



武汉大学



GNSS/INS组合导航开源代码

KF-GINS分享交流



i2Nav微信公众号

王立强、陈起金
武汉大学多源智能导航实验室 (i2Nav)
www.i2nav.com
<https://github.com/i2Nav-WHU>

2023年5月14日



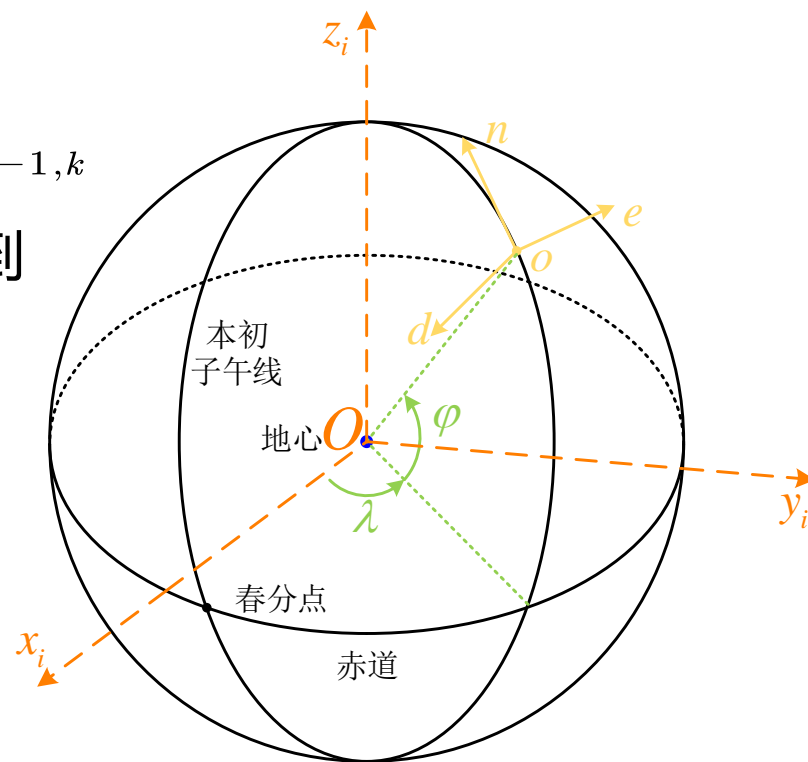
i2Nav开源算法交流QQ群

□ IMU测量值

- 速率型数据：加速度和角速度： f_{ib}^b 、 ω_{ib}^b
- **增量型数据**：采样间隔内速度增量和角度增量： $\Delta v_{k-1,k}^b$ 、 $\Delta \theta_{k-1,k}^b$
- 高精度惯导输出增量数据；通过模拟积分，或高频采样积分得到

□ 坐标系

- 载体坐标系(b系)，IMU几何中心为原点，前向、右向和垂向（下）
- 导航坐标系(n系)，导航结果描述对象为原点，北向、东向和地向下
- 大地坐标系，纬度、经度、高程表示绝对位置
- (地心)惯性坐标系(i系)，惯性传感器测量值的参考坐标系



常用参考坐标系

新手入门系列4——MEMS IMU**原始数据采集和时间同步**的那些坑

http://i2nav.com/index/newListDetail_zw.do?newskind_id=13a8654e060c40c69e5f3d4c13069078&newsinfo_id=2e05f5cdac6b4725b8bfe54a689c7add



捷联惯性导航算法



简化IMU
微分方程

$$\dot{\mathbf{p}}_{wb}^w = \mathbf{v}_{wb}^w$$

$$\dot{\mathbf{v}}_{wb}^w = \mathbf{C}_b^w \mathbf{f}_{ib}^b + \mathbf{g}_l^w$$

$$\dot{\mathbf{C}}_b^w = \mathbf{C}_b^w (\boldsymbol{\omega}_{wb}^b \times)$$

$$\dot{\varphi} = \frac{v_N}{R_M + h}, \dot{\lambda} = \frac{v_E}{(R_N + h) \cos \varphi}, \dot{h} = -v_D$$

$$\dot{\mathbf{v}}_{eb}^n = \mathbf{C}_b^n \mathbf{f}_{ib}^b + \mathbf{g}_l^n - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \mathbf{v}_{eb}^n$$

$$\dot{\mathbf{C}}_b^n = \mathbf{C}_b^n [\boldsymbol{\omega}_{nb}^b]_{\times} = \mathbf{C}_b^n [\boldsymbol{\omega}_{ib}^b]_{\times} - [\boldsymbol{\omega}_{in}^n]_{\times} \mathbf{C}_b^n$$

