# 加速度计和陀螺仪指南——数学模型和基本算法

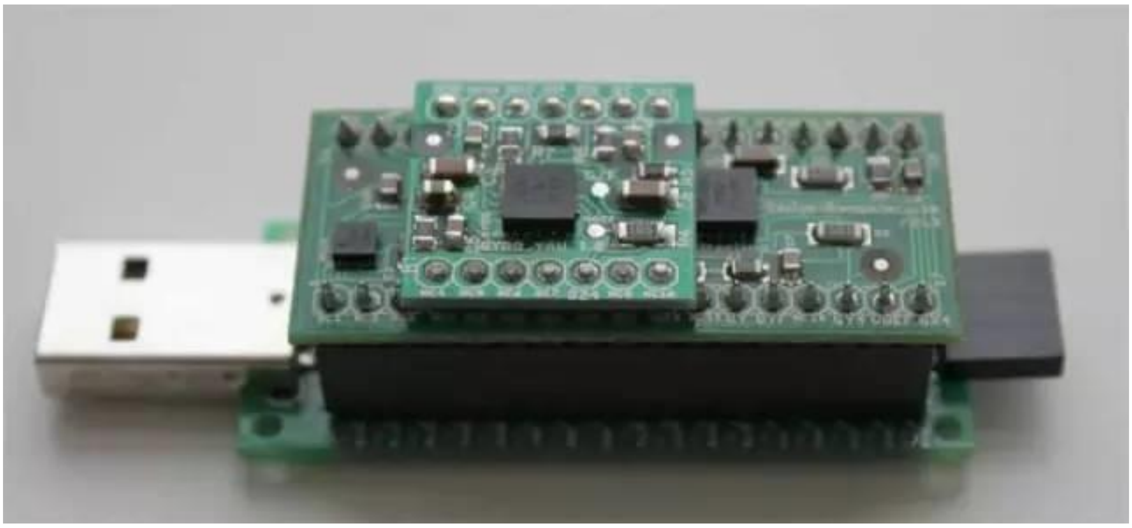
本帖转自 http://www.geek-workshop.com/thread-1695-1-1.html

本帖翻译自 [IMU（加速度计和陀螺仪设备）在嵌入式应用中使用的指南。]()

这篇文章主要介绍加速度计和陀螺仪的数学模型和基本算法，以及如何融合这两者，侧重算法、思想的讨论.

介绍

本指南旨在向兴趣者介绍惯性 MEMS（微机电系统）传感器，特别是加速度计和陀螺仪以及其他整合 IMU（惯性测量单元）设备。



IMU 单元例子：上图中 MCU 顶端的 ACC Gyro 6DOF，名为 USBThumb，支持 USB/串口通信

在这篇文章中我将概括这么几个基本并且重要的话题：

- 加速度计（accelerometer）检测什么

- 陀螺仪（gyroscope，也称作 gyro）检测什么

- 如何将传感器 ADC 读取的数据转换为物理单位（加速度传感器的单位是 g，陀螺仪的单位是 度/秒）

- 如何结合加速度传感器和陀螺仪的数据以得到设备和地平面之间的倾角的准确信息

在整篇文章中我尽量将数学运算降低到最少。如果你知道什么是正弦、余弦、正切函数，那无论你的项目使用哪种平台你应该都会明白和运用这篇文章中的思想，这些平台如 Arduino、Propeller、Basic Stamp、Ateml 芯片、PIC 芯片等等。总有些人认为使用 IMU 单元需要复杂的数学运算（复杂的 FIR 或 IIR 滤波，如卡尔曼滤波，Parks-McClellan 滤波等）。你如果研究这些会得到很棒且很复杂的结果。我解释事情的方式，只需要基本的数学。我非常坚信简单的原则。我认为一个简单的系统更容易操作和监控，另外许多嵌入式设备并不具备能力和资源去实现需要进行矩阵运算的复杂算法。

我会用我设计的一个新 IMU 模块——Acc_Gyro Accelerometer + Gyro IMU 作为例子。在下面的例子中我们会使用这个设备的参数。用这个模块作为介绍非常合适，因为它由 3 个设备组成：

- LIS331AL (datasheet) - 3 轴 2G 模拟加速度计

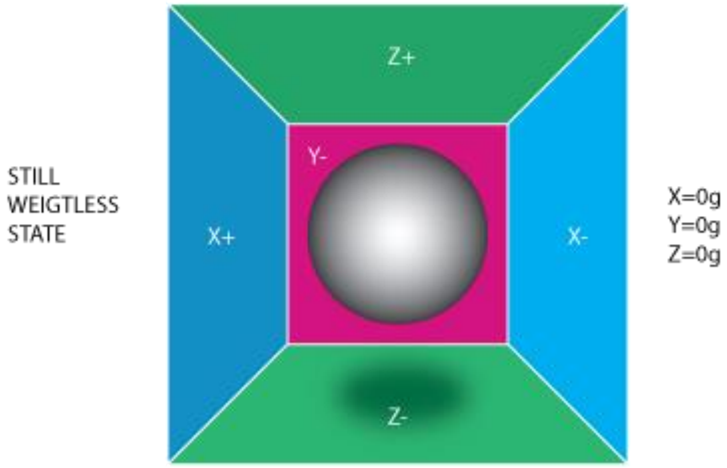
- LPR550AL (datasheet) - 双轴（俯仰、翻滚） 500°/s 加速度传感器

- LY550ALH (datasheet) - 单轴（偏航）陀螺仪 最后这个设备在这篇介绍中不使用，不过他在 DCM Matrix implementation 中有重要作用

