

# KF-GINS相关问题

邓成剑 王立强

更新时间：20230219

组合导航算法讲义、PPT及KF-GINS源码下载地址如下：

[《组合导航算法讲义》下载链接](#)

[《惯性导航原理与方法》PPT下载链接](#)

[KF-GINS源码下载链接](#)

## 目录

### KF-GINS相关问题

#### KF-GINS程序问题整理

1. 数据及生成文件格式
2. 导航坐标系与地心地固坐标系的转换
3. 由地心地固坐标系下的坐标计算大地坐标系下的纬、经、高
4. kf-gins中补偿了速度旋转效应和划桨效应、姿态圆锥效应，低端imu还是高端imu更应该补偿？
5. 重力引起的重力加速度需要去掉吗？
6. 使用组合导航需要初始对准吗？
7. 算法中误差反馈时为什么只对姿态从bp转到bn，而位置和速度没有转呢？
8. 欧拉角为小角度时的方向余弦矩阵DCM
9. KF-GINS可以达到怎样的定位精度？
10. 初始导航状态和初始导航状态的标准差怎么设定？
11. 零偏和比例因子误差建模的相关时间
12. GNSS/INS组合导航在丢失GNSS信号，纯惯导推算时，误差项可以补偿吗？
13. GNSS/INS组合导航系统，是不是不需要考虑惯导系和车体系的转换？
14. IMU数据 速率形式到增量形式的转换
15. 开源代码位置更新过程中，没有用讲义上的公式
16. 参考系统、GNSS、IMU之间存在杆臂
17. 代码中qne\_mid的计算方式是怎么得出的？
18. 比力积分项投影到n系，等效旋转矢量理论上用中间时刻，而程序里用的是k时刻

#### 讲义及PPT勘误

##### 《惯性导航原理与方法》PPT勘误：

1. 惯导误差微分方程笔误（PPT第二部分，P104）

##### 《组合导航算法讲义》勘误：

1. 四元数旋转变换部分笔误（惯导姿态算法，P18）

## KF-GINS程序问题整理

# 1. 数据及生成文件格式

文件格式参考说明文档。

# 2. 导航坐标系与地心地固坐标系的转换

```
/* n系(导航坐标系)到e系(地心地固坐标系)转换四元数 */
static Quaterniond qne(const Vector3d &blh) {

    // 传进来是一个pvapre.pos, 包括纬度, 经度, 高度
    Quaterniond quat;

    double coslon, sinlon, coslat, sinlat;

    coslon = cos(blh[1] * 0.5);
    sinlon = sin(blh[1] * 0.5);
    coslat = cos(-M_PI * 0.25 - blh[0] * 0.5);
    sinlat = sin(-M_PI * 0.25 - blh[0] * 0.5);

    quat.w() = coslat * coslon;
    quat.x() = -sinlat * sinlon;
    quat.y() = sinlat * coslon;
    quat.z() = coslat * sinlon;

    return quat;
}
```

对应代码

- 1、代码里面的n系转换到e系是指坐标变换【注意与坐标系变换矩阵区别】；
- 2、代码对应的公式是先得到方向余弦矩阵，再由方向余弦矩阵与四元数的转换关系得到：

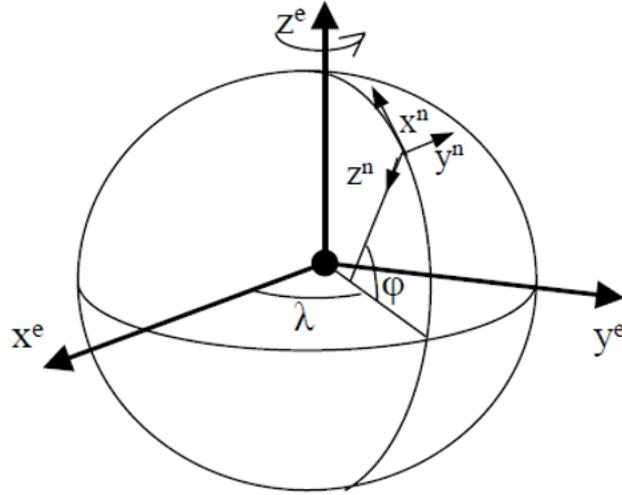


Figure 2.5: The Earth frame and the navigation frame

1) 先将e系绕e系Z轴旋转 $\lambda$ 到a系，坐标变换矩阵【注意与坐标系变换矩阵区别】如下 (纬度:  $\varphi$ , 经度:  $\lambda$ ):

$$\mathbf{C}_e^a = \begin{bmatrix} \cos \lambda & \sin \lambda & 0 \\ -\sin \lambda & \cos \lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