捷联惯性导航
微分方程

简化
中值
积分

$$\mathbf{v}_{wb,k}^w = \mathbf{v}_{wb,k-1}^w + \mathbf{C}_{b,k-1}^w \Delta \mathbf{v}_k + \mathbf{g}_l^w \Delta t_k$$

$$\mathbf{p}_{wb,k}^w = \mathbf{p}_{wb,k-1}^w + \frac{1}{2} (\mathbf{v}_{wb,k-1}^w + \mathbf{v}_{wb,k}^w) \Delta t_k$$

$$\mathbf{q}_{b_k}^w = \mathbf{q}_{b_{k-1}}^w \otimes \mathbf{q}_{b_k}^{b_{k-1}} = \mathbf{q}_{b_{k-1}}^w \otimes \begin{bmatrix} 1 \\ \Delta \boldsymbol{\theta}_k / 2 \end{bmatrix}$$

$$\mathbf{v}_{eb,k}^n = \mathbf{v}_{eb,k-1}^n + \underbrace{\left[\mathbf{I} - \frac{1}{2} (\boldsymbol{\zeta}_{n(k-1)n(k)} \times) \right] \mathbf{C}_{b(k-1)}^{n(k-1)} \Delta \mathbf{v}_{f,k}^{b(k-1)}}_{\text{比力积分项}}$$

$$+ \underbrace{\mathbf{g}_l^n \Delta t_k - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \mathbf{v}_{eb}^n \big|_{t=t_{k-1/2}} \Delta t_k}_{\text{重力/哥氏积分项}}$$

捷联惯导
速度更新

$$\boldsymbol{\zeta}_{n(k-1)n(k)} \approx \boldsymbol{\omega}_{in}^n \big|_{t=t_{k-1/2}} \Delta t_k$$

$$\Delta \mathbf{v}_{f,k}^{b(k-1)} = \Delta \mathbf{v}_k + \frac{1}{2} \Delta \boldsymbol{\theta}_k \times \Delta \mathbf{v}_k + \frac{1}{12} (\Delta \boldsymbol{\theta}_{k-1} \times \Delta \mathbf{v}_k + \Delta \mathbf{v}_{k-1} \times \Delta \boldsymbol{\theta}_k)$$

哥氏积分项
划桨、旋转效应

捷联
惯导
位置
更新

$$h_k = h_{k-1} - \frac{1}{2} (v_{D,k-1} + v_{D,k}) \Delta t_k$$

$$\varphi_k = \varphi_{k-1} + \frac{v_{N,k-1} + v_{N,k}}{2(R_{M,k-\frac{1}{2}} + h_{k-\frac{1}{2}})} \Delta t_k$$

$$\lambda_k = \lambda_{k-1} + \frac{v_{E,k-1} + v_{E,k}}{2(R_{N,k-\frac{1}{2}} + h_{k-\frac{1}{2}}) \cos \varphi_{k-\frac{1}{2}}} \Delta t_k$$

等效旋转矢量更新
圆锥效应

$$\mathbf{q}_{b_k}^{n_k} = \mathbf{q}_{n_{k-1}}^{n_k} \mathbf{q}_{b_{k-1}}^{n_{k-1}} \mathbf{q}_{b_k}^{b_{k-1}}$$

$$\mathbf{q}_{n_{k-1}}^{n_k} \leftarrow \boldsymbol{\zeta}_k \approx \boldsymbol{\omega}_{in}^n \big|_{t=t_{k-1/2}} \Delta t_k$$

$$\mathbf{q}_{b_k}^{b_{k-1}} \leftarrow \boldsymbol{\phi}_k = \Delta \boldsymbol{\theta}_k + \frac{1}{12} \Delta \boldsymbol{\theta}_{k-1} \times \Delta \boldsymbol{\theta}_k$$

捷联惯导
姿态更新



代码实现-惯导速度更新



$$\mathbf{v}_{eb,k}^n = \mathbf{v}_{eb,k-1}^n + \underbrace{\left[\mathbf{I} - \frac{1}{2}(\boldsymbol{\zeta}_{n(k-1)n(k)} \times) \right] \mathbf{C}_{b(k-1)}^{n(k-1)} \Delta \mathbf{v}_{f,k}^{b(k-1)}}_{n \text{系比力积分项}}$$

捷联惯导
速度更新

$$+ \underbrace{\mathbf{g}_l^n \Delta t_k - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \mathbf{v}_{eb}^n|_{t=t_{k-1/2}} \Delta t_k}_{\text{重力/哥氏积分项}}$$