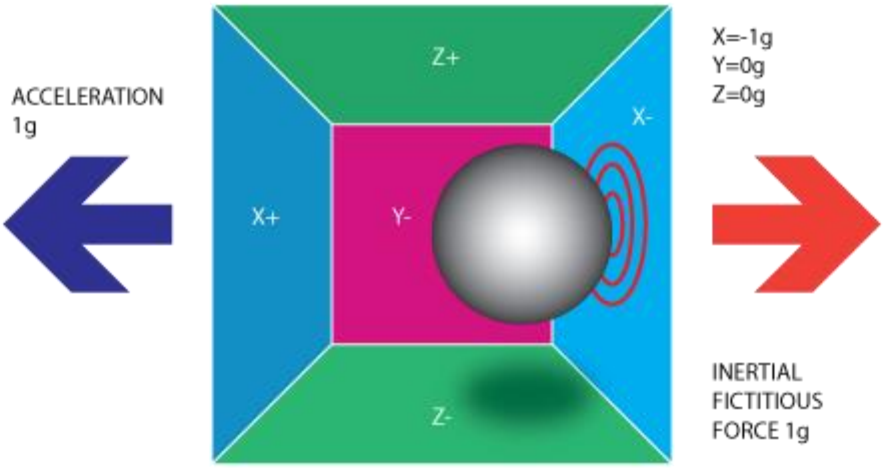
它们一起组成了一个 6 自由度的惯性测量单元。这是个花哨的名字！然而，在花哨的名字后面是个非常有用的设备组合，接下来我们会详细介绍之。

## 第一部分 加速度计

要了解这个模块我们先从加速度计开始。当我们在想象一个加速度计的时候我们可以把它想作一个圆球在一个方盒子中。你可能会把它想作一个饼干或者甜圈，但我就把它当做一个球好了：



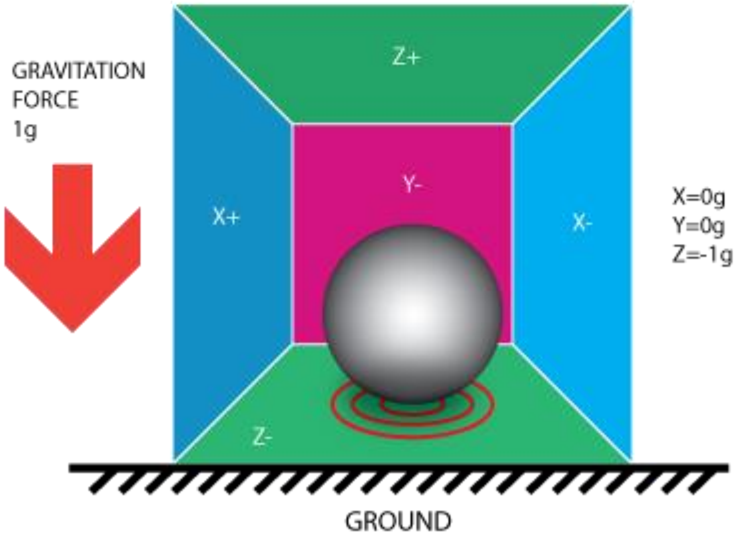

我们假定这个盒子不在重力场中或者其他任何会影响球的位置的场中，球处于盒子的正中央。你可以想象盒子在外太空中，远离任何天体，如果很难想象，那就当做盒子在航天飞机中，一切东西都处于无重力状态。在上面的图中你可以看到我们给每个轴分配了一对墙（我们移除了 Y+以此来观察里面的情况）。设想每面墙都能感测压力。如果我们突然把盒子向左移动（加速度为 1g=9.8m/s^2），那么球会撞上 X-墙。然后我们检测球撞击墙面产生的压力，X 轴输出值为-1g




请注意加速度计检测到得力的方向与它本身加速度的方向是相反的。这种力量通常被称为惯性力或假想力 。在这个模型中你你应该学到加速度计是通过间接测量力对一个墙面的作用来测量加速度的，在实际应用中，可能通过弹簧等装置来

测量力。这个力可以是加速度引起的，但在下面的例子中，我们会发现它不一定是加速度引起的。

如果我们把模型放在地球上，球会落在 Z-墙面上并对其施加一个 1g 的力，见下图：

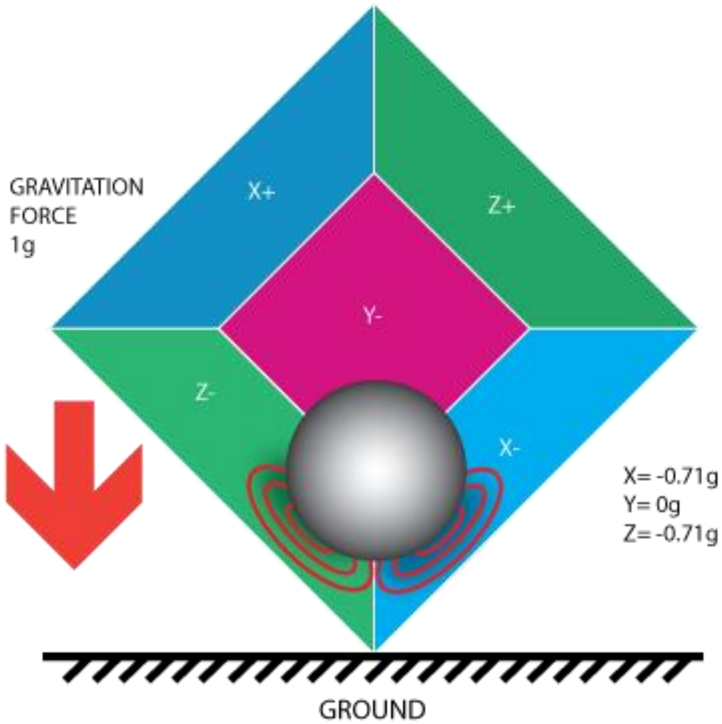



在这种情况下盒子没有移动但我们任然读取到 Z 轴有-1g 的值。球在墙壁上施加的压力是由引力造成的。在理论上，它可以是不同类型的力量 – 例如，你可以想象我们的球是铁质的，将一个磁铁放在盒子旁边那球就会撞上另一面墙。引用这个例子只是为了说明加速度计的本质是检测力而非加速度。只是加速度所引起的惯性力正好能被加速度计的检测装置所捕获。

