

雾膜软件

WMS507 代码包

惯性卫星紧组合导航仿真 Mat lab

版本 20240824

1. 内容介绍

本代码包的功能主要有：生成一组惯性卫星紧组合导航的仿真数据；根据陀螺仪和加速度计信息计算惯性导航；根据卫星伪距、伪距率计算组合导航。

惯性导航在以地理系作为导航系解算，输出纬经高、东北天速度。卫星导航和组合导航在地心系 ECEF 解算，即 XYZ 位置。

采用扩展卡尔曼滤波方法，17 维状态量。观测量与可见卫星数有关。当卫星数为 7 时，观测量为 14 维度。

代码包适用于：组合导航的学习、研究、教学、科研。

本代码为了便于学习理解，力求简洁清楚。只保留了关键的、普适的导航计算原理。一些差异化功能没有包含在本代码中，例如：通信协议、初始对准、传感器标定、生成仿真路线等等。需要其余功能的用户可以联系本店定制开发。

2. 应用说明

2.1. 概念定义

惯性测量单元为 3 轴陀螺仪和 3 轴加速度计。定义 x 向东、y 向北、z 向天为姿态 0 位置。旋转方向和角速度方向满足右手法则，即右手握住坐标轴，大拇指位于坐标轴正向，则其余四个手指指向旋转正向。姿态的欧拉角旋转顺序定义为依次绕 z、x、y 旋转。

组合导航中，扩展卡尔曼滤波的状态变量定义为 15 维度的误差量：纬经高，东北天速度，东北天姿态，三轴陀螺仪零偏，三轴加速度计零偏。

如果无特殊说明，一般采用国际单位制。角度单位为 rad，角速度单位为 rad/s，速度单位 m/s，加速度单位 m/s/s。

2.2. 生成仿真数据

直接运行 `creatdata.m`，即生成一组仿真数据，存储于 `data.mat` 文件。

常见的生成仿真数据的方式有 2 种：一种是给定路线、计算得到陀螺仪、加速度计数值；另一种是给定陀螺仪、加速度计数值，计算得到路线。本代码包采用的是后一种方式，因为这样更简单。

生成仿真数据的流程为：

| |
|-------------------|
| 给定陀螺仪、加速度计数值 |
| 给定卫星位置、速度、卫星修正量数值 |
| 惯性导航计算，得到准确路线 |
| 计算伪距、伪距率 |
| 给数据添加额外误差 |
| 让卫星导航数据间隔变大 |
| 保存 |

卫星的修正量包括钟差、电离层、对流层。卫星数据的内容基本对应于 OEM7 系列卫星接收机的 SATXYZ2 和 RANGE 数据。如果需要实物实验，可以采用类似的硬件配置。

The OEM7 family defines ionospheric and tropospheric corrections positively which means that ionospheric and tropospheric corrections are added to the geometric ranges or subtracted from the measured pseudoranges. A positive clock offset indicates the clock is running ahead of the reference time. For example:

$$P = p + pd + c(dT - dt) + d(\text{ion}) + d(\text{trop}) + E_p$$

is equivalent to

$$P - c(dT - dt) - d(\text{ion}) - d(\text{trop}) = p + pd + E_p$$

where

P = measured pseudorange

p = geometric range

pd = orbit error

dt = satellite clock offset

dT = receiver clock offset

d (ion) = ionospheric delay

d (trop) = tropospheric delay

c = speed of light

E_p = noise and multipath

代码包生成数据的卫星位置、速度、修正量数值等与实际情况无关，只是随便给定一个数值以演示原理。代码包没有根据卫星椭圆轨道参数计算卫星位置速度的功能。

2.3. 主程序

主程序为 `instance1.m`。直接运行即可。

程序主要工作流程为：

```

设定初值
while(1)
{
    惯性导航计算
    更新状态方程
    if(间隔一段时间收到卫星数据)
    {
        计算伪距、伪距率误差，即 Z
        计算卡尔曼滤波
        根据卡尔曼滤波的计算结果补偿惯性导航的误差
    }
    保存数据
}
绘图

```

2.4. 设定初值

设定传感器的采样间隔时间 `dTins`，单位为 s。

初始姿态存储于四元数 `atti1` 中。可以直接修改 `atti1` 以改变初始姿态。或者可以通过 `setoula` 函数，用欧拉角设置姿态四元数。`setoula` 函数输入单位为度。

初始速度存储于 `speed1` 中，可以直接修改。

初始位置存储于 `pos1` 中，可以直接修改。顺序为纬度、经度、高度。注意纬度和经度的单位为 rad。

卡尔曼滤波中，初始状态的方差矩阵为 `Pk1`。测量的方差矩阵为 `R`。系统噪声的方差矩阵为 `Q1`，在本代码包中，`Q1` 与时间和 `Q0` 有关。可以根据需要设置上述参数。

2.5. 导航计算

惯性导航计算为 `insgyroacc(gyro,acc,att1,speed1,pos1,dTins)`。把 `gyro` 和 `acc` 换为实际的陀螺仪和加速度计数据即可，单位为 `rad/s` 和 `m/s/s`。

卫星导航的位置和速度分别为 `gnsspos` 和 `gnssspeed`，换为真实数据即可。卡尔曼滤波由 `kal` 函数计算。

现有代码按照固定间隔计算卫星数据；但是在采用真实数据时，应该按照实际的数据间隔计算。

2.6. 数据格式

程序默认状态仿真了 7 个可见卫星的情况。

输入数据包括：3 个角速度 (`rad/s`)，3 个加速度 (`m/s/s`)，7 个伪距，7 个伪距率，7 个*3 种的卫星修正量，1 列卫星数据更新标志。为了对照，仿真数据包含姿态参考、速度参考、位置参考各 3 列。

输出数据前 9 列为导航结果，存储于 `dataA` 矩阵，用于画图或保存。前 9 列为 3 个姿态角 (`deg`)、3 个速度 (`m/s`)、3 个位置 (`rad, rad, m`)。注意此处姿态欧拉角单位为度。

3. 计算原理

3.1. 紧组合算法概述

紧组合与松组合算法的绝大部分是一致的，比如惯性导航部分、状态方程的惯性导航部分。主要区别有：1.紧组合卡尔曼滤波的状态量增加了反映接收机时钟误差的部分。如果有些接收机内部补偿了时差，则可以沿用松组合的状态方程。2.紧组合的观测方程与松组合不同。松组合的观测量是位置、速度，紧组合的观测量是伪距、伪距率。观测方程的矩阵相应修改。

3.2. 坐标系

载体系 `b` 定义为与载体固定连接的坐标系，不妨取 `xyz` 轴为右前上。

地理系 `t` 定义为与载体处地面重合的坐标系，不妨取 `xyz` 轴为东北天。

导航坐标系 `n` 是表示导航结果的坐标系。在航海、航空领域中，为了避免船只、飞机通过南北极附近时 `n` 系快速旋转导致导航结果异常，`n` 系会与 `t` 系有一定的夹角。在普通导航系统中，可以不虑载体通过南北极的情况，因此选取 `n` 系与 `t` 系重合以使导航算法简化。

平台坐标系 `p`，是平台式导航系统中传感器的指向，或者是捷联式导航系统中数学换算后的传感器的指向。理想情况下 `p` 系与 `n` 系重合；但是由于陀螺仪误差等因素，真实的 `p` 系与 `n` 系有误差角。捷联式导航系统希望把加速度换算到 `n` 系中，但是实际上是换算到了 `p` 系中。在一般的导航计算中，不必刻意区分 `p` 系和 `n` 系，但是在分析误差时需要引入 `p` 系。