2) 再将a系绕a系Y轴旋转 $-(\varphi + 90^\circ)$ 与n系对齐，坐标变换矩阵【注意与坐标系变换矩阵区别】如下:

$$\mathbf{C}_a^n = \begin{bmatrix} -\sin \varphi & 0 & \cos \varphi \\ 0 & 1 & 0 \\ -\cos \varphi & 0 & -\sin \varphi \end{bmatrix} \quad (2)$$

$$\mathbf{r}^n = \mathbf{C}_a^n \mathbf{C}_e^a \mathbf{r}^e \quad (3)$$

3) n系转换到e系是指坐标变换矩阵:

$$\mathbf{C}_n^e = (\mathbf{C}_e^n)^T = (\mathbf{C}_a^n \mathbf{C}_e^a)^T = \begin{bmatrix} -\sin \varphi \cos \varphi & -\sin \lambda & -\cos \varphi \cos \lambda \\ -\sin \varphi \sin \lambda & \cos \lambda & -\cos \varphi \sin \lambda \\ \cos \varphi & 0 & -\sin \varphi \end{bmatrix} \quad (4)$$

4) 将 $\mathbf{C}_n^e$ 转化为四元素 $\mathbf{q}_n^e$  (代码对应公式):

The DCM from the n-frame to the e-frame is expressed in terms of the geodetic latitude and longitude as

$$C_n^e = \begin{bmatrix} -\sin \varphi \cos \lambda & -\sin \lambda & -\cos \varphi \cos \lambda \\ -\sin \varphi \sin \lambda & \cos \lambda & -\cos \varphi \sin \lambda \\ \cos \varphi & 0 & -\sin \varphi \end{bmatrix}. \quad (2.26)$$

The quaternion corresponding to  $C_n^e$  is written as

$$q_n^e = \begin{bmatrix} \cos(-\pi/4 - \varphi/2) \cos(\lambda/2) \\ -\sin(-\pi/4 - \varphi/2) \sin(\lambda/2) \\ \sin(-\pi/4 - \varphi/2) \cos(\lambda/2) \\ \cos(-\pi/4 - \varphi/2) \sin(\lambda/2) \end{bmatrix}. \quad (2.27)$$

图片来自文献【Estimation Techniques for Low-Cost Inertial Navigation: P23】

### 3. 由地心地固坐标系下的坐标计算大地坐标系下的纬、经、高

```

/* 地心地固坐标转地理坐标 */
static Vector3d ecef2blh(const Vector3d &ecef) {
    double p = sqrt(ecef[0] * ecef[0] + ecef[1] * ecef[1]);
    double rn;
    double lat, lon;
    double h = 0, h2;

    // 初始状态
    lat = atan(ecef[2] / (p * 1.0 - WGS84_E1));
    lon = 2.0 * atan2(ecef[1], ecef[0] + p);

    do {
        h2 = h;
        rn = RN(lat);
        h = p / cos(lat) - rn;
        lat = atan(ecef[2] / (p * (1.0 - WGS84_E1 * rn / (rn + h))));
    } while (fabs(h - h2) > 1.0e-4);

    return {lat, lon, h};
}

```

为何不用下面的公式？

Geodetic coordinates can be computed from the e-frame coordinates using the geometry shown in Figures 2.4 and 2.5. The longitude can be computed simply by

$$\lambda = \tan_2^{-1}(y, x), \quad (2.21)$$

经度计算时要考虑 $x=0$ 的情况，具体参考文献【Vermeille H. Direct transformation from geocentric coordinates to geodetic coordinates[J]. Journal of Geodesy, 2002, 76(8): 451-454.】

#### **4. kf-gins中补偿了速度旋转效应和划桨效应、姿态圆锥效应，低端imu还是高端imu更应该补偿？**

参考文献【Zhang Q, Niu X, Zhang H, et al. Algorithm improvement of the low-end GNSS/INS systems for land vehicles navigation[J]. Mathematical Problems in Engineering, 2013, 2013.】