虽然这个模型并非一个 MEMS 传感器的真实构造，但它用来解决与加速度计相关的问题相当有效。实际上有些类似传感器中有金属小球，它们称作倾角开关，但是它们的功能更弱，只能检测设备是否在一定程度内倾斜，却不能得到倾斜的程度。

到目前为止，我们已经分析了单轴的加速度计输出，这是使用单轴加速度计所能得到的。三轴加速度计的真正价值在于它们能够检测全部三个轴的惯性力。让我们回到盒子模型，并将盒子向右旋转 45 度。现在球会与两个面接触：Z-和 X-，

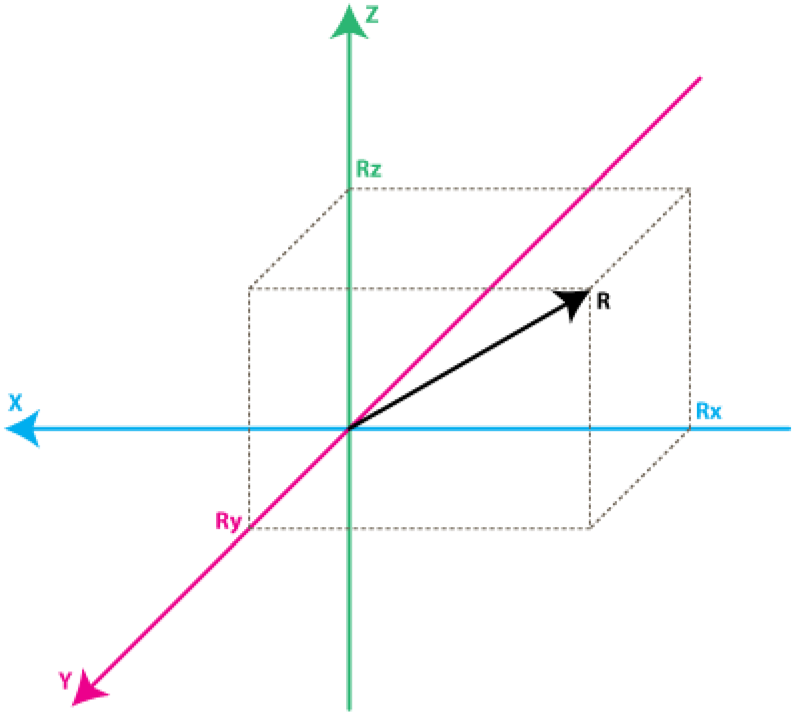
见下图：




0.71g 这个值是不是任意的，它们实际上是 1/2 的平方根的近似值。我们介绍加速度计的下一个模型时这一点会更清楚。

在上一个模型中我们引入了重力并旋转了盒子。在最后的两个例子中我们分析了盒子在两种情况下的输出值，力矢量保持不变。虽然这有助于理解加速度计是怎么和外部力相互作用的，但如果我们将坐标系换为加速度的三个轴并想象矢量力

在周围旋转，这会更方便计算。




请看看在上面的模型，我保留了轴的颜色，以便你的思维能更好的从上一个模型转到新的模型中。想象新模型中每个轴都分别垂直于原模型中各自的墙面。矢量 R 是加速度计所检测的矢量（它可能是重力或上面例子中惯性力的合成）。RX，RY，RZ 是矢量 R 在 X，Y，Z 上的投影。请注意下列关系：

，R ^ 2 = RX ^ 2 + RY ^ 2 + RZ ^ 2（公式 1）

此公式等价于三维空间勾股定理。

还记得我刚才说的 1/2 的平方根 0.71 不是个随机值吧。如果你把它们代回上式，回顾一下重力加速度是 1g，那我们就能验证：

1 ^ 2 =（SQRT（1/2））^ 2 + 0 ^ 2 +（SQRT（1/2））^ 2

在公式 1 中简单的取代： R=1, Rx = -SQRT(1/2), Ry = 0 , Rz = -SQRT(1/2)

经过一大段的理论序言后，我们和实际的加速度计很靠近了。RX，RY，RZ 值是实际中加速度计输出的线性相关值，你可以用它们进行各种计算。

在我们运用它之前我们先讨论一点获取加速度计数据的方法。大多数加速度计可归为两类：数字和模拟。数字加速度计可通过 I2C，SPI 或 USART 方式获取信息，而模拟加速度计的输出是一个在预定范围内的电压值，你需要用 ADC（模拟量转数字量）模块将其转换为数字值。我将不会详细介绍 ADC 是怎么工作的，部分原因是这是个很广的话题，另一个原因是不同平台的 ADC 都会有差别。有些 MCU 具有内置 ADC 模块，而有些则需要外部电路进行 ADC 转换。不管使用什么类型的 ADC 模块，你都会得到一个在一定范围内的数值。例如一个 10 位 ADC 模块的输出值范围在 0 .. 1023 间，请注意，1023 = 2 ^ 10 -1。一个 12 位 ADC 模块的输出值范围在 0 .. 4095 内，注意，4095 = 2 ^ 12-1。

我们继续，先考虑下一个简单的例子，假设我们从 10 位 ADC 模块得到了以下的三个轴的数据：

AdcRx = 586

AdcRy = 630

AdcRz = 561

每个 ADC 模块都有一个参考电压，假设在我们的例子中，它是 3.3V。要将一个 10 位的 ADC 值转成电压值，我们使用下列公式：

VoltsRx = AdcRx * VREF / 1023

小注：8 位 ADC 的最大值是 255 = 2 ^ 8 -1，12 位 ADC 最大值是 4095 = 2 ^ 12 -1。

将 3 个轴的值代入上式，得到：

VoltsRx = 586 * 3.3 / 1023 =～1.89V（结果取两位小数）

VoltsRy = 630 * 3.3 / 1023 =～2.03V

VoltsRz = 561 * 3.3 / 1023 =～1.81V

每个加速度计都有一个零加速度的电压值，你可以在它的说明书中找到，这个电压值对应于加速度为 0g。通过计算相对 0g 电压的偏移量我们可以得到一个有符号的电压值。比方说，0g 电压值 VzeroG= 1.65V，通过下面的方式可以得到相对 0g 电压的偏移量:

DeltaVoltsRx = 1.89V - 1.65V = 0.24V

DeltaVoltsRy = 2.03V - 1.65V = 0.38V

DeltaVoltsRz = 1.81V - 1.65V = 0.16V

现在我们得到了加速度计的电压值，但它的单位还不是 g（9.8m/s^2），最后的转换，我们还需要引入加速度计的灵敏度（Sensitivity），单位通常是 mV/g。比方说，加速度计的灵敏度 Sensitivity= 478.5mV / g = 0.4785V /g。灵敏度值可以在加速度计说明书中找到。要获得最后的单位为 g 的加速度，我们使用下列公式计算:

RX = DeltaVoltsRx /Sensitivity

RX = 0.24V / 0.4785V / G =～0.5g

RY = 0.38V / 0.4785V / G =～0.79g

RZ = 0.16V / 0.4785V / G =～0.33g

当然，我们可以把所有的步骤全部放在一个式子里，但我想通过介绍每一个步骤以便让你了解怎么读取一个 ADC 值并将其转换为单位为 g 的矢量力的分量。

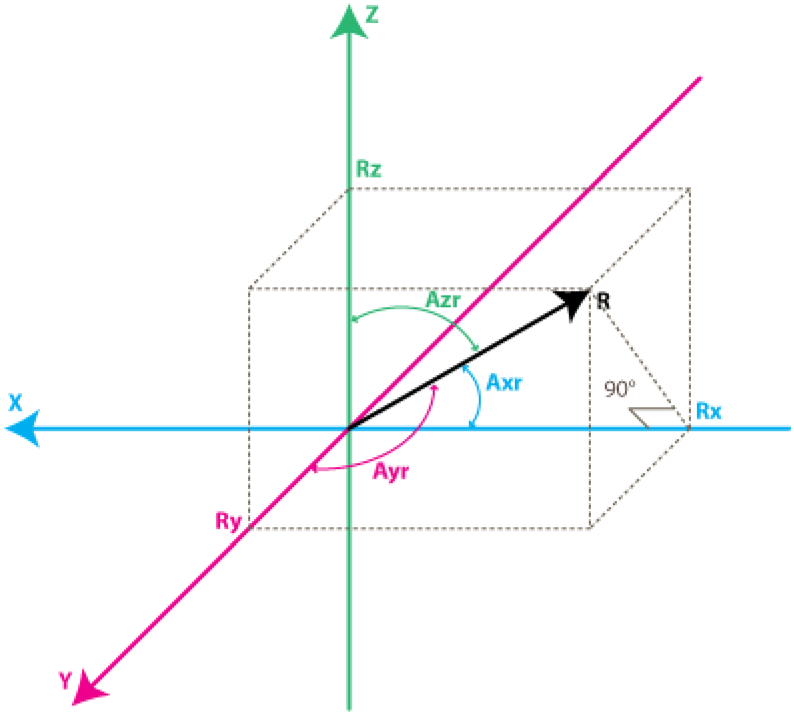
Rx = (AdcRx * Vref / 1023 - VzeroG) / Sensitivity （公式 2）

Ry = (AdcRy * Vref / 1023 - VzeroG) / Sensitivity

Rz = (AdcRz * Vref / 1023 - VzeroG) / Sensitivity

现在我们得到了惯性力矢量的三个分量，如果设备除了重力外不受任何外力影响，那我们就可以认为这个方向就是重力矢量的方向。如果你想计算设备相对于地面

的倾角，可以计算这个矢量和 Z 轴之间的夹角。如果你对每个轴的倾角都感兴趣，你可以把这个结果分为两个分量：X 轴、Y 轴倾角，这可以通过计算重力矢量和 X、Y 轴的夹角得到。计算这些角度比你想象的简单，现在我们已经算出了 Rx，Ry，Rz 的值，让我们回到我们的上一个加速度模型，再加一些标注上去：




我们感兴趣的角度是向量 R 和 X，Y，Z 轴之间的夹角，那就令这些角度为 Axr，Ayr，Azr。观察由 R 和 Rx 组成的直角三角形：

cos(Axr) = Rx / R ， 类似的：

cos(Ayr) = Ry / R

cos(Azr) = Rz / R

从公式 1 我们可以推导出 R = SQRT( Rx^2 + Ry^2 + Rz^2)

通过 arccos()函数（cos()的反函数）我们可以计算出所需的角度：

Axr = arccos(Rx/R)

Ayr = arccos(Ry/R)

Azr = arccos(Rz/R)

我们花了大段的篇幅来解释加速度计模型，最后所要的只是以上这几个公式。根据你的应用场合，你可能会用到我们推导出来的几个过渡公式。我们接下来要介绍陀螺仪模块，并向大家介绍怎么融合加速度计和陀螺仪的数据以得到更精确的倾角值。