地球坐标系 `e`（地心坐标系），是和地球固连的坐标系，不妨规定 `z` 轴沿着南北极方向指向北，`x` 轴指向 0 经度方向。

惯性参考系 `i`。惯性参考系主要用于描述概念。惯性导航中一般不需要真正地在惯性参考系中投影，所以不必在惯性参考系中规定坐标系。

完整地描述角速率、姿态、加速度、速度、位移等需要 3 个坐标系。坐标系 `β` 相对于坐标系 `α` 的变化量 `x` 在坐标系 `γ` 的投影表示为 $x_{αβ}^γ$ 。例如，地球自转在地理系的坐标为

$$\omega_{ie}^t = \begin{bmatrix} 0 \\ \omega_e \cos L \\ \omega_e \sin L \end{bmatrix} \quad (3-1)$$

其中 ω_e 是地球自转角速率， L 是纬度。这是地球系 `e` 相对于惯性系 `i` 的转动在地理系 `t` 的投影。在这种表示方法下，一些简单的计算规则如下：

同一个坐标系内表示的变量符合向量加法规则，即

$$x_{AB}^γ + x_{BC}^γ = x_{AC}^γ \quad (3-2)$$

同一个变量在不同坐标系的换算可以用矩阵表示。

$$\mathbf{x}_{\alpha\beta}^{\mu} = \mathbf{C}_{\gamma}^{\mu} \mathbf{x}_{\alpha\beta}^{\gamma} \quad (3-3)$$

坐标变换矩阵表示旋转关系。例如二维的坐标变换矩阵为

$$\mathbf{C}_n^b = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (3-4)$$

三维的坐标旋转有 3 个自由度，可以看作是类似形式矩阵相乘。

坐标变换矩阵是正交矩阵，逆矩阵是原矩阵的转置

$$\mathbf{C}_{\mu}^{\gamma} = (\mathbf{C}_{\gamma}^{\mu})^{-1} = (\mathbf{C}_{\gamma}^{\mu})^T \quad (3-5)$$

3.3. 惯性导航

3.3.1. 基本原理

惯性导航的基本原理是：陀螺仪测量角速度，角速度积分得到姿态。加速度计测量加速度，加速度积分得到速度，速度积分得到位置。

实际情况中有一些因素导致上述计算变得复杂。1.需要进行一些坐标系变换。2.需要考虑地球的自转、重力、以及球形形状。

3.3.2. 姿态更新

三维空间有 3 个旋转自由度。类似式(3-4)，依次绕三个坐标轴旋转，则坐标变换矩阵为

$$\mathbf{C}_n^b = \begin{bmatrix} \cos \theta_y & 0 & -\sin \theta_y \\ 0 & 1 & 0 \\ \sin \theta_y & 0 & \cos \theta_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & \sin \theta_x \\ 0 & -\sin \theta_x & \cos \theta_x \end{bmatrix} \begin{bmatrix} \cos \theta_z & \sin \theta_z & 0 \\ -\sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3-6)$$

把坐标变换矩阵表示为绕坐标轴分别旋转三次，三次旋转的角度即为欧拉角。旋转的顺序并不是唯一的，也可以定义旋转顺序不同的欧拉角。同一个坐标变换矩阵，在不同的旋转顺序定义下，有不同的欧拉角角度；同样的旋转角度，按照不同的坐标轴顺序旋转，会得到不同的坐标变换矩阵；这个性质称为姿态角的不可交换性。所以使用欧拉角描述姿态时必须规定清楚旋转顺序。本书中欧拉角定义为：初始状态右前上（xyz）三轴位于东北天方向，依次绕上轴旋转偏航角，绕右轴旋转俯仰角，绕前轴旋转横滚角。

如果每次旋转的角度很小，则坐标变换矩阵近似为

$$d\mathbf{C} = \begin{bmatrix} 1 & 0 & -d\theta_y \\ 0 & 1 & 0 \\ d\theta_y & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & d\theta_x \\ 0 & -d\theta_x & 1 \end{bmatrix} \begin{bmatrix} 1 & d\theta_z & 0 \\ -d\theta_z & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3-7)$$

略去二阶小量，则有

$$d\mathbf{C} = \begin{bmatrix} 1 & d\theta_z & -d\theta_y \\ -d\theta_z & 1 & d\theta_x \\ d\theta_y & -d\theta_x & 1 \end{bmatrix} \quad (3-8)$$

上式表示了坐标旋力矩阵与旋转角度的关系。如果旋转角度很小，则不必考虑旋转顺序。为了表示的方便，引入角增量反对称矩阵

$$[\boldsymbol{\theta}] = \begin{bmatrix} 0 & -\theta_z & \theta_y \\ \theta_z & 0 & -\theta_x \\ -\theta_y & \theta_x & 0 \end{bmatrix} \quad (3-9)$$

那么姿态矩阵更新公式为

$$\mathbf{C}_b^i(t+T) = \mathbf{C}_b^i(t) \lim_{k \rightarrow +\infty} \left(\mathbf{I} + \frac{[\boldsymbol{\theta}_{ib}^b]}{k} \right)^k = \mathbf{C}_b^i(t) \exp([\boldsymbol{\theta}_{ib}^b]) \quad (3-10)$$

其中 \exp 表示自然常数 e 为底数的指数函数。 $\mathbf{C}_b^i(t)$ 是上一时刻的姿态矩阵， $\mathbf{C}_b^i(t+T)$ 是下一时刻的姿态矩阵。上式即姿态更新公式。

利用麦克劳林公式，能得到更便于计算的如下公式

$$\exp([\boldsymbol{\theta}]) = \mathbf{I} + \frac{\sin|\boldsymbol{\theta}|}{|\boldsymbol{\theta}|} [\boldsymbol{\theta}] + \frac{1 - \cos|\boldsymbol{\theta}|}{|\boldsymbol{\theta}|^2} [\boldsymbol{\theta}]^2 \quad (3-11)$$

如果旋转角度较小，同时为了避免分母为 0，可以采用如下近似公式

$$\exp([\boldsymbol{\theta}]) \approx \mathbf{I} + [\boldsymbol{\theta}] \quad (3-12)$$

根据上述若干公式，使用陀螺仪数据计算得到姿态。

实际导航系统中，为了防止计算误差导致姿态矩阵失去正交性，也为了减少计算量，往往采用四元数代替姿态矩阵进行姿态更新。四元数定义为

$$\mathbf{q} = \left[\cos \frac{\theta}{2} \quad u_x \sin \frac{\theta}{2} \quad u_y \sin \frac{\theta}{2} \quad u_z \sin \frac{\theta}{2} \right]^T \quad (3-13)$$

其中 θ 是旋转的角度， $[u_x \quad u_y \quad u_z]^T$ 是旋转轴的单位向量。

四元数也可以表示为

$$\mathbf{q} = \cos \frac{\theta}{2} + \mathbf{A} \sin \frac{\theta}{2} \quad (3-14)$$

其中 \mathbf{A} 是旋转轴的单位向量。

四元数姿态微分方程为

$$\dot{\mathbf{q}} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \mathbf{q} \quad (3-15)$$

引入 4 维的角增量矩阵

$$[\boldsymbol{\theta}] = \begin{bmatrix} 0 & -\theta_x & -\theta_y & -\theta_z \\ \theta_x & 0 & \theta_z & -\theta_y \\ \theta_y & -\theta_z & 0 & \theta_x \\ \theta_z & \theta_y & -\theta_x & 0 \end{bmatrix} \quad (3-16)$$