#### **5. 重力引起的重力加速度需要去掉吗？**

算法在位置速度递推的时候补偿了重力加速度，直接使用imu测量值就可以，不需要去掉重力加速度。

#### **6. 使用组合导航需要初始对准吗？**

需要，可以设定初始状态，也可以使用其它初始对准方法。目前，开源代码只支持给定全部初始状态进行初始对准。

#### **7. 算法中误差反馈时为什么只对姿态从bp转到bn，而位置和速度没有转呢？**

位置和速度误差就是建模在n系下的，误差反馈的时候就是直接减去了。姿态误差建模的就是p系和n系的误差角，所以姿态反馈的时候补偿了p->n的旋转。

#### **8. 欧拉角为小角度时的方向余弦矩阵DCM**



- 当三个欧拉角均为小角度时（小角度近似），对应的方向余弦矩阵可简化为：

$$\mathbf{C}_b^R \approx \begin{bmatrix} 1 & -\psi & \theta \\ \psi & 1 & -\phi \\ -\theta & \phi & 1 \end{bmatrix} = \mathbf{I} + (\boldsymbol{\alpha}_{Rb} \times), \quad \boldsymbol{\alpha}_{Rb} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}$$

- 思考

$$\mathbf{C}_{b(t_{k-1})}^{b(t_k)} \approx \mathbf{I} - (\Delta\boldsymbol{\theta}_{b(t_{k-1})b(t_k)} \times)$$

注意：从哪个坐标系转动到哪个坐标系；vs 相对于

$$\mathbf{C}_{b(t_k)}^{b(t_{k-1})} \approx \mathbf{I} + (\Delta\boldsymbol{\theta}_{b(t_{k-1})b(t_k)} \times)$$

\*Tips:  $\Delta\boldsymbol{\theta}_k = \int_{t_{k-1}}^{t_k} \boldsymbol{\omega}_{ib}(t)^b dt, \quad \mathbf{C}_b^R = \begin{bmatrix} c\theta c\psi & -c\phi s\psi + s\phi s\theta c\psi & s\phi s\psi + c\phi s\theta c\psi \\ c\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & -s\phi c\psi + c\phi s\theta s\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}$

参考文献【Estimation Techniques for Low-Cost Inertial Navigation: P23-27】

## 9. KF-GINS可以达到怎样的定位精度？

定位精度跟惯导等级和测试时GNSS定位质量相关。不同等级惯导测试可以参考 Impact of the Earth Rotation Compensation on MEMS-IMU Preintegration of Factor Graph Optimization 这篇文章中的实验。此外，组合导航算法比较成熟，对于同样的测试条件，只要算法正确实现了，基本不会影响定位精度。

## 10. 初始导航状态和初始导航状态的标准差怎么设定？

初始位置利用GNSS定位结果给定，初始速度和初始姿态可以从参考结果中给定；

初始位置和速度标准差如果从GNSS获取，可以直接利用GNSS输出的标准差，初始姿态标准差可以给经验值，横滚俯仰通常小一些，航向大一些。

初始IMU误差标准差可以按照IMU误差参数给定，对于MEMS IMU，初始比例因子标准差可以给1000ppm。

## 11. 零偏和比例因子误差建模的相关时间

陀螺和加速度计零偏及比例因子误差均建模为一阶高斯马尔可夫过程。实践上，我们一般根据经验确定相关时间。对于MEMS一般取1hr，或者可以更低，如1000秒。

## 12. GNSS/INS组合导航在丢失GNSS信号，纯惯导推算时，误差项可以补偿吗？

可以用有GNSS时组合导航估计出来的传感器误差继续对IMU测量值进行补偿。

## 13. GNSS/INS组合导航系统，是不是不需要考虑惯导系和车体系的转换？

GNSS/INS组合本身不受什么载体限制，跟车体系无关；如果使用里程计和NHC约束，就需要考虑车体系。

## 14. IMU数据 速率形式到增量形式的转换

[参考我们实验室公众号文章](#)