但在此之前，我们再介绍几个很常用的公式：

cosX = cos(Axr) = Rx / R

cosY = cos(Ayr) = Ry / R

cosZ = cos(Azr) = Rz / R

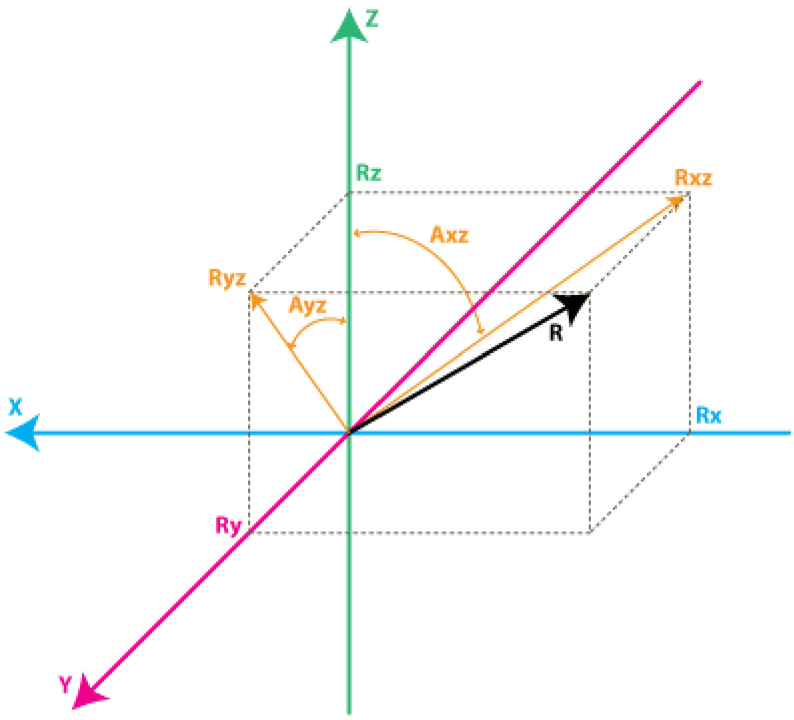
这三个公式通常称作方向余弦 ，它主要表达了单位向量（长度为 1 的向量）和 R 向量具有相同的方向。你可以很容易地验证：

SQRT（cosX ^ 2 + COSY ^ 2 + cosZ ^ 2）= 1

这是个很好的性质，因为它避免了我们一直检测 R 向量的模（长度）。通常如果我们只是对惯性力的方向感兴趣，那标准化模长以简化其他计算是个明智的选择。

**第二部分陀螺仪**

对于陀螺仪我们将不会像加速度计一样介绍它的等价盒子模型，而是直接跳到加速度计的第二个模型，通过这个模型我们会向大家介绍陀螺仪是怎么工作的。




陀螺仪的每个通道检测一个轴的旋转。例如，一个 2 轴陀螺仪检测绕 X 和 Y 轴的旋转。为了用数字来表达这些旋转，我们先引进一些符号。首先我们定义：

Rxz – 惯性力矢量 R 在 XZ 平面上的投影

Ryz – 惯性力矢量 R 在 YZ 平面的上投影

在由 Rxz 和 Rz 组成的直角三角形中，运用勾股定理可得：

Rxz^2 = Rx^2 + Rz^2 ，同样：

Ryz^2 = Ry^2 + Rz^2

同时注意：

R^2 = Rxz^2 + Ry^2 ，这个公式可以公式 1 和上面的公式推导出来，也可由 R 和 Ryz 所组成的直角三角形推导出来

R ^ 2 = Ryz ^ 2 + RX ^ 2

在这篇文章中我们不会用到这些公式，但知道模型中的那些数值间的关系有助于理解。

相反，我们按如下方法定义 Z 轴和 Rxz、Ryz 向量所成的夹角：

AXZ - Rxz（矢量 R 在 XZ 平面的投影）和 Z 轴所成的夹角

AYZ - Ryz（矢量 R 在 YZ 平面的投影）和 Z 轴所成夹角

现在我们离陀螺仪要测量的东西又近了一步。陀螺仪测量上面定义的角度的变化率。换句话说，它会输出一个与上面这些角度变化率线性相关的值。为了解释这一点，我们先假设在 t0 时刻，我们已测得绕 Y 轴旋转的角度（也就是 Axz），定义为 Axz0，之后在 t1 时刻我们再次测量这个角度，得到 Axz1。

角度变化率按下面方法计算：

RateAxz = (Axz1 - Axz0) / (t1 - t0).

如果用度来表示角度，秒来表示时间，那这个值的单位就是 度/秒。这就是陀螺仪检测的东西。

在实际运用中，陀螺仪一般都不会直接给你一个单位为度/秒的值（除非它是个特殊的数字陀螺仪）。就像加速度计一样，你会得到一个 ADC 值并且要用类似公式 2 的式子将其转换成单位为 度/秒的值。让我们来介绍陀螺仪输出值转换中的 ADC 部分（假设使用 10 位 ADC 模块，如果是 8 位 ADC，用 1023 代替 255，如果是 12 为 ADC 用 4095 代替 1023）。

RateAxz = (AdcGyroXZ * Vref / 1023 - VzeroRate) / Sensitivity 公式 3

RateAyz = (AdcGyroYZ * Vref / 1023 - VzeroRate) / Sensitivity

AdcGyroXZ，AdcGyroYZ - 这两个值由 ADC 读取，它们分别代表矢量 R 的投影在 XZ 和 YZ 平面内里的转角，也可等价的说，旋转可分解为单独绕 Y 和 X 轴的运动。

Vref - ADC 的参考电压，上例中我们使用 3.3V

VzeroRate - 是零变化率电压，换句话说它是陀螺仪不受任何转动影响时的输出值，对 Acc Gyro 板来说，可以认为是 1.23V（此值通常可以在说明书中找到

——但千万别相信这个值，因为大多数的陀螺仪在焊接后会有一定的偏差，所以可以使用电压计测量每个通道的输出值，通常这个值在焊接后就不会改变，如果有跳动，在设备使用前写一个校准程序对其进行测量，用户应当在设备启动的时候保持设备静止以进行校准）。

Sensitivity – 陀螺仪的灵敏度，单位 mV/(deg/s)，通常写作 mV/deg/s，它的意思就是如果旋转速度增加 1°/s，陀螺仪的输出就会增加多少 mV。Acc_Gyro 板的灵敏度值是 2mV/deg/s 或 0.002V/deg/s

让我们举个例子，假设我们的 ADC 模块返回以下值：

AdcGyroXZ = 571

AdcGyroXZ = 323

用上面的公式，在代入 Acc Gyro 板的参数，可得：

RateAxz = (571 * 3.3V / 1023 - 1.23V) / ( 0.002V/deg/s) =~ 306 deg/s

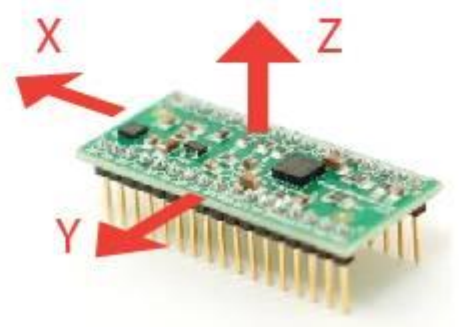
RateAyz = (323 * 3.3V / 1023 - 1.23V) / ( 0.002V/deg/s) =~ -94 deg/s

