



GNSS/INS组合导航开源代码

KF-GINS分享交流



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





捷联惯性导航算法



□ IMU测量值

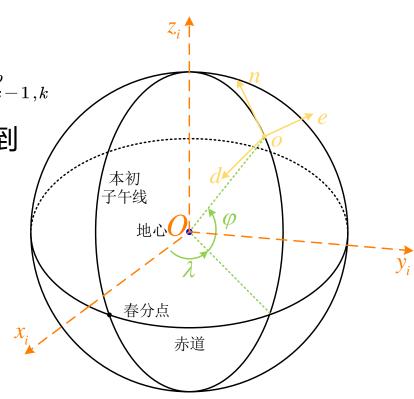
 $lacksymbol{\blacksquare}$ 速率型数据:加速度和角速度: $oldsymbol{f}_{ib}^b$ 、 $oldsymbol{\omega}_{ib}^b$

 $lacksymbol{\blacksquare}$ 增量型数据:采样间隔内速度增量和角度增量: $\Delta oldsymbol{v}_{k-1,k}^b$ 、 $\Delta oldsymbol{ heta}_{k-1,k}^b$

■ 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

口 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
- 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=13a8654e060c40c69e 5f3d4c13069078&newsinfo id=2e05f5cdac6b4725b8bfe54a689c7add



违联惯性导航算法



简化IMU 微分方程

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

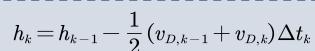
$$\dot{oldsymbol{v}}_{wb}^{w}\!=\!oldsymbol{C}_{b}^{\,w}oldsymbol{f}_{ib}^{\,b}+oldsymbol{g}_{l}^{\,w}$$

$$\dot{oldsymbol{C}}_{b}^{w} = oldsymbol{C}_{b}^{w}(oldsymbol{\omega}_{wb}^{b} imes)$$

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

简化
中值
$$oldsymbol{p}_{wb,k}^w = oldsymbol{p}_{wb,k-1}^w + rac{1}{2} \left(oldsymbol{v}_{wb,k-1}^w + oldsymbol{v}_{wb,k}^w
ight) \Delta t_k$$

$$oldsymbol{q}_{b_k}^{w} \!=\! oldsymbol{q}_{b_{k-1}}^{w} \!\otimes oldsymbol{q}_{b_k}^{b_{k-1}} \!=\! oldsymbol{q}_{b_{k-1}}^{w} \!\otimes \! egin{bmatrix} 1 \ \Delta heta_k/2 \end{bmatrix}$$



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

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

$$\dot{arphi} = rac{v_N}{R_M + h}, \; \dot{\lambda} = rac{v_E}{(R_N + h) {\cos arphi}}, \; \dot{h} = - \, v_D$$

$$\dot{oldsymbol{v}}_{eb}^n = oldsymbol{C}_b^n oldsymbol{f}_{ib}^b + oldsymbol{g}_l^n - ig(2oldsymbol{\omega}_{ie}^n + oldsymbol{\omega}_{en}^nig) imes oldsymbol{v}_{eb}^n$$

$$\dot{m{C}}_b^{\,n}\!=\!m{C}_b^{\,n}[m{\omega}_{nb}^{\,b}]_{\! imes}\!=\!m{C}_b^{\,n}[m{\omega}_{ib}^{\,b}]_{\! imes}\!-\![m{\omega}_{in}^{\,n}]_{\! imes}\!m{C}_b^{\,n}$$

$$m{v}_{eb,k}^n \! = \! m{v}_{eb,k-1}^n \! + \! \left[m{I} \! - \! rac{1}{2} (m{\zeta}_{n(k-1)n(k)} \! imes)
ight] \! m{C}_{b(k-1)}^{n(k-1)} \Delta m{v}_{f,k}^{b(k-1)} \; .$$

 $+\left.oldsymbol{g}_{l}^{n}\Delta t_{k}\!-\!\left(2oldsymbol{\omega}_{ie}^{n}\!+\!oldsymbol{\omega}_{en}^{n}
ight)\! imes\!oldsymbol{v}_{eb}^{n}\!\mid_{t=t_{k-1/2}}\!\!\Delta t_{k}$

 $oldsymbol{\zeta}_{n\,(k-1)\,n\,(k)}\!pproxoldsymbol{\omega}_{in}^{\,n}ig|_{t=t_{k-1/2}}\!\Delta t_k$