## 15. 开源代码位置更新过程中，没有用讲义上的公式

```
void INSMech::posUpdate(const PVA &pvapre, PVA &pvacur, const IMU &imupre, const IMU &imucur) {

    Eigen::Vector3d temp1, temp2, midvel, midpos;
    Eigen::Quaterniond qne, qee, qnn;

    // 重新计算中间时刻的速度和位置
    // recompute velocity and position at k-1/2
    midvel = (pvacur.vel + pvapre.vel) / 2;
    midpos = pvapre.pos + Earth::DRi(pvapre.pos) * midvel * imucur.dt / 2;

    // 重新计算中间时刻地理参数
    // recompute rmrn, wie_n, wen_n at k-1/2
    Eigen::Vector2d rmrn;
    Eigen::Vector3d wie_n, wen_n;
    rmrn = Earth::meridianPrimeVerticalRadius(midpos[0]);
    wie_n << WGS84_WIE * cos(midpos[0]), 0, -WGS84_WIE * sin(midpos[0]);
    wen_n << midvel[1] / (rmrn[1] + midpos[2]), -midvel[0] / (rmrn[0] + midpos[2]),
            -midvel[1] * tan(midpos[0]) / (rmrn[1] + midpos[2]);

    // 重新计算 k时刻到k-1时刻 n系旋转矢量
    // recompute n-frame rotation vector (n(k) with respect to n(k-1)-frame)
    temp1 = (wie_n + wen_n) * imucur.dt;
    qnn = Rotation::rotvec2quaternion(temp1);
    // e系转动等效旋转矢量 (k-1时刻k时刻，所以取负号)
    // e-frame rotation vector (e(k-1) with respect to e(k)-frame)
    temp2 << 0, 0, -WGS84_WIE * imucur.dt;
    qee = Rotation::rotvec2quaternion(temp2);
```

讲义上的位置更新，是通过n系下速度计算出n系下相对位置变化，然后转到大地坐标的相对位置进行位置更新，这里因为两个时刻位置不一样，n系是有略微变化的，然而这里相当于忽略了这个变化。代码中四元数形式更新的时候，是考虑了n系的变化。中间时刻位置计算的时候，这里中间时刻位置其实也是线性假设下得到的，本身假设也是有一些误差，而且中间时刻位置影响也不大，所以就用讲义上的方法写的，比较简洁。

## 16. 参考系统、GNSS、IMU之间存在杆臂

当忽略参考系统与被测系统的杆臂时，位置误差曲线会存在明显的杆臂误差。

## 17. 代码中qne\_mid的计算方式是怎么得出的？

```

insmech.cpp > attUpdate(const PVA &, PVA &, const IMU &, const IMU &)
INSMech::attUpdate(const PVA &pvapre, PVA &pvacur, const IMU &imupre, const
Eigen::Quaterniond qne_pre, qne_cur, qne_mid, qnn, qbb, qne_test;
Eigen::Vector3d temp1, midpos, midvel;

// 重新计算中间时刻的速度和位置
// recompute velocity and position at k-1/2
midvel = (pvapre.vel + pvacur.vel) / 2;
qne_pre = Earth::qne(pvapre.pos);
qne_cur = Earth::qne(pvacur.pos);
qne_test = Earth::qne((pvacur.pos+pvapre.pos)/2.0);
temp1 = Rotation::quaternion2vector(qne_cur.inverse() * qne_pre);
qne_mid = qne_pre * Rotation::rotvec2quaternion(temp1 / 2).inverse();
midpos[2] = (pvacur.pos[2] + pvapre.pos[2]) / 2;
midpos = Earth::blh(qne_mid, midpos[2]);

// 重新计算中间时刻地理参数
// recompute rmrn, wie_n, wen_n at k-1/2
Eigen::Vector2d rmrn;
    
```

这里qne\_mid计算的时候，用旋转矢量表示k时刻和k-1两个时刻之间n系的变化，然后取旋转矢量的一半认为是k-1到k-1/2时刻的变化。直接位置求平均也是可以表示k-1/2时刻的位置。因为车载动态比较小，而且IMU采样率高，所以两种方式应该没啥差别的。

## 18. 比力积分项投影到n系，等效旋转矢量理论上用中间时刻，而程序里用的是k时刻

```

// 比力积分项投影到n系
// velocity increment due to the specific force
temp1 = (wie_n + wen_n) * imucur.dt / 2;
cnn = I33 - Rotation::skewSymmetric(temp1);
d_vfn = cnn * pvapre.att.cbn * d_vfb;
        
```