四元数更新姿态的公式为

$$\mathbf{q}(t+T) = \left(\cos \frac{|\boldsymbol{\theta}|}{2} \mathbf{I} + \frac{\sin \frac{|\boldsymbol{\theta}|}{2}}{|\boldsymbol{\theta}|} [\boldsymbol{\theta}] \right) \mathbf{q}(t) \quad (3-17)$$

姿态可以用 3*3 矩阵或者 4*1 的四元数表示。本代码包采用四元数计算姿态，这是主流方法。但是矩阵对于坐标系变换的计算比较方便，所以坐标变换的地方使用了矩阵表示姿态。根据角度增量更新姿态四元数，为 `qupdate` 函数。

实际导航中计算姿态时，要扣除地球自转的影响。此外，由于地球是球形的，位置变化时地面会向下弯曲，相对于地面的姿态就变了，所以速度会导致一个额外的角速度。

惯性导航需要依次计算姿态、速度、位置。陀螺仪直接测到 **b** 系相对于 **i** 系的转动，而导航解算中需要计算 **b** 系相对于 **n** 系的姿态。导航系 **n** 系相对于惯性系 **i** 系的相对转动包括两个部分：一是地球自转角速率；二是因为地球表面是曲面，载体位置变化会导致 **n** 系相对 **e** 系的姿态发生变化，角速率 $\boldsymbol{\omega}_{en}^n$ 与速度有关

$$\boldsymbol{\omega}_{en}^n = \begin{bmatrix} -\frac{v_N}{R_m + H} \\ \frac{v_E}{R_p + H} \\ \frac{v_E \tan L}{R_p + H} \end{bmatrix} \quad (3-18)$$

其中 R_m 和 R_p 分别是子午圈和卯酉圈的半径， L 是纬度。

因为

$$\boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib}^b - \boldsymbol{\omega}_{ie}^b - \boldsymbol{\omega}_{en}^b \quad (3-19)$$

所以

$$\boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib}^b - \mathbf{C}_n^b(\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \quad (3-20)$$

根据上述角速率计算 \mathbf{C}_n^b ，即姿态。

3.3.3. 速度更新和位置更新

加速度要从传感器的坐标系换算到东北天。本来速度是加速度积分。但是由于地球自转，需要扣除离心力和科里奥利力。此外，加速度计不能区分重力和一般的加速度，所以还需要扣除重力。

加速度为

$$\dot{\mathbf{V}}_{en}^n = \mathbf{C}_b^n \mathbf{f}_b - 2\boldsymbol{\omega}_{ie}^n \times \mathbf{V}_{en}^n - \boldsymbol{\omega}_{en}^n \times \mathbf{V}_{en}^n + \mathbf{g} \quad (3-21)$$

其中 \mathbf{f}_b 是加速度计的数值。根据等效原理，加速度计不能把重力与真正的加速度相区分，所以要扣除地球重力 \mathbf{g} 的影响。因为地球是圆的，所以要补偿离心加速度项 $\boldsymbol{\omega}_{en}^n \times \mathbf{V}_{en}^n$ 。因为地球在自转，所以要补偿科氏加速度 $2\boldsymbol{\omega}_{ie}^n \times \mathbf{V}_{en}^n$ 。

加速度积分计算得到速度。速度积分计算得到位置。如果用经纬高表示位置，则有

$$\dot{L} = V_N / R_m \quad (3-22)$$

$$\dot{\lambda} = \frac{V_E}{R_p \cos L} \quad (3-23)$$

其中 L 和 λ 分别是纬度和经度， R_m 和 R_p 分别是当前位置的子午圈和卯酉圈半径。

地球是个椭圆，函数 `earthmodelupdate` 计算两个方向的半径和重力。

3.4. 组合导航

3.4.1. 原理概述

连续计算惯性导航；当获取卫星数据时，采用扩展卡尔曼滤波修正导航误差。

卡尔曼滤波可以理解为：根据方差求权重，做加权平均。

原始的卡尔曼滤波适用于线性系统。因为导航系统不是线性的，所以采用扩展卡尔曼滤波。扩展卡尔曼滤波的主要方法是，选用误差量，利用一阶微分近似为线性系统。滤波得到误差量估计值后，立刻补偿误差。

有的文献把EKF算法进一步细化为ESKF算法，严格意义上本代码包的方法属于ESKF算法。但是大量的文献没有把EKF算法进行如此细致的划分，本代码包的算法完全可以说就是EKF算法。

3.4.2. 卡尔曼滤波

比较复杂的系统中，一方面系统具有多个自由度，另一方面被测量随着时间而变化。因此用状态空间方程的形式描述系统的关系，并把加权平均数计算方法用矩阵表示，则得到卡尔曼滤波。

系统表示为：

$$\mathbf{x}_k = \boldsymbol{\Phi} \mathbf{x}_{k-1} + \mathbf{w}_{k-1} \quad (3-24)$$

$$\mathbf{z}_k = \mathbf{H} \mathbf{x}_k + \mathbf{v}_k \quad (3-25)$$

其中是 \mathbf{x} 状态量，是希望获得而又难以准确测量的量。式(3-24)描述了被测量的变化关系，这里是离散形式。 \mathbf{z} 表示量测量，是能测量得到但是包含随机误差的量。式(3-25)描述了量测量与状态量的关系。 \mathbf{w} 和 \mathbf{v} 是随机噪声。有的系统中 \mathbf{w} 和 \mathbf{v} 会乘以系数矩阵，但是大多数惯性导航装置的三轴传感器精度大体相当，因此没必要引入标准卡尔曼滤波的 $\boldsymbol{\Gamma}$ 矩阵。

状态量的变化也可以描述为连续方程

$$\dot{\mathbf{x}}_k = \mathbf{F} \mathbf{x}_{k-1} \quad (3-26)$$

如果采样间隔足够小，离散方程与连续方程的关系为

$$\Phi = I + FT \quad (3-27)$$

其中 T 为采样间隔， I 为单位矩阵。

卡尔曼滤波的解算过程就是根据 \mathbf{z} 估计 \mathbf{x} ，具体方法如下：

如果不考虑误差，前后时刻的 \mathbf{x} 具有关系

$$\hat{\mathbf{X}}_{k|k-1} = \Phi \hat{\mathbf{X}}_{k-1} \quad (3-28)$$

$\hat{\mathbf{X}}_{k-1}$ 是前一时刻 \mathbf{x} 的估计值， $\hat{\mathbf{X}}_{k|k-1}$ 是推算的后一时刻的 \mathbf{x} 。但是因为误差的存在，这个推算并不准确，需要根据 \mathbf{z} 修正，因此取

$$\hat{\mathbf{X}}_k = \hat{\mathbf{X}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{X}}_{k|k-1}) \quad (3-29)$$

其中 \mathbf{K}_k 是反映权重的滤波增益。这个增益由如下方法计算

$$\mathbf{P}_{k|k-1} = \Phi \mathbf{P}_{k-1} \Phi^T + \mathbf{Q} \quad (3-30)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}^T (\mathbf{H} \mathbf{P}_{k|k-1} \mathbf{H}^T + \mathbf{R})^{-1} \quad (3-31)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_{k|k-1} (\mathbf{I} - \mathbf{K}_k \mathbf{H})^T + \mathbf{K}_k \mathbf{R} \mathbf{K}_k^T \quad (3-32)$$

其中 \mathbf{P} 、 \mathbf{Q} 、 \mathbf{R} 分别是 $\hat{\mathbf{X}}$ 、 \mathbf{w} 、 \mathbf{v} 的方差矩阵。

