于校准俯仰和横滚，因为他具有零漂。我们不必担心离心加速度

When one of us (Bill) built his first board, 他希望能使用加速度计单独控制俯仰和横滚.但是他失败了, 有许多原因. 主要原因是因为他需要测量速度和重力进行组合.如果他测量的仅仅是重力, 它将是完美的.但是他测得是加速度这很碍事. Bill once tried to use accelerometer-only based pitch stabilization during a hand launch of a sailplane. The acceleration of the launch fooled the controls into estimating that the plane was pitching up. The controls responded by pitching the plane straight down.

加速度计的工作方式that it measures the deflection of a small mass suspended by springs. The natural frequencies of the dynamics of the accelerometer are high, so it does respond quickly. The deflection depends on the total force on the mass, which is equal to its mass times the sum of the gravity vector plus the acceleration vector. (The usual sign convention for accelerometers is such that they indicate gravity minus the acceleration.)

So in addition to gravity, an accelerometer also measures acceleration.

That should not be too surprising, since that is what they are called. Therefore, an accelerometer is useful as a roll-pitch indicator only when the plane is not accelerating. The problem is it is often accelerating. Some of the accelerations, such as centrifugal acceleration, are easy enough to compute and compensate for without having a model of the dynamics of the plane.

然而，有没有简单的方法来分别计算的正向加速度。

All is not lost. 通常, 。飞机不会向前加速，很少一直加速或者一直减速，不会一直加速因为空气阻力，也不会一直减速除非停下来了或者坠落了。只要我们不依赖于加速度计的快速瞬态响应, 我们可以用它来对陀螺仪的PITCH –ROLL进行矫正，因为加速度计不会漂移.

There are many good accelerometers on the market, most of them will work just fine with the DCM algorithm. They are not as critical as gyros, because any change in their offsets does generate an accumulated error in the way that a gyro offset does.加速度计是一种直观测量方法，而陀螺仪的测量是根据时间变化率来的

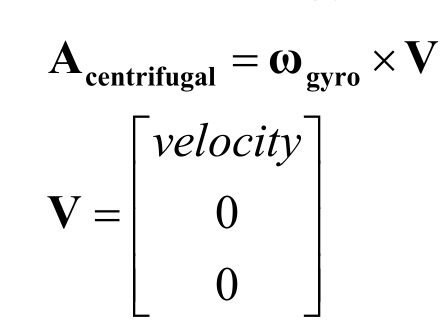
There are a variety of interface types, including analog voltage, pulse width modulation, and several standard communications interfaces. We chose an accelerometer with an analog voltage output as the simplest interface.

The greatest advantage of using direction cosines is that they work for any orientation of the plane, without any singularities（奇点） or special logic. Any orientation can be well-described by the 9 elements of the direction cosine matrix. Since we will need to perform the drift cancellation calculations for any orientation of the plane, we will need to measure acceleration along all three axes of the plane. This can be done with commercially available 3 axis accelerometers, or with 3 separate units.

在我们可以使用加速度计信息来进行漂移补偿之前，我们必须考虑到与飞机前进速度的方向变化的离心加速度。尽管飞机能沿着前进的方向进行加速或者加速，但是不能无限期地进行。

幸运的是，所需的信息计算的离心加速度是现成的。 离线加速度是旋转矢量和速度矢量的叉积。我们并不需要一个准确的答案, 而是需要一个准确的平均值。飞机在其朝向方向的移动的平均值。 因此我们假设速度矢量与飞机的X轴平行. GPS给了我们一个对地速度大小. 因为地面是惯性参考系, 我们能够计算出飞机对于地面在X轴上的速度