换句话说设备绕 Y 轴（也可以说在 XZ 平面内）以 306°/s 速度和绕 X 轴（或者说 YZ 平面内）以-94°/s 的速度旋转。请注意，负号表示该设备朝着反方向旋转。按照惯例，一个方向的旋转是正值。一份好的陀螺仪说明书会告诉你哪个方向是正的，否则你就要自己测试出哪个旋转方向会使得输出脚电压增加。最好使用示波器进行测试，因为一旦你停止了旋转，电压就会掉回零速率水平。如果你使用的是万用表，你得保持一定的旋转速度几秒钟并同时比较电压值和零速率电压值。如果值大于零速率电压值那说明这个旋转方向是正向。

## 第三部分 将它们综合起来。融合加速度计和陀螺仪的数据

如果你在阅读这篇文章你可能已经有了或准备购买一个 IMU 设备，或者你准备用独立的加速度计和陀螺仪搭建一个。

在使用整合了加速度计和陀螺仪的 IMU 设备时，首先要做的就是统一它们的坐标系。最简单的办法就是将加速度计作为参考坐标系。大多数的加速度计技术说明书都会指出对应于物理芯片或设备的 XZY 轴方向。例如，下面就是 Acc Gyro 板




接下来的步骤是：

- 确定陀螺仪的输出对应到上述讨论的 RateAxz，RateAyz 值。

- 根据陀螺仪和加速度计的位置决定是否要反转输出值

不要设想陀螺仪陀的输出有 XY，它会适应加速度计坐标系里的任何轴，尽管这个输出是 IMU 模块的一部分。最好的办法就是测试。

接下来的示例用来确定哪个陀螺仪的输出对应 RateAxz。

- 首先将设备保持水平。加速度计的 XY 轴输出会是零加速度电压（Acc Gyro 板的值是 1.65V）

- 接下来将设备绕 Y 轴旋转，换句话说就是将设备在 XZ 平面内旋转，所以 X、Z 的加速度输出值会变化而 Y 轴保持不变。

-当以匀速旋转设备的时候，注意陀螺仪的哪个通道输出值变化了，其他输出应该保持不变。

- 在陀螺仪绕 Y 轴旋转(在 XZ 平面内旋转)的时候输出值变化的就是 AdcGyroXZ，用于计算 RateAxz

-最后一步，确认旋转的方向是否和我们的模型对应，因为陀螺仪和加速度的位置关系，有时候你可能要把 RateAxz 值反向

-重复上面的测试，将设备绕 Y 轴旋转，这次查看加速度计的 X 轴输出（也就是 AdcRx）。如果 AdcRx 增大（从水平位置开始旋转的第一个 90°），那 AdcGyroXZ 应当减小。这是因为我们观察的是重力矢量，当设备朝一个方向旋转时矢量会朝相反的方向旋转（相对坐标系运动）。所以，如果你不想反转 RateAxz，你可以在公式 3 中引入正负号来解决这个问题：

RateAxz = InvertAxz * (AdcGyroXZ * Vref / 1023 - VzeroRate) / Sensitivity , 其中 InvertAxz= 1 或-1

同样的方法可以用来测试 RateAyz，将设备绕 X 轴旋转，你就能测出陀螺仪的哪个输出对应于 RateAyz，以及它是否需要反转。一旦你确定了 InvertAyz，你就能可以用下面的公式来计算 RateAyz：

RateAyz = InvertAyz * (AdcGyroYZ * Vref / 1023 - VzeroRate) / Sensitivity

如果对 Acc Gyro 板进行这些测试，你会得到下面的这些结果：

- RateAxz 的输出管脚是 GX4，InvertAxz = 1

- RateAyz 输出管脚是 GY4，InvertAyz = 1

从现在开始我们认为你已经设置好了 IMU 模块并能计算出正确的 Axr，Ayr，Azr 值（在第一部分加速度计中定义）以及 RateAyz，RateAyz（在第二部分陀螺仪中）。下一步，我们分析这些值之间的关系并得到更准确的设备和地平面之间的倾角。

你可能会问自己一个问题，如果加速度计已经告诉我们 Axr，Ayr，Azr 的倾角，为什么还要费事去得到陀螺仪的数据？答案很简单：加速度计的数据不是 100%准确的。有几个原因，还记加速度计测量的是惯性力，这个力可以由重力引起（理想情况只受重力影响），当也可能由设备的加速度（运动）引起。因此，就算加速度计处于一个相对比较平稳的状态，它对一般的震动和机械噪声很敏感。这就是为什么大部分的 IMU 系统都需要陀螺仪来使加速度计的输出更平滑。但是怎么办到这点呢？陀螺仪不受噪声影响吗？

陀螺仪也会有噪声，但由于它检测的是旋转，因此对线性机械运动没那么敏感，不过陀螺仪有另外一种问题，比如漂移（当选择停止的时候电压不会回到零速率电压）。然而，通过计算加速度计和陀螺仪的平均值我们能得到一个相对更准确的当前设备的倾角值，这比单独使用加速度计更好。

接下来的步骤我会介绍一种算法，算法受卡尔曼滤波中的一些思想启发，但是它更简单并且更容易在嵌入式设备中实现。在此之前，让我们先看看我们需要算法计算什么值。所要算的就是重力矢量 R=[Rx,Ry,Rz]，它可由其他值推导出来，如 Axr，Ayr，Azr 或者 cosX，cosY，cosZ，由这些值我们能得到设备相对地平面的倾角值，这些关系我们在第一部分已经讨论论过。有人可能会说-根据第一部分的公式 2 我们不是已经得到 Rx，Ry，Rz 的值了吗？是的，但是记住，这些值只是由加速度计数据推导出来的，如果你直接将它们用于你的程序你会得到难以忍受的噪声。为了避免进一步的混乱，我们重新定义加速度计的测量值：