哥氏积分项
划桨、旋转效应

$$\boldsymbol{\zeta}_{n(k-1)n(k)} \approx \boldsymbol{\omega}_{in}^n|_{t=t_{k-1/2}} \Delta t_k$$

$$\underbrace{\Delta \mathbf{v}_{f,k}^{b(k-1)}}_{b \text{系比力积分项}} = \Delta \mathbf{v}_k + \frac{1}{2} \Delta \boldsymbol{\theta}_k \times \Delta \mathbf{v}_k + \frac{1}{12} (\Delta \boldsymbol{\theta}_{k-1} \times \Delta \mathbf{v}_k + \Delta \mathbf{v}_{k-1} \times \Delta \boldsymbol{\theta}_k)$$

速度更新

计算b系比力积分项和n系下的比力积分项

k-1时刻状态计算重力/哥氏积分项

更新得到k时刻速度，位置

计算k-1/2时刻的位置

k-1/2时刻状态重新计算n系比力积分项

重新计算重力/哥氏积分项

更新得到k时刻的速度

kf-gins/insmech.cpp/INSMech::velUpdate

// b系比力积分项, 旋转效应和划桨效应

temp1 = imucur.dtheta.cross(imucur.dvel) / 2;

temp2 = imupre.dtheta.cross(imucur.dvel) / 12;

temp3 = imupre.dvel.cross(imucur.dtheta) / 12;

d_vfb = imucur.dvel + temp1 + temp2 + temp3;

// 比力积分项投影到n系

temp3 = (wie_n + wen_n) * imucur.dt / 2;

cnn = I33 - Rotation::skewSymmetric(temp3);

d_vfn = cnn * pvapre.att.cbn * d_vfb;

// 计算重力/哥氏积分项

d_vgn = (gl - (2 * wie_n +
wen_n).cross(pvapre.vel)) * imucur.dt;

// 外推得到中间时刻位置

midpos = Earth::blh(qne, midpos[2]);

// 重新计算n系下平均比力积分项

d_vfn = cnn * pvapre.att.cbn * d_vfb;

// 重新计算重力、哥氏积分项

d_vgn = (gl - (2 * wie_n + wen_n).cross(midvel))
* imucur.dt;

// 速度更新完成

pvacur.vel = pvapre.vel + d_vfn + d_vgn;



代码实现-惯导位置/姿态更新



捷联惯导位置更新

$$h_k = h_{k-1} - \frac{1}{2}(v_{D,k-1} + v_{D,k})\Delta t_k$$

$$\varphi_k = \varphi_{k-1} + \frac{v_{N,k-1} + v_{N,k}}{2(R_{M,k-\frac{1}{2}} + h_{k-\frac{1}{2}})}\Delta t_k$$

$$\lambda_k = \lambda_{k-1} + \frac{v_{E,k-1} + v_{E,k}}{2(R_{N,k-\frac{1}{2}} + h_{k-\frac{1}{2}})\cos\varphi_{k-\frac{1}{2}}}\Delta t_k$$

位置更新

更新高程，计算平均高程

更新纬度，计算平均纬度

更新经度

得到k时刻的位置

姿态更新

计算k-1/2时刻的位置和速度

计算n系旋转四元数

双子样假设计算b系旋转四元数

更新k时刻姿态

$$\mathbf{q}_{b_k}^{n_k} = \mathbf{q}_{n_{k-1}}^{n_k} \mathbf{q}_{b_{k-1}}^{n_{k-1}} \mathbf{q}_{b_k}^{b_{k-1}}$$

$$\mathbf{q}_{n_{k-1}}^{n_k} \leftarrow \zeta_k \approx \omega_{in}^n \big|_{t=t_{k-1/2}} \Delta t_k$$

$$\mathbf{q}_{b_k}^{b_{k-1}} \leftarrow \phi_k = \Delta \theta_k + \frac{1}{12} \Delta \theta_{k-1} \times \Delta \theta_k$$

等效旋转矢量更新
圆锥效应

捷联惯导
姿态更新

`kf-gins/insmech.cpp/INSMech::attUpdate`

// 重新计算中间时刻的速度和位置，地理参数

```
midpos = Earth::blh(qne_mid, midpos[2]);
```

// 计算n系的旋转四元数 k-1时刻到k时刻变换

```
temp1 = -(wie_n + wen_n) * imucur.dt;
```

```
qnn = Rotation::rotvec2quaternion(temp1);
```

// 计算b系旋转四元数 补偿二阶圆锥误差

```
temp1 = imucur.dtheta + imupre.dtheta.cross(imucur.dtheta) / 12;
```

```
qbb = Rotation::rotvec2quaternion(temp1);
```

// 姿态更新完成 `pvacur.att.qbn = qnn * pvapre.att.qbn * qbb;`

```
pvacur.att.cbn = Rotation::quaternion2matrix(pvacur.att.qbn);
```

```
pvacur.att.euler = Rotation::matrix2euler(pvacur.att.cbn);
```

E.-H. Shin, "Estimation techniques for low-cost inertial navigation," Cal, 2005. (位置更新方程参考)