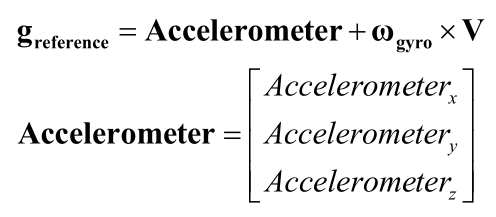
在飞机参考系中，我们计算离心加速度等于陀螺矢量和速度矢量的叉积



Eqn.25

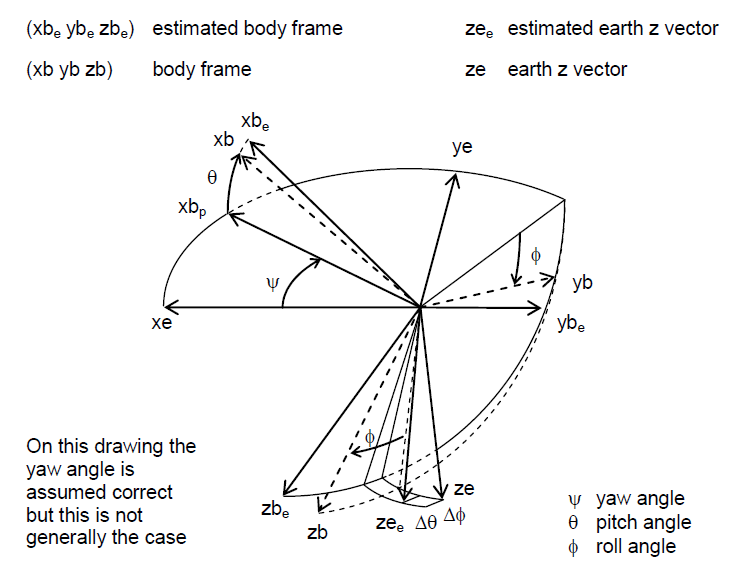
注意在方程25，我们只需要完成两次乘法，因为速度向量有两个元素是零。

常见的商业加速度传感器的方向是Z轴朝下, 并且向下的重力产生正直输出。 因此加速度计的输出是重力减去加速度。 要恢复对离心加速度进行校准了的重力的估计值, 我们需要添加离心加速度的估计值。因此， 飞机坐标系下重力相关的测量如下:



Eqn.26

处理参考重力测量，我们需要一个基于余弦矩阵的估计。由余弦矩阵的Z行展示,也就是地球坐标系“向下 ”的轴在机体坐标系的投影。



The roll-pitch rotational correction vector in the body frame of reference is computed by taking the cross product of the Z row of the direction cosine matrix with the normalized gravity reference vector:

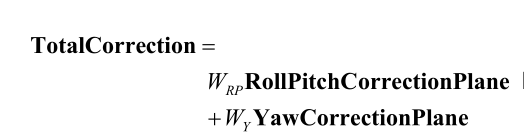


Eqn.27

在非常紧密的连续的转动下，加速度计可能会变得饱和。换句话说，加速度计可能会超过量程。鉴于这个情况，方向估计会引入误差。在控制时应该避免饱和发生。相似地，角速度在快速转动时也会饱和。可以通过在控制反馈中限制旋转速率。

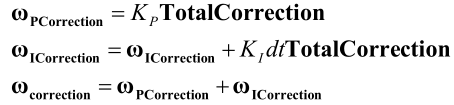
# 反馈控制器

每一个转动漂移修正向量(yaw and roll-pitch) 乘上一个权重 来提供一个比例积分控制器 ，它将产生一个修正陀螺仪的向量，将其带入方程17. (Now is a good time to go back an look at Figure 1.) The calculation proceeds as follows.首先我们计算一个旋转修正的加权平均数. 在我们的例子中只有两个要修改，但一般情况可能会更多:



Eqn.28

Next, we pass the total correction through a PI controller:



Eqn.29

然后我们给陀螺修正向量回旋转的更新方程通过添加修正向量的陀螺信号，如方程式。16。

在这一点上，要完成一个完整的通过计算。在下一个时间步我们重复整个计算。

Some readers may be wondering why we use a single feedback controller with weighted inputs rather than separate controllers for each of the two vectors. Actually, we could, except over a long period of time the separate integrators could accumulate equal and opposite errors that could eventually cause the integrators to saturate or roll over. Tests have shown that that would take a very long time. However, it is more correct to use a single controller.