Racc – 是由加速度计测量到得惯性力矢量，它可分解为下面的分量（在 XYZ 轴上的投影）：

RxAcc = (AdcRx * Vref / 1023 – VzeroG) / Sensitivity
RyAcc = (AdcRy * Vref / 1023 – VzeroG) / Sensitivity

RzAcc = (AdcRz * Vref / 1023 – VzeroG) / Sensitivity

现在我们得到了一组只来自于加速度计 ADC 的值。我们把这组数据叫做“vector”，并使用下面的符号：

Racc = [RxAcc,RyAcc,RzAcc]

因为这些 Racc 的分量可由加速度计数据得到，我们可以把它当做算法的输入。

请注意Racc测量的是重力,如果你得到的矢量长度约等于1g那么你就是正确的：

|Racc| = SQRT(RxAcc^2 +RyAcc^2 + RzAcc^2),

但是请确定把矢量转换成下面的矢量非常重要：

Racc(normalized) = [RxAcc/|Racc| , RyAcc/|Racc| , RzAcc/|Racc|].

这可以确保标准化 Racc 始终是 1。

接来下我们引进一个新的向量：

Rest = [RxEst,RyEst,RzEst]

这就是算法的输出值，它经过陀螺仪数据的修正和基于上一次估算的值。

这是算法所做的事：

-加速度计告诉我们：“你现在的位置是 Racc”

我们回答：“谢谢，但让我确认一下”

-然后根据陀螺仪的数据和上一次的Rest值修正这个值并输出新的估算值Rest。

-我们认为 Rest 是当前设备姿态的“最佳值”。

让我们看看它是怎么实现的。

数列的开始，我们先认为加速度值正确并赋值：

Rest(0) = Racc(0)

Rest 和 Racc 是向量，所以上面的式子可以用 3 个简单的式子代替，注意别重复了：

RxEst（0）= RxAcc（0）

RyEst（0）= RyAcc（0）

RzEst（0）= RzAcc（0）

接下来我们在每个等时间间隔 T 秒做一次测量，得到新的测量值，并定义为 Racc（1），Racc（2），Racc（3）等等。同时，在每个时间间隔我们也计算出新的估算值 Rest（1）,Rest（2），Rest（3），等等。

假设我们在第 n 步。我们有两列已知的值可以用：

Rest(n-1) - 前一个估算值，Rest(0) = Racc(0)

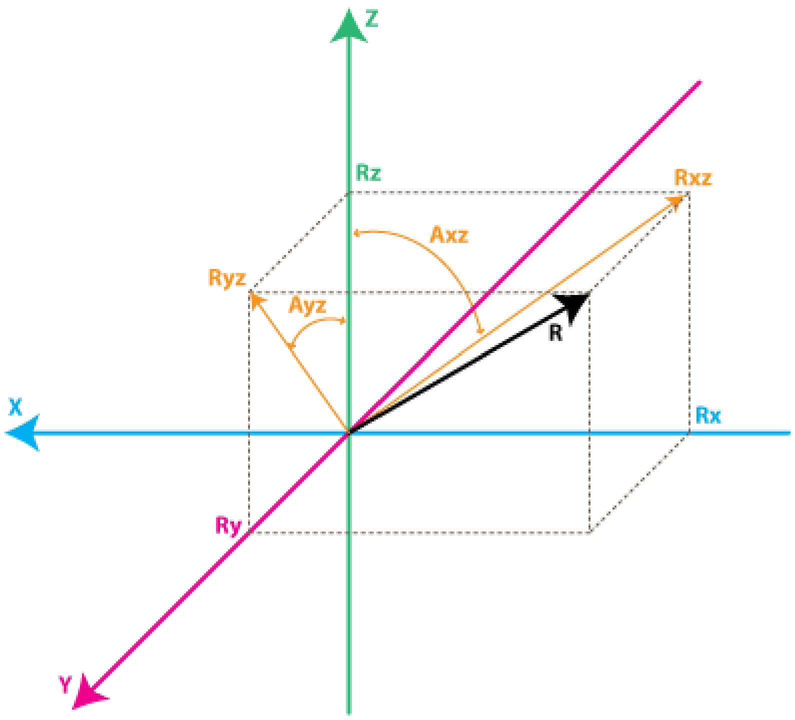
Racc(n) - 当前加速度计测量值

在计算 Rest（n）前，我们先引进一个新的值，它可由陀螺仪和前一个估算值得到。

叫做 Rgyro，同样它是个矢量并由 3 个分量组成：

Rgyro = [RxGyro,RyGyro,RzGyro]

我们分别计算这个矢量的分量，从 RxGyro 开始。




首先观察陀螺仪模型中下面的关系，根据由 Rz 和 Rxz 组成的直角三角形我们能推出：

tan(Axz) = Rx/Rz => Axz = atan2(Rx,Rz)

你可能从未用过 atan2 这个函数，它和 atan 类似，但 atan 返回值范围是（-PI/2,PI/2），atan2 返回值范围是（-PI，PI），并且他有两个参数。它能将 Rx，Rz 值转换成 360°（-PI，PI）内的角度。

所以，知道了 RxEst（n-1）和 RzEst（n-1）我们发现：

Axz(n-1) = atan2( RxEst(n-1) , RzEst(n-1) ).