上述公式给出了线性系统的卡尔曼滤波方法。非线性系统可以局部微分而近似为线性系统，采用扩展卡尔曼滤波方法解算。扩展卡尔曼滤波中的 \mathbf{x} 是误差量，扩展卡尔曼滤波获得误差量后，及时修正，使得误差量总维持在较小范围内；在误差量较小时，局部微分得到的线性系统与原始的非线性系统基本一致，卡尔曼滤波能取得较好效果。

代码包采用闭环反馈校正的方式，滤波后修正惯导误差，所以标准卡尔曼滤波中的 $\hat{\mathbf{X}}_{k-1}$ 取 0，简化后的计算公式为

$$\hat{\mathbf{X}}_k = \mathbf{K}_k \mathbf{z}_k \quad (3-33)$$

导航系统是非线性系统。取扩展卡尔曼滤波的状态量 \mathbf{x} 为 15 维向量，包含位置误差、速度误差、姿态误差、陀螺仪零偏、加速度计零偏各 3 各自由度，即

$$\mathbf{x} = [\delta L \quad \delta \lambda \quad \delta h \quad \delta v_E \quad \delta v_N \quad \delta v_U \quad \delta \phi_E \quad \delta \phi_N \quad \delta \phi_U \quad \varepsilon_x \quad \varepsilon_y \quad \varepsilon_z \quad \nabla_x \quad \nabla_y \quad \nabla_z]^T \quad (3-34)$$

用扩展卡尔曼滤波进行组合导航的步骤是：1. 进行惯性导航解算。2. 卫星修正时，比较惯性导航与卫星导航的结果偏差，即 \mathbf{z} 。3. 用卡尔曼滤波计算 \mathbf{x} 。4. 根据 \mathbf{x} 修正惯性导航的结果，并返回步骤 1。

3.4.3. 组合导航的状态矩阵

惯性和卫星组合导航系统关键在于具体列出状态矩阵 Φ ，即可实现组合导航的计算。扩展卡尔曼滤波的矩阵 \mathbf{F} 是雅可比矩阵，即偏微分矩阵。根据惯性导航的计算公式，可以得到 \mathbf{F} 如下。

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{pp} & \mathbf{F}_{vp} & \begin{bmatrix} \square & \square & \square \end{bmatrix} & \begin{bmatrix} \square & \square & \square \end{bmatrix} & \begin{bmatrix} \square & \square & \square \end{bmatrix} & \begin{bmatrix} \square & \square & \square \end{bmatrix} \\ \mathbf{F}_{pv} & \mathbf{F}_{vv} & \mathbf{F}_{av} & \begin{bmatrix} \square & \square & \square \end{bmatrix} & \mathbf{C}_b^n & \begin{bmatrix} \square & \square & \square \end{bmatrix} \\ \mathbf{F}_{pa} & \mathbf{F}_{va} & \mathbf{F}_{aa} & -\mathbf{C}_b^n & \begin{bmatrix} \square & \square & \square \end{bmatrix} & \begin{bmatrix} \square & \square & \square \end{bmatrix} \\ \begin{bmatrix} \square & \square & \square \end{bmatrix} & \begin{bmatrix} \square & \square & \square \end{bmatrix} & \begin{bmatrix} \square & \square & \square \end{bmatrix} & \mathbf{O}_3 & \begin{bmatrix} \square & \square & \square \end{bmatrix} & \begin{bmatrix} \square & \square & \square \end{bmatrix} \\ \begin{bmatrix} \square & \square & \square \end{bmatrix} & \begin{bmatrix} \square & \square & \square \end{bmatrix} & \begin{bmatrix} \square & \square & \square \end{bmatrix} & \begin{bmatrix} \square & \square & \square \end{bmatrix} & \mathbf{O}_3 & \begin{bmatrix} \square & \square & \square \end{bmatrix} \\ \begin{bmatrix} \square & \square & \square \end{bmatrix} & \begin{bmatrix} \square & \square & \square \end{bmatrix} & \begin{bmatrix} \square & \square & \square \end{bmatrix} & \begin{bmatrix} \square & \square & \square \end{bmatrix} & \begin{bmatrix} \square & \square & \square \end{bmatrix} & \begin{bmatrix} 0 & 1 & \square \\ \square & \square & \square \end{bmatrix} \\ \begin{bmatrix} \square & \square & \square \end{bmatrix} & \begin{bmatrix} \square & \square & \square \end{bmatrix} & \begin{bmatrix} \square & \square & \square \end{bmatrix} & \begin{bmatrix} \square & \square & \square \end{bmatrix} & \begin{bmatrix} \square & \square & \square \end{bmatrix} & \begin{bmatrix} \square & \square & 0 \\ \square & \square & \square \end{bmatrix} \end{bmatrix} \quad (3-35)$$

松组合的状态量为 15 维度，包括位置、速度、姿态、陀螺仪零偏、加速度计零偏各 3 个。而紧组合的状态量为 17 维度，多了接收机时间和时间漂移。

其中每个子矩阵都是 3 阶方阵， \mathbf{O}_3 表示 0 矩阵。

反映位置误差对位置误差影响的子矩阵为

$$\mathbf{F}_{pp} = \begin{bmatrix} 0 & 0 & -\frac{v_N}{(R_m + h)^2} \\ \frac{v_E \sec L \tan L}{R_p + h} & 0 & -\frac{v_E \sec L}{(R_p + h)^2} \\ 0 & 0 & 0 \end{bmatrix} \quad (3-36)$$

反映速度误差对位置误差影响的子矩阵为

$$\mathbf{F}_{vp} = \begin{bmatrix} 0 & \frac{1}{R_m + h} & 0 \\ \frac{\sec L}{R_p + h} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3-37)$$

反映位置误差对速度误差影响的子矩阵为

$$\mathbf{F}_{pv} = \begin{bmatrix} 2\omega_e v_N \cos L + 2\omega_e v_U \sin L + \frac{v_N v_E \sec^2 L}{R_p + h} & 0 & \frac{v_U v_E - v_N v_E \tan L}{(R_p + h)^2} \\ -\left(2\omega_e v_E \cos L + \frac{v_E^2 \sec^2 L}{R_p + h}\right) & 0 & \frac{v_N v_U}{(R_m + h)^2} + \frac{v_E^2 \tan L}{(R_p + h)^2} \\ -2v_E \omega_e \sin L & 0 & -\frac{v_N^2}{(R_m + h)^2} - \frac{v_E^2}{(R_p + h)^2} \end{bmatrix} \quad (3-38)$$

反映速度误差对速度误差影响的子矩阵为

$$\mathbf{F}_{vv} = \begin{bmatrix} \frac{v_N \tan L - v_U}{R_p + h} & 2\omega_e \sin L + \frac{v_E \tan L}{R_p + h} & -2\omega_e \cos L - \frac{v_E}{R_p + h} \\ -2\omega_e \sin L - \frac{2v_E \tan L}{R_p + h} & \frac{-v_U}{R_m + h} & \frac{-v_N}{R_m + h} \\ 2\left(\omega_e \cos L + \frac{v_E}{R_p + h}\right) & \frac{2v_N}{R_m + h} & 0 \end{bmatrix} \quad (3-39)$$

反映姿态误差对速度误差影响的子矩阵为

$$\mathbf{F}_{av} = \begin{bmatrix} 0 & -f_U & f_N \\ f_U & 0 & -f_E \\ -f_N & f_E & 0 \end{bmatrix} \quad (3-40)$$

