

# 雾膜软件

## 传递对准仿真 Matlab

版本 20240824

### 1. 内容介绍

#### 1.1. 概述

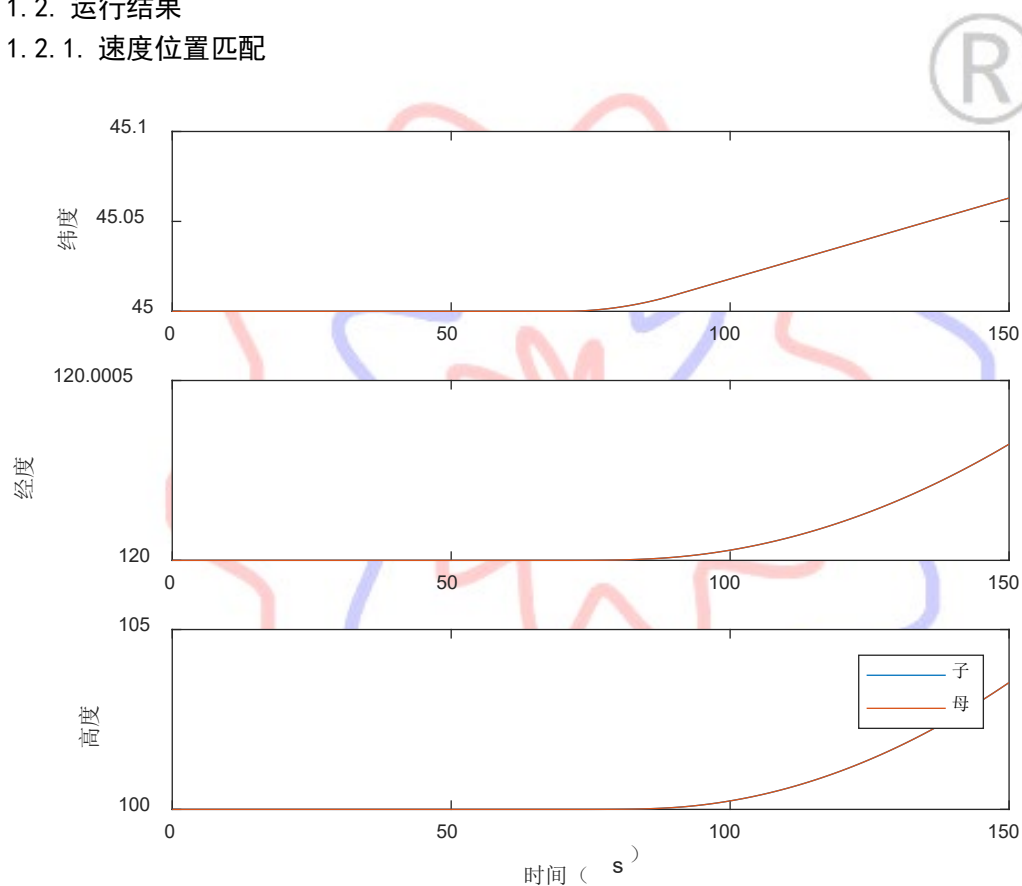
代码包包含 2 个文件夹，分别为速度位置匹配，和姿态匹配，两种方法的传递对准。

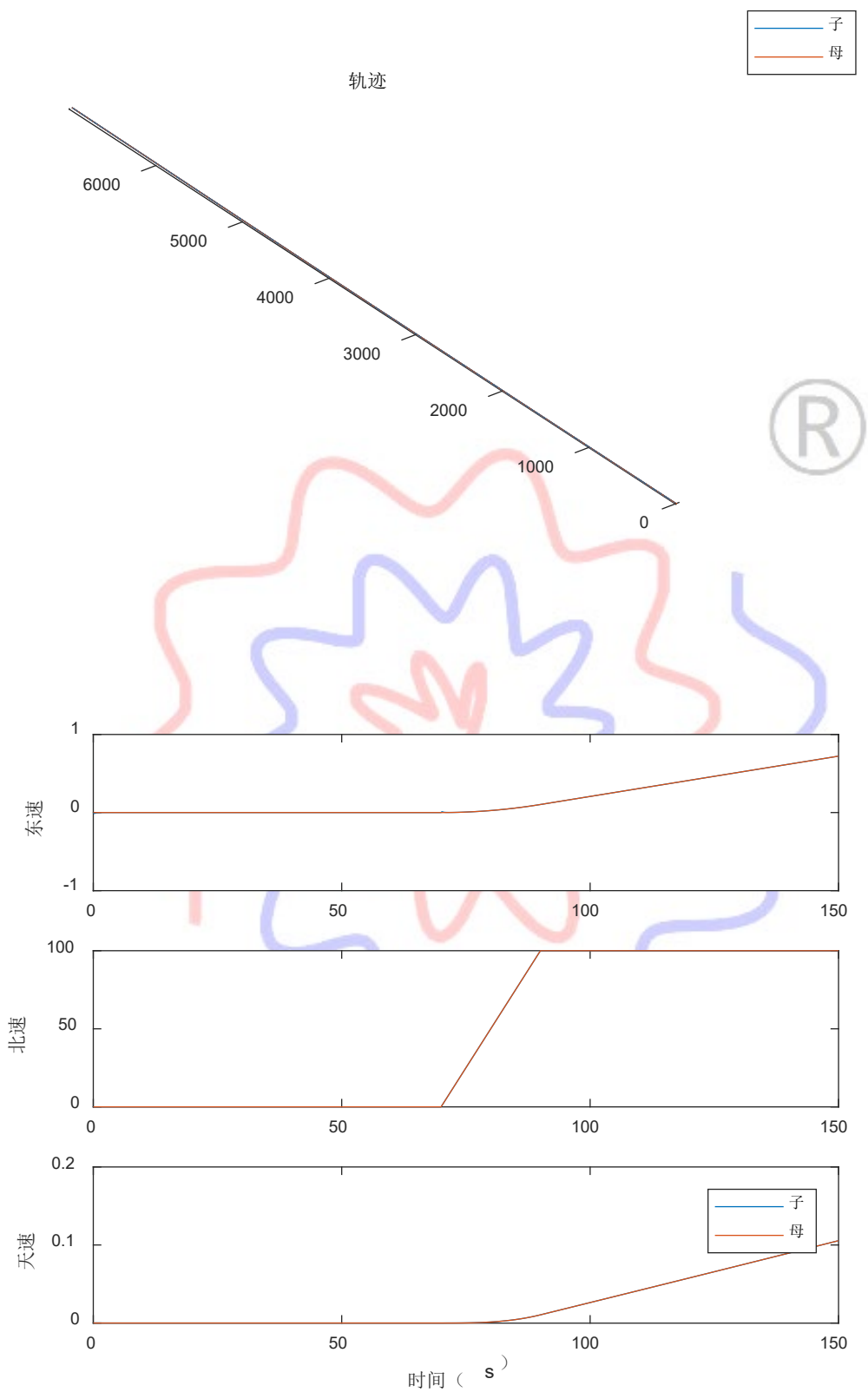
直接运行 `creatdata.m` 生成仿真数据。包括子惯导角速度、加速度、参考姿态，主惯导姿态、速度、位置。