记住，陀螺仪测量的是 Axz 角度变化率，因此，我们可以按如下方法估算新的角度 Axz（n）：

Axz(n) = Axz(n-1) + RateAxz(n) * T

请记住，RateAxz 可由陀螺仪 ADC 读取得到。通过使用平均转速可由得到一个更准确的公式：

$$RateAxzAvg = ( RateAxz(N) + RateAxz(N-1) ) / 2$$

$$Axz(n) = Axz(n-1) + RateAxzAvg * T$$

同理可得：

$$Ayz(n) = Ayz(n-1) + RateAyz(n) * T$$

好了，现在我们有了 Axz（n），Ayz（n）。现在我们如何推导出 RxGyro/RyGyro? 根据公式 1 我们可以把 Rgyro 长度写成下式：

$$| Rgyro | = SQRT( RxGyro ^ 2 + RyGyro ^ 2 + RzGyro ^ 2)$$

同时，因为我们已经将 Racc 标准化，我们可以认为它的长度是 1 并且旋转后保持不变，所以写成下面的方式相对比较安全：

$$| Rgyro | = 1$$

我们暂时采用更短的符号进行下面的计算：

$x = RxGyro$ ， $y=RyGyro$, $z=RzGyro$

根据上面的关系可得：

$$x = x / 1 = x / SQRT(x^2+y^2+z^2)$$

分子分母同除以 SQRT（X ^ 2 + Z ^ 2）

$$x = ( x / SQRT(x^2 + z^2) ) / SQRT( (x^2 + y^2 + z^2) / (x^2 + z^2) )$$

注意 $x / SQRT(x^2 + z^2) = sin(Axz)$, 所以:

$$x = sin(Axz) / SQRT (1 + y^2 / (x^2 + z^2) )$$

将 SQRT 内部分式的分子分母同乘以 $z^2$

x = sin(Axz) / SQRT (1 + y^2 * z ^2 / (z^2 * (x^2 + z^2)) )

注意 z / SQRT(x^2 + z^2) = cos(Axz)， y / z = tan(Ayz), 所以最后可得:

x = sin(Axz) / SQRT (1 + cos(Axz)^2 * tan(Ayz)^2 )

替换成原来的符号可得：

RxGyro = sin(Axz(n)) / SQRT (1 + cos(Axz(n))^2 * tan(Ayz(n))^2 )

同理可得：

RyGyro = sin(Ayz(n)) / SQRT (1 + cos(Ayz(n))^2 * tan(Axz(n))^2 )

提示：这个公式还可以更进一步简化。分式两边同除以 sin（axz（你））可得：

RxGyro = 1 / SQRT (1/ sin(Axz(n))^2 + cos(Axz(n))^2 / sin(Axz(n))^2 * tan(Ayz(n))^2 )

RxGyro = 1 / SQRT (1/ sin(Axz(n))^2 + cot(Axz(n))^2 * sin(Ayz(n))^2 / cos(Ayz(n))^2 )

现在加减 cos(Axz(n))^2/sin(Axz(n))^2 = cot(Axz(n))^2

RxGyro = 1 / SQRT (1/ sin(Axz(n))^2 - cos(Axz(n))^2/sin(Axz(n))^2 + cot(Axz(n))^2 * sin(Ayz(n))^2 / cos(Ayz(n))^2 + cot(Axz(n))^2 )

综合条件 1、2 和 3、4 可得：

RxGyro = 1 / SQRT (1 + cot(Axz(n))^2 * sec(Ayz(n))^2 ), 其中 cot(x) = 1 / tan(x) ， sec(x) = 1 / cos(x)

这个公式只用了 2 个三角函数并且计算量更低。如果你有 Mathematica 程序，通过使用 FullSimplify [Sin[A]^2/ ( 1 + Cos[A]^2 * Tan[B]^2)]你可以验证这个公式。

现在我们发现：

RzGyro = Sign(RzGyro)*SQRT(1 - RxGyro^2 - RyGyro^2).

其中，当 RzGyro>=0 时，Sign(RzGyro) = 1， 当 RzGyro<0 时，Sign(RzGyro) = -1 。

一个简单的估算方法：

Sign(RzGyro) = Sign(RzEst(n-1))

在实际应用中，当心 RzEst（n-1）趋近于 0。这时候你可以跳过整个陀螺仪阶段并赋值：Rgyro=Rest（n-1）。Rz 可以用作计算 Axz 和 Ayz 倾角的参考，当它趋近于 0 时，它可能会溢出并引发不好的后果。这时你会得到很大的浮点数据，并且 tan()/atan()函数得到的结果会缺乏精度。

现在我们回顾一下已经得到的结果，我们在算法中的第 n 步，并计算出了下面的值：

Racc - 加速度计读取的当前值

Rgyro - 根据 Rest(-1)和当前陀螺仪读取值所得

我们根据哪个值来更新 Rest（n）呢？你可能已经猜到，两者都采用。我们会用一个加权平均值，得：

Rest(n) = (Racc * w1 + Rgyro * w2 ) / (w1 + w2)

分子分母同除以 w1，公式可简化成：

Rest(n) = (Racc * w1/w1 + Rgyro * w2/w1 ) / (w1/w1 + w2/w1)

令 w2=w1=wGyro，可得：

Rest(n) = (Racc + Rgyro * wGyro ) / (1 + wGyro)

在上面的公式中，wGyro 表示我们对加速度计和陀螺仪的相信程度。这个值可以

通过测试确定，根据经验值 5-20 之间会得到一个很好的结果。

此算法和卡尔曼滤波最主要的差别是它的权重是相对固定的，而卡尔曼滤波中的权重会随着加速度计读取的噪声而改变。卡尔曼滤波注重给你一个“最好”的理论结果，而此算法给你的是实际项目中“够用”的结果。你可以实现一个算法，它能根据测量的噪声而改变 wGyro 值，但对大部分应用来说固定的权重也能工作的很好。

现在得到最新的估算值还差一步：

RxEst(n) = (RxAcc + RxGyro * wGyro ) / (1 + wGyro)

RyEst(n) = (RyAcc + RyGyro * wGyro ) / (1 + wGyro)

RzEst(n) = (RzAcc + RzGyro * wGyro ) / (1 + wGyro)

现在，再次标准化矢量：

R = SQRT(RxEst(n) ^2 + RyEst(n)^2 + RzEst(n)^2 )

RxEst(n) = RxEst(n)/R

RyEst(n) = RyEst(n)/R

RzEst(n) = RzEst(n)/R

现在，可以再次进行下一轮循环了。