Selection of the weights and gains is a compromise between accuracy and speed of recovery to disturbances. The practical realities of the wind and gyro saturation favor using weights and gains that are large enough to recover in about 10 seconds. In the feedback loop, the DCM algorithm is a nonlinear integrator. Therefore, you can select the gains for the linearized equivalent dynamic model of the complete loop.

# Gyro characteristics

陀螺仪的灵敏度、操作范围、偏移、漂移、校准、饱和度 在使用DCM是都必须进行考虑

* 陀螺仪的灵敏度– Usually expressed in millivolts per degree per second for a gyro with an analog output, gyro sensitivity is the gyro gain for converting rotation rate to a voltage. In the early phases of the development of DCM, it was thought that sensitive gyros, on the order of 15 millivolts per degree per second, were needed because they usually had low offset and drift. It turns out that DCM works well with other units with lower sensitivity. Gyro sensitivity is related to operating range. The more sensitive the gyro, the narrower the useful operating range and vice-versa. Gyro sensitivity must be taken into account for gyro calibration. Some analog gyros provide an output voltage that is referred to an absolute voltage reference. If such gyros are measured with a ratiometric A/D, then you should measure a known voltage reference to account for an apparent dependency of the sensitivity and offset of the gyro with supply voltage. Other analog gyros may provide a ratiometric referenced output, in which case you use a ratiometric A/D, then you do not need to adjust for supply voltage variation.
* Offset – 是陀螺仪没有旋转时的输出. Gyro offset varies from unit to unit and also may vary with temperature and supply voltage. Most of the offset can be removed by simply measuring the offset during power up, provided you keep the gyros motionless at that time. The variation of offset with supply voltage and temperature is usually rather slow, 所以DCM可以不断的消除offset并且保存锁定.
* Drift – 是一个随时间缓慢变化的offset和噪声. 在飞机移动时，可以通过DCM完全消除三个轴的漂移. I如果停止移动, 偏航的漂移将取决与2陀螺仪的零漂. 当它开始移动时。漂移会在几秒内被取消.
* 校准– By calibration we mean applying the correct gain multipliers to the gyro signals before applying the update algorithm. S自更新算法基本上是一个非线性积分器, 如果收益过高, 在连续转动期间，DCM将会 "over-rotate".如果太低, DCM will "under-rotate". 我们发现你可以不设置增益而准确控制DCM。 我们为了锁定漂移必须设置准确. 当反馈增益足够大时校准也会很好的工作。在一个连续的转向过程中开始累积的错误被视为一个偏移量，并且它会被补偿
* 超调 - 如果飞行器的最大旋转角速度超出了陀螺仪的最大测量范围

,陀螺仪将会饱和. This will generate an angular estimation error equal to the area between the actual rate and the saturated rate. There several practical ways of addressing this issue. The simplest solution is to use feedback gains that are large enough to erase the error in a reasonable amount of time, on the order of 10 seconds, for example. This will still retain the “smoothness” and overall accuracy of IMU control. A second solution is to include some gyro feedback in the controls to reduce the rotation rate for the axis most likely to saturate. A third solution is to implement the full version of Mahony’s approach, and to integrate estimation with control, with a constraint applied to the desired turning rates.