其中 f_E 、 f_N 、 f_U 是换算到n系的加速度计数值，即不扣除重力的比力信息。

$$\mathbf{f}_n = \begin{bmatrix} f_E \\ f_N \\ f_U \end{bmatrix} = \mathbf{C}_b^n \mathbf{f}_b \quad (3-41)$$

反映位置误差对姿态误差影响的子矩阵为

$$\mathbf{F}_{pa} = \begin{bmatrix} 0 & 0 & \frac{v_N}{(R_m + h)^2} \\ -\omega_e \sin L & 0 & \frac{-v_E}{(R_p + h)^2} \\ \omega_e \cos L + \frac{v_E \sec^2 L}{R_p + h} & 0 & \frac{-v_E \tan L}{(R_p + h)^2} \end{bmatrix} \quad (3-42)$$

反映速度误差对姿态误差影响的子矩阵为

$$\mathbf{F}_{va} = \begin{bmatrix} 0 & -\frac{1}{R_m + h} & 0 \\ \frac{1}{R_p + h} & 0 & 0 \\ \frac{\tan L}{R_p + h} & 0 & 0 \end{bmatrix} \quad (3-43)$$

反映姿态误差对姿态误差影响的子矩阵为

$$\mathbf{F}_{aa} = \begin{bmatrix} 0 & \omega_e \sin L + \frac{v_E \tan L}{R_p + h} & -\omega_e \cos L - \frac{v_E}{R_p + h} \\ -\omega_e \sin L - \frac{v_E \tan L}{R_p + h} & 0 & -\frac{v_N}{R_m + h} \\ \omega_e \cos L + \frac{v_E}{R_p + h} & \frac{v_N}{R_m + h} & 0 \end{bmatrix} \quad (3-44)$$

导航计算机每次收到惯性数据时，要计算 \mathbf{F} 矩阵，并更新 Φ 矩阵。导航计算机收到卫星数据时再进行卡尔曼滤波解算，并根据滤波计算得到的误差量修正导航结果。

3.4.4. 组合导航的观测矩阵

松组合采用位置、速度作为观测量，而紧组合采用伪距作为观测量。观测矩阵 \mathbf{H} 相应变化。

距离对坐标的偏微分为：

$$r_1 = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2} \quad (3-45)$$

$$\frac{\partial r_1}{\partial x_1} = \frac{x_1 - x_0}{r_1} \quad (3-46)$$

相应的偏微分矩阵为

$$\mathbf{H}_r = \begin{bmatrix} \frac{x_1 - x_0}{r_1} & \frac{y_1 - y_0}{r_1} & \frac{z_1 - z_0}{r_1} \\ \vdots & \vdots & \vdots \\ \frac{x_n - x_0}{r_n} & \frac{y_n - x_0}{r_n} & \frac{z_n - x_0}{r_n} \end{bmatrix} \quad (3-47)$$

惯性导航计算时，采用纬度经度高度表示位置；但是紧组合时，需要把地理系的位置和速度转换为 ECEF 系的 XYZ 坐标。使用代码包的函数 `n2e.m` 实现转换，并计算转换对应的偏微分矩阵，位置转换偏分矩阵 \mathbf{M}_p ，以及速度转换偏微分矩阵 \mathbf{M}_v 。

$$\mathbf{M}_p = \begin{bmatrix} -(R_e + h) \sin L \cos \lambda & -(R_e + h) \cos L \sin \lambda & \cos L \cos \lambda \\ -(R_e + h) \sin L \sin \lambda & (R_e + h) \cos L \cos \lambda & \cos L \sin \lambda \\ (R_e(1 - f)^2 + h) \cos L & 0 & \sin L \end{bmatrix} \quad (3-48)$$

$$\mathbf{M}_v = \begin{bmatrix} -\sin \lambda & -\sin L \cos \lambda & \cos L \cos \lambda \\ \cos \lambda & -\sin L \sin \lambda & \cos L \sin \lambda \\ 0 & \cos L & \sin L \end{bmatrix} \quad (3-49)$$

根据上述公式改写卡尔曼滤波的 \mathbf{H} 矩阵。

$$\mathbf{H}_r = \begin{bmatrix} \mathbf{H}_r \mathbf{M}_p & \mathbf{O} & \mathbf{O} & \mathbf{c} & \mathbf{O} \\ \mathbf{O} & \mathbf{H}_r \mathbf{M}_v & \mathbf{O} & \mathbf{O} & \mathbf{c} \end{bmatrix} \quad (3-50)$$

其中 \mathbf{c} 为光速，但是扩充为列向量。

4. 代码包预览

4.1. 部分代码截图

（由于版本迭代，实际代码可能与截图有轻微差别）

```

cspeed=299792458;%光速

%% 计算轨迹、伪距、伪距率
for k=1:L
    gyro=w(k,:)' ;
    Cbn=cbn(atti1);
    acc=Cbn*(an(k,:)'+[0;0;ge]);
    [atti1, speed1, pos1, accn1]=insgyroacc(gyro, acc, atti1, speed1, pos1, dTins);%给出轨

    [pb, ~, vb, ~]=n2e(pos1, speed1);%载体在e系的速度位置
    d=zeros(ns, 2);%伪距, 伪距率
    for p=1:ns
        rbs=pb-ps(p,:)' ;
        d(p, 1)=norm(rbs)+saterr(p, 1)+saterr(p, 2)+saterr(p, 3)+rt*cspeed;
        vbs=vb-vs(p,:)' ;
        d(p, 2)=norm(vbs)+rtd*cspeed;
    end

    data(k, :)= [gyro', acc', d(:, 1)', d(:, 2)', reshape(ps', 1, ns*3), reshape(vs', 1, ns*3)

    ps=ps+vs*dTins;
    rt=rt+rtd*dTins;
end

%根据需要下面加误差