哥氏积分项 划桨、旋转效应

捷联惯导

速度更新

捷联惯性导航

微分方程

$$\Delta oldsymbol{v}_{f,k}^{b\,(k-1)} = \Delta oldsymbol{v}_k + rac{1}{2}\Delta oldsymbol{ heta}_k imes \Delta oldsymbol{v}_k + rac{1}{12}\left(\Delta oldsymbol{ heta}_{k-1} imes \Delta oldsymbol{v}_k + \Delta oldsymbol{v}_{k-1} imes \Delta oldsymbol{ heta}_k
ight)$$

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

 $oldsymbol{q}_{n_{k-1}}^{n_k} \leftarrow oldsymbol{\zeta}_k \!pprox oldsymbol{\omega}_{in}^{n}ig|_{t=t_{k-1/2}} \!\Delta t_k$

 $m{q}_{b_k}^{b_{k-1}} \leftarrow m{\phi}_k = \Delta m{ heta}_k + rac{1}{12} \Delta m{ heta}_{k-1} imes \Delta m{ heta}_k$ 姿态更新

圆锥效应



代码实现-惯导速度更新



$$oldsymbol{v}_{eb,k}^n = oldsymbol{v}_{eb,k-1}^n + igg[oldsymbol{I} - rac{1}{2} (oldsymbol{\zeta}_{n(k-1)n(k)} imes) igg] oldsymbol{C}_{b(k-1)}^{n(k-1)} \Delta oldsymbol{v}_{f,k}^{b(k-1)}$$

捷联惯导 速度更新

n系比力积分项

$$+\left.oldsymbol{g}_{l}^{n}\Delta t_{k}\!-\!\left(2oldsymbol{\omega}_{ie}^{n}\!+\!oldsymbol{\omega}_{en}^{n}
ight)\!\! imes\!oldsymbol{v}_{eb}^{n}ig|_{t=t_{k-1/2}}\!\Delta t_{k}$$

重力/哥氏积分项

$$oldsymbol{\zeta}_{n\,(k-1)\,n\,(k)}\!pprox\!oldsymbol{\omega}_{in}^{n}ig|_{t=t_{k-1/2}}\!\Delta t_{k}$$

哥氏积分项 划桨、旋转效应

$$\Delta oldsymbol{v}_{f,k}^{b(k-1)} = \Delta oldsymbol{v}_k + rac{1}{2}\Delta oldsymbol{ heta}_k imes \Delta oldsymbol{v}_k + rac{1}{12}\left(\Delta oldsymbol{ heta}_{k-1} imes \Delta oldsymbol{v}_k + \Delta oldsymbol{v}_{k-1} imes \Delta oldsymbol{ heta}_k
ight)$$

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

速度更新

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; = I33 - Rotation::skewSymmetric(temp3); d vfn = cnn * pvapre.att.cbn * d vfb; // 计算重力/哥式积分项 $d \, vgn = (gl - (2 * wie n +$ 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、哥式积分项 $d_vgn = (gl - (2 * wie_n + wen_n).cross(midvel))$ * 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

$$h_k \! = \! h_{k-1} \! - \! rac{1}{2} \left(v_{D,k-1} \! + \! v_{D,k} \!
ight) \! \Delta t_k$$

操験
$$arphi_k = arphi_{k-1} + rac{v_{N,k-1} + v_{N,k}}{2(R_{M,k-rac{1}{2}} + h_{k-rac{1}{2}})} \Delta t_k$$

更新
$$\lambda_k = \lambda_{k-1} + rac{v_{E,k-1} + v_{E,k}}{2(R_{N,k-rac{1}{2}} + h_{k-rac{1}{2}}){\cos{arphi}_{k-rac{1}{2}}}} \Delta t_k$$

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

计算k-1/2时刻的位置和速度

计算n系旋转四元数 姿态更新

双子样假设计算b系旋转四元数

更新k时刻姿态

 $\mathbf{q}_{b_k}^{n_k} = \mathbf{q}_{n_{k-1}}^{n_k} \mathbf{q}_{b_{k-1}}^{n_{k-1}} \mathbf{q}_{b_k}^{b_{k-1}}$ 等效旋转矢量更新 圆锥效应 $\|oldsymbol{q}_{n_{k-1}}^{n_k} \leftarrow oldsymbol{\zeta}_k pprox oldsymbol{\omega}_{in}^n\|_{t=t_{k-1/2}} \Delta t_k$

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

```
// 重新计算中间时刻的速度和位置, 地理参数
        = Earth::blh(qne mid, midpos[2]);
midpos
// 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;
     = 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;
     = 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)