□ 等效旋转矢量  $\zeta_{n(k-1)n(k)}$

$$\zeta_{n(k-1)n(k)} \approx \int_{t_{k-1}}^{t_k} \omega_{n(k-1)n(t)}^n dt = \int_{t_{k-1}}^{t_k} \omega_{in}^n(t) dt$$

$$\approx (\omega_{ie}^n|_{k-1/2} + \omega_{en}^n|_{k-1/2}) \Delta t$$

程序里面是有外推的过程的，利用外推计算得到k-1/2时刻的状态。这两个量都是跟n系变化相关的量，在IMU连续历元间n系位置变化很小，可以认为这两个量是不变的，所以用k时刻或k-1时刻均可。

## 讲义及PPT勘误

### 《惯性导航原理与方法》PPT勘误：

#### 1. 惯导误差微分方程笔误（PPT第二部分，P104）

惯导误差微分方程中噪声项矩阵中，因为笔误，加速度计噪声“ $\omega_a$ ”和陀螺仪噪声“ $\omega_g$ ”写反了。

□ 将传感器误差模型带入惯导误差微分方程，得

$$\begin{bmatrix} \delta \dot{\mathbf{r}}^n \\ \delta \dot{\mathbf{v}}^n \\ \dot{\phi} \\ \dot{b}_g \\ \dot{b}_a \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{rr} \mathbf{I}_{3 \times 3} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{vr} & \mathbf{F}_{vv} & [(\mathbf{C}_b^n \mathbf{f}^b) \times] & \mathbf{0} & \mathbf{C}_b^n \\ \mathbf{F}_{\phi r} & \mathbf{F}_{\phi v} & -(\boldsymbol{\omega}_{in}^n \times) & -\mathbf{C}_b^n & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \frac{-1}{\tau_{gb}} \mathbf{I}_{3 \times 3} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \frac{-1}{\tau_{ab}} \mathbf{I}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \delta \mathbf{r}^n \\ \delta \mathbf{v}^n \\ \phi \\ b_g \\ b_a \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_b^n & \mathbf{0} & \mathbf{0} \\ -\mathbf{C}_b^n & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{3 \times 3} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \mathbf{w}_a \\ \mathbf{w}_g \\ \mathbf{w}_{gb} \\ \mathbf{w}_{ab} \end{bmatrix}$$

© Dr. Xiaoji Niu, Nav. Group, WHU

104

图片红色框处的部分应为： $[\boldsymbol{\omega}_g \quad \boldsymbol{\omega}_a \quad \boldsymbol{\omega}_{gb} \quad \boldsymbol{\omega}_{ab}]$

### 《组合导航算法讲义》勘误：

#### 1. 四元数旋转变换部分笔误（惯导姿态算法，P18）

公式 (99) 第三行，左边数第三项少了符号“k”。

$$\begin{aligned}
L_q(\mathbf{a}) &= (q_s + \mathbf{q}_v) \circ \mathbf{a} \circ (q_s - \mathbf{q}_v) \\
&= (q_s + \mathbf{q}_v) \circ k\mathbf{q}_v \circ (q_s - \mathbf{q}_v) \\
&= (kq_s\mathbf{q}_v - k\mathbf{q}_v^T\mathbf{q}_v + \mathbf{q}_v \times \mathbf{q}_v) \circ (q_s - \mathbf{q}_v) \\
&= (kq_s\mathbf{q}_v - k\mathbf{q}_v^T\mathbf{q}_v) \circ (q_s - \mathbf{q}_v) \\
&= kq_s^2\mathbf{q}_v - kq_s\mathbf{q}_v^T\mathbf{q}_v - kq_s\mathbf{q}_v \circ \mathbf{q}_v + (k\mathbf{q}_v^T\mathbf{q}_v)\mathbf{q}_v \\
&= kq_s^2\mathbf{q}_v - kq_s\mathbf{q}_v^T\mathbf{q}_v - kq_s(-\mathbf{q}_v^T\mathbf{q}_v + \mathbf{q}_v \times \mathbf{q}_v) + (k\mathbf{q}_v^T\mathbf{q}_v)\mathbf{q}_v \\
&= kq_s^2\mathbf{q}_v + k(\mathbf{q}_v^T\mathbf{q}_v)\mathbf{q}_v \\
&= k(q_s^2 + \mathbf{q}_v^T\mathbf{q}_v)\mathbf{q}_v \\
&= k\mathbf{q}_v \\
&= \mathbf{a}
\end{aligned} \tag{99}$$

图片红色框的部分应为： $k\mathbf{q}_v \times \mathbf{q}_v$