```

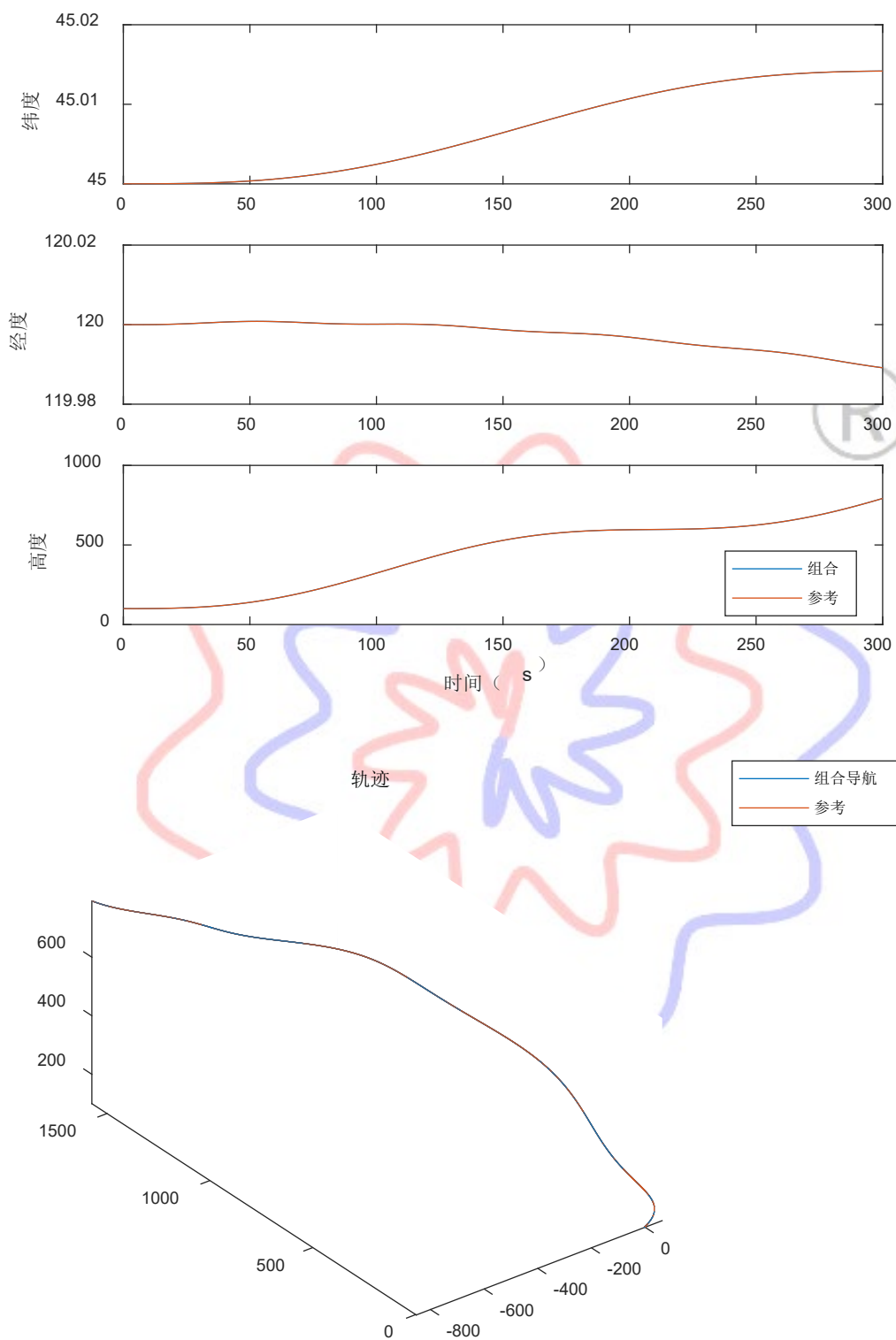


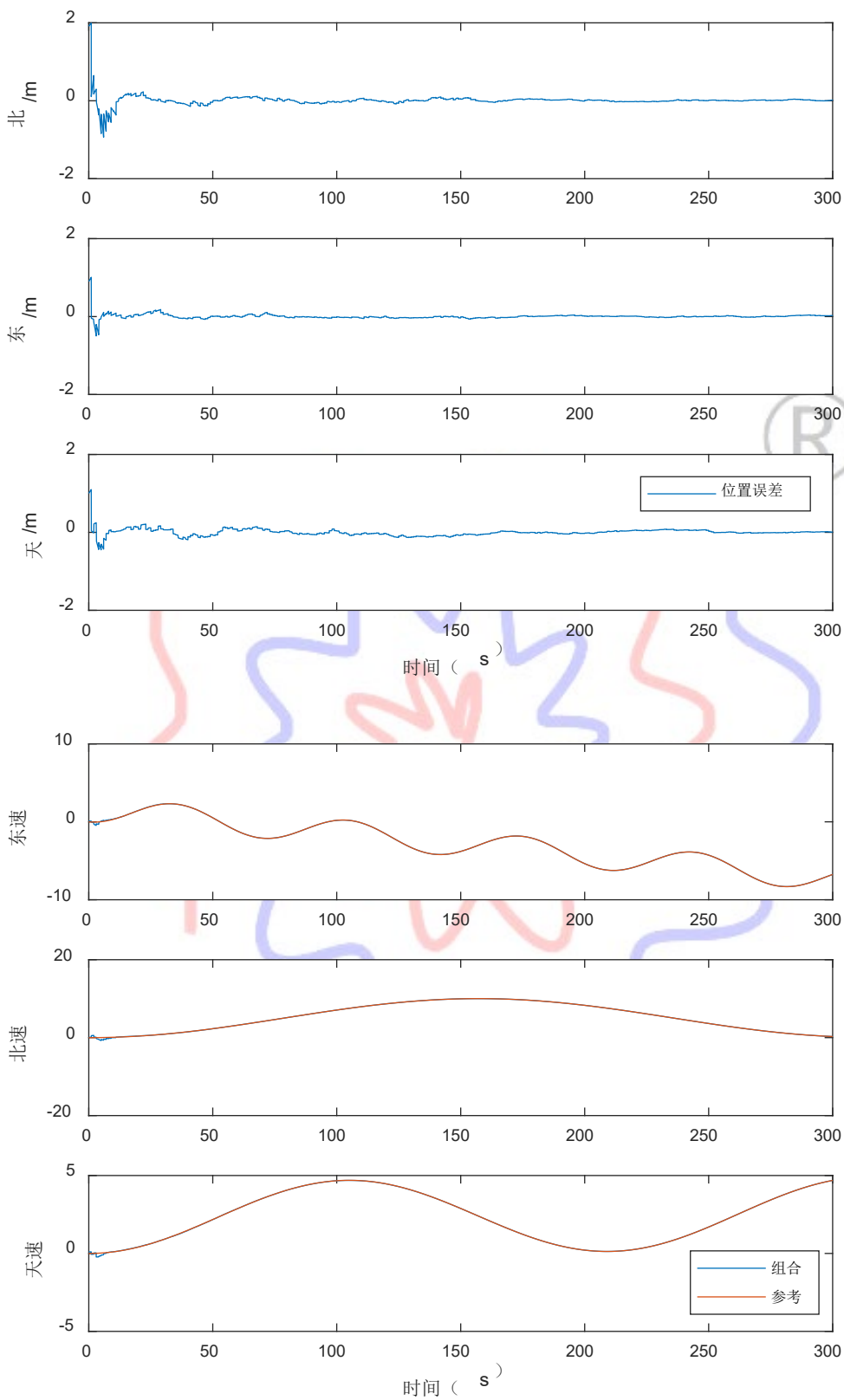
```

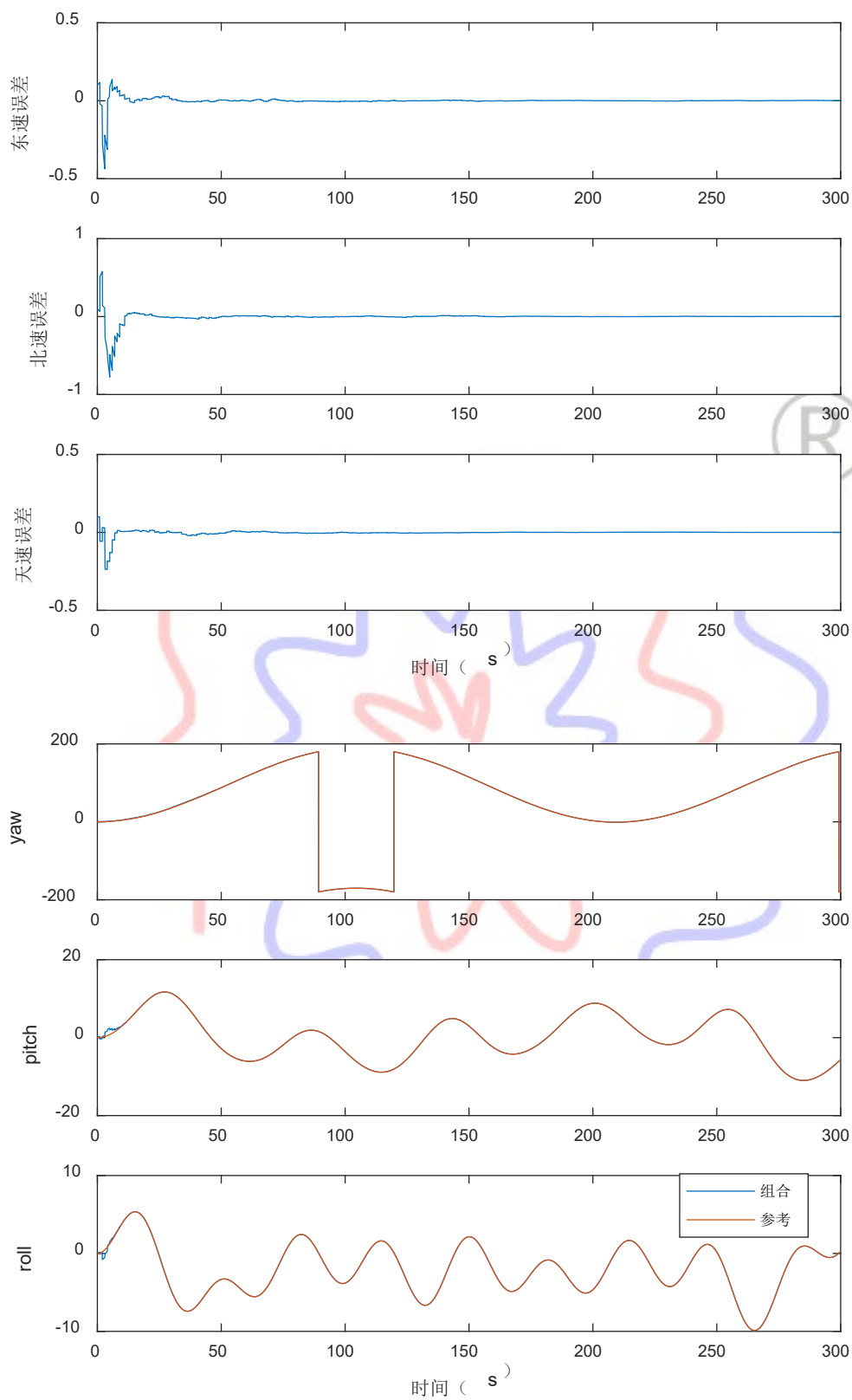
58
59 -     if(data(k, 84)==1)
60
61 -         d=[data(k, 6+(1:ns))', data(k, 6+ns+(1:ns))'];%伪距伪距率
62 -         ps=reshape(data(k, 6+2*ns+(1:(3*ns))), 3, ns)';%卫星位置
63 -         vs=reshape(data(k, 6+5*ns+(1:(3*ns))), 3, ns)';%卫星速度
64 -         saterr=reshape(data(k, 6+8*ns+(1:(3*ns))), 3, ns)';%卫星修正量
65
66 -         [pb, Mp, vb, Mv]=n2e(pos1, speed1);%载体位置速度转到e系
67
68 -         H=zeros(2*ns1, 17);
69 -         Z1=zeros(2*ns1, 1);
70
71 -         Hr=zeros(ns1, 3);
72
73 -         for p=1:1:ns1
74 -             pbs=pb-ps(p, :)' ;
75 -             abspbs=norm(pbs);
76 -             Z1(p)=abspbs-(d(p, 1)-saterr(p, 1)-saterr(p, 2)-saterr(p, 3))+csp;
77 -             vbs=vb-vs(p, :)' ;
78 -             Z1(ns1+p)=norm(vbs)-d(p, 2)+cspeed*biasdt;
79 -             Hr(p, :)=pbs'/abspbs;
80 -         end
81