# Wind

必须考虑风的影响, 主要是对偏航有影响.由于我们要使用航向锁,方向余弦矩阵的轴向最终将会和飞机方向对其, 而不是所指方向. This is fine when the plane is headed in a straight course back to RTL, or between way points if you use DCM for an autopilot. What will happen in that case is that the 该算法将会把风作为漂移, and gradually rotate the plane by the angle that is required to maintain the desired course over ground. However, when the plane makes a turn, it will take a finite amount of time for DCM to adapt to the new angle between the wind direction and the course over ground. There are two 问题的解决方案:

• 使用足够大的反馈增益 to adapt to the wind within a few seconds after a change in wind or in course. This is the approach that we are presently using in our firmware. • Somehow compute the wind vector. In principle this should be possible to do, given the low residual gyro offset, provided the plane makes some turns. We are presently looking into this approach, to see if it will work better than the above approach. For now, we will continue to adapt to the wind after a turn, because that is actually working rather well.

# Using DCM in control and navigation

在上一节中我们介绍了控制和导航方向余弦的若干应用。在这一节中，我们提供了更多的细节：:

1. 对于空中的飞行器来说，只有在得知了正确的俯仰角信息之后，才能准确控制其俯仰动作。

控制飞机的俯仰，你需要了解飞机的俯仰姿态，你可以通过把飞机的滚动轴的与地面的垂线做点积。

点积就是方向余弦的, *rzx* . 它等于 横滚轴和 地球坐标系上飞机的投影的正弦角. 矩阵的元素是一个直接指示，无论飞机的滚动轴是不是平行于对面, pitch的反馈控制回路能够被直接使用. 当飞机水平时, *rzx* =0

1. To control the roll of an aircraft, you need to know the bank attitude of the aircraft, which you can find by taking the dot product of the pitch axis of the aircraft with the ground vertical.

飞机与地面的垂线（Z）与俯仰轴(Y)的点积为：, *rzy* . 它等于 俯仰轴和 地球坐标系上飞机的投影的正弦角. So that element of the matrix is a direct indication of whether or not the pitch axis of the plane is parallel to the ground, and can be used directly in a feedback loop to control pitch. 当飞机是水平的, *rzy*=0.

1. To navigate, you need to know the yaw attitude of the aircraft with respect to the direction that you want to go, 你可以通过飞机的横滚轴和你想要去的方向做点积得到。 这种方式你可能得到相反到的方向. To find out if your aircraft might be pointing in the opposite direction than you want to go, take the dot product of the roll axis with the desired direction vector. 如果是负的，飞机处于不规则运动中

飞机的滚动轴矢量是矩阵的第一列。我们只需要水平组件，所以我们设置了零组件为零. 由此产生的向量等于= [*rxx ryx* 0] . We take the cross product of that vector with the desired direction vector to get the sine of the deviation angle, and we take the dot product to get the cosine.

1. To find out if the aircraft is upside down, examine the sign of the dot product of the aircraft yaw axis with the vertical. If it is less than zero, the aircraft is upside down.

飞机偏航轴和垂线的点积是矩阵元 *rzz* . When the plane is flying more or less level, this element is approximately one. When the plane is upside down, this element is approximately minus one. 当飞机倾斜的侧面呈90度角，这个元素是零。.

1. To find out the turning rate of the aircraft around the vertical earth axis, transform the gyro rotation vector to the earth frame of reference, and take the dot product with the vertical axis.

这相当于用陀螺旋转矢量的第三行 So, 飞机绕垂直轴旋转的速度等于

# 实现

We are planning to write a separate paper on how to implement the DCM algorithm in C code. Some readers may have access to firmware that we have written. If so, to avoid confusion, you should be aware of the following: *This paper expresses all quantities using an aviation convention for the 3 axes. X is forward, Y points along the right wing, and Z is down. However, the firmware that we have written uses a coordinate system that is somewhat different. X points along the left wing, Y is forward, and Z is down. The reason that we did this is because in the design of the board, the three axis accelerometer was mounted in such a way that when you mount the board in a plane with the longer dimension aligned in the most convenient orientation, the accelerometer, and the labels on the board, point the Y axis forward.*

**参考**

Several papers written by Robert Mahony et al are available at http://gentlenav.googlecode.com/files/MahonyPapers.zip.