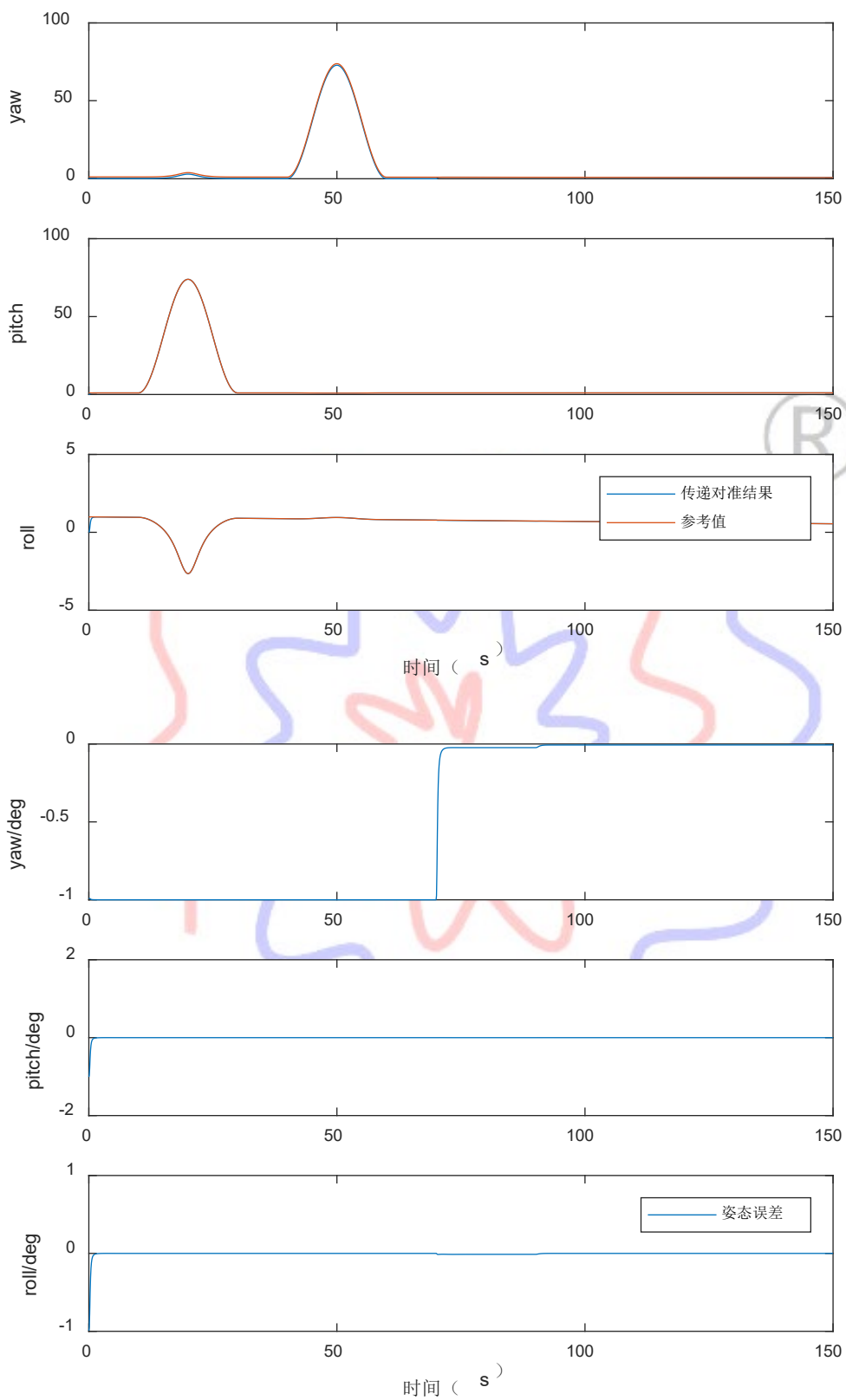
直接运行 `instance1.m` 计算传递对准。

#### 1.2. 运行结果