82 -         H1=[Hr*Mp, zeros(ns1, 12), ones(ns1, 1)*cspeed, zeros(ns1, 1);
83 -             zeros(ns1, 3), Hr*Mv, zeros(ns1, 9), zeros(ns1, 1), ones(ns1, 1)*csp;
84 -             R=[eye(ns1)*0.01, zeros(ns1); zeros(ns1), eye(ns1)*0.04];
85
86 -         %一、计算姿态
87 -         wien=twe(pos);
88 -         wenn=twv(pos, speed, Rmeri, Rprim);
89 -         Cbn=cbn(atti);
90 -         wnbb=gyrol-Cbn*(wien+wenn);%扣除地球自转、扣除速度
91 -         atti=qupdate(atti, wnbb*dTins);%更新姿态
92
93 -         %二、计算速度
94 -         accn=Cbn*acc1;%更新这个数，以便于卡尔曼滤波的部分使
95
96 -         gn=[0;0;-ge];
97 -         an=accn-cross(wien+wien+wenn, speed)+gn;
98 -         speed=speed+dTins*an;%更新速度
99
100 -         %三、计算位置
101 -         dpos=zeros(3, 1);
102
103 -
104 -
105 -
106 -
107 -
108 -
109 -
110 -
111 -
112 -
113 -
114 -
115 -
116 -
117 -
118 -
119 -
120 -
121 -
122 -
123 -
124 -
125 -
126 -
127 -
128 -
129 -
130 -
131 -
132 -
133 -
134 -
135 -
136 -
137 -
138 -

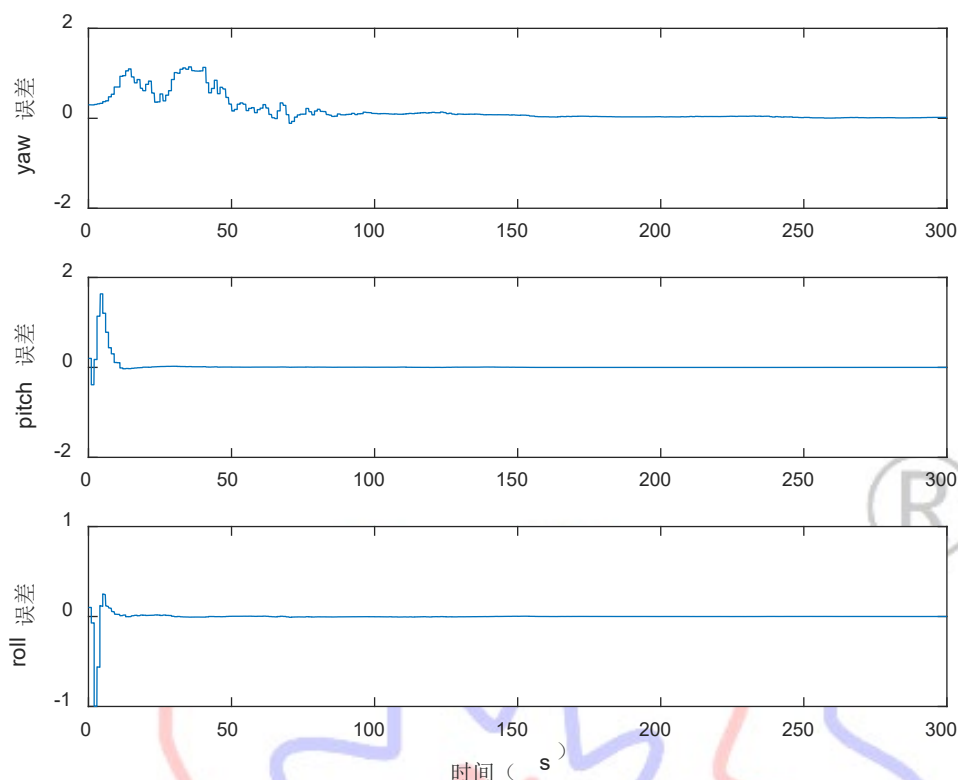
```

4.2. 输出数据曲线









5. 常见问题

5.1. 代码容易看懂吗

本代码为了便于学习理解，力求简洁清楚。但是代码包技术含量较高，本店不保证用户能理解代码。用户可以参照前文的代码截图，预先评估自己是否可以理解代码包的内容。

为了防止非法转卖，代码包的注释比较简化。如有难以理解的部分，可以联系本店人工答疑。

本代码包是紧组合算法，难度较大。建议初学者先学习松组合算法（本店另售），待理解松组合算法后再购买紧组合算法。

5.2. 组合导航比卫星导航更准吗

组合导航结果基本相当于卫星导航结果的平滑。对于低精度 IMU，组合结果甚至可能不如卫星导航准确。

组合导航与卫星导航相比，主要优势有：1.组合导航的数据刷新率更高。2.组合导航可以输出姿态信息。

5.3. 算法的精度是多少

对于绝大多数情况，组合导航的精度主要取决于传感器精度，算法精度不是导航结果精度的瓶颈。如果用户得到的导航结果精度偏低，应当优先检查传感器精度。

代码包处理附带数据的精度，不能代表代码包处理用户数据的精度。

真实实验数据的绝对精度难以评估。如果用户期望在实验中评估绝对精度，有几种可行的方案：1.采用更高精度的导航设备作为位置基准，与自制设备一同实验；比如采用成品高精度组合导航装置，或采用 RTK 卫星导航装置。2.测绘特定轨道位置，沿着特定轨道运动，比如摄像轨道车。

5.4. 怎样判断组合导航的结果是否正确

用户可以尝试故意把代码包的初始姿态改错、然后再运行，以观察组合导航异常时的计算结果，直观体会组合导航结果正常与异常的差别。

5.5. 代码可以处理自己的实验数据吗

本代码包难以处理实验数据，主要原因在于实验数据过于五花八门。卫星导航中可能的系统误差包括：信号飞行时间引起的卫星轨道位置误差，地球自转 Sagnac 效应引起的误差，卫星的钟差，电离层误差，对流层误差，等等。用户数据中，有时候补偿了某些误差，有时候没有补偿某些误差。如果不能明确这些误差的状态，可能发生漏补偿或重复补偿，导致组合导航结果异常。

可以委托本店处理紧组合导航实验数据（另行收费）。

5.6. 卫星数量很少的时候可以进行紧组合导航吗

卫星导航通常要求可见卫星数大于等于 4 个。

如果可见卫星数量小于 4 个，在一定条件下可以进行组合导航，但是组合导航效果会有所折扣。可行的情况有：

(1) 可见卫星数量较多，卫星接收机内部已经补偿了接收机钟差；但是只选取小于等于 4 个卫星的伪距、伪距率用于组合导航。

(2) 大部分时候可见卫星数量较多，已经基本补偿了接收机钟差；但是偶尔卫星数量较少，小于 4 个。

难以实现组合导航的情况是：自始至终可见卫星数量小于 4 个，接收机钟差没有充分补偿。

5.7. 坐标系定义不一致怎么办

本代码包采用东北天坐标系。如果有人采用北东地、北天东等，也可以使用本代码包计算，但是需要稍作修改。不同坐标系定义，影响姿态四元数或欧拉角的定义，即重写公式(3-6)，不影响其余计算。采用其他坐标系，只需要换算初始姿态、输出姿态角即可；不要换算传感器轴向等无关量。

5.8. 代码依据哪个参考资料的原理

本店正在筹备出版相关书籍，预计 2025 年出版。

暂时可以参考：Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration (作者 Aboelmagd Noureldin, Tashfeen B. Karamat, Jacques Georgy, 出版商 Springer) (此书国内亦有翻译版)。代码与此书不完全一致。

本店另行录制了教学视频，陆续上传 B 站。

5.9. 怎样调卡尔曼滤波参数

根据大量用户的实践结果，结果异常的情况中 80% 以上可以归结于其余部分的错误，参数不合适造成的异常只占很少的比例。此外，算法对于卡尔曼滤波参数不敏感，通常情况下即使参数调整 10 到 100 倍，仍然能得到正常的计算结果。所以，当结果显著异常时，应当优先检查其余部分是否正确，而不是调整参数。

P_k 是初始误差的平方。 R 是卫星的误差的平方。 P_k 和 R 都能直接计算得到，通常不必大调。注意单位换算。例如，如果初始位置北向误差大约 10 米，那么 P_k 的第一个数取 $(10/R_e)^2$ ，其中 R_e 是地球半径。其余数值计算方法类似。

Q 反映了惯导噪声等误差项的影响，如速率随机游走等。理论上，根据传感器艾伦方差曲线，推算角速率随机游走等误差项的数值，进一步计算得到 Q 的值。但是实际上，因为惯导的误差并没有全部纳入 15 维卡尔曼滤波的方程中，比如不交误差、 g 值敏感系数、温度系数等等，所以不宜直接根据理论推算 Q 值。实践中，采用人工调参法比较方便。

如果惯导的传感器零偏误差较大，可以调大 P_k 和 Q 的第 7 到 12 个数值。如果卫星导航波动较大，则 Q 可以小一点， R 可以大一点。反之亦然。

5.10. 卫星导航只有伪距没有伪距率怎么办

此时需要微调代码。设可见卫星数为 n ，则卡尔曼滤波参数 R 改为 $n*n$ 矩阵；观测量 Z_1

改为 $n \times 1$ 矩阵；观测方程 H1 改为 $n \times 17$ 矩阵。如果不修改代码，而是直接取伪距率为 0，会导致计算结果异常，不要这样做。

6. 著作权和服务

6.1. 工作原理参考什么资料

参考实体书《组合导航应用笔记》，东南大学出版社，2025 年。

讲解视频，哔哩哔哩视频网搜索“大胡子刘师傅”。

6.2. 著作权声明

本店保留著作权。

电路、说明书、全部附属代码（以下简称本代码包）仅限于学习和研究用途的少量使用；包含改编文件、写入嵌入式系统的编译后程序，所有副本总计不得超过 5 份。

本代码包有偿使用。

严禁转卖或公开发布本代码包的全部或一部分。

大规模应用本代码包需要额外取得本店的授权。

对于违反上述要求的用户，本店有权要求停止销售、撤稿、赔偿损失等。

6.3. 服务内容

赠送 30 分钟语音答疑服务，用于解决较为复杂的疑问。

赠送长期文字答疑，用于解决简单的、零散的疑问。

答疑服务仅限直接购买人本人使用。答疑服务不能转让、不能共享。用户需要保留购买凭证截图；丢失购买凭证的，本店可以不提供答疑服务；不是从本店购买的，而是从其他渠道获得代码包的，不提供答疑服务。

本商品技术含量较高，本店不保证能在限时内解答所有疑问。有需要的用户，可以付费购买额外的语音答疑服务。

本店可提供少量的数据判读服务。但是大量的数据判读服务需要额外收费。较为复杂的数据处理，或者定制化修改代码，可能需要额外收费。

上述服务可能需要排队，本店不能保证服务的实时性。

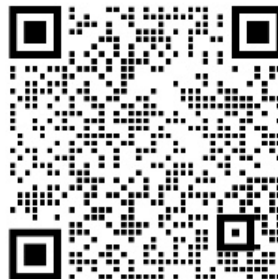
6.4. 联系方式

西安市雁塔区雾膜软件开发站

销售、答疑、定制开发：

微信：（扫码）

雾膜软件



电子邮箱：braun@wmsoft.wang

网站：<http://wmsoft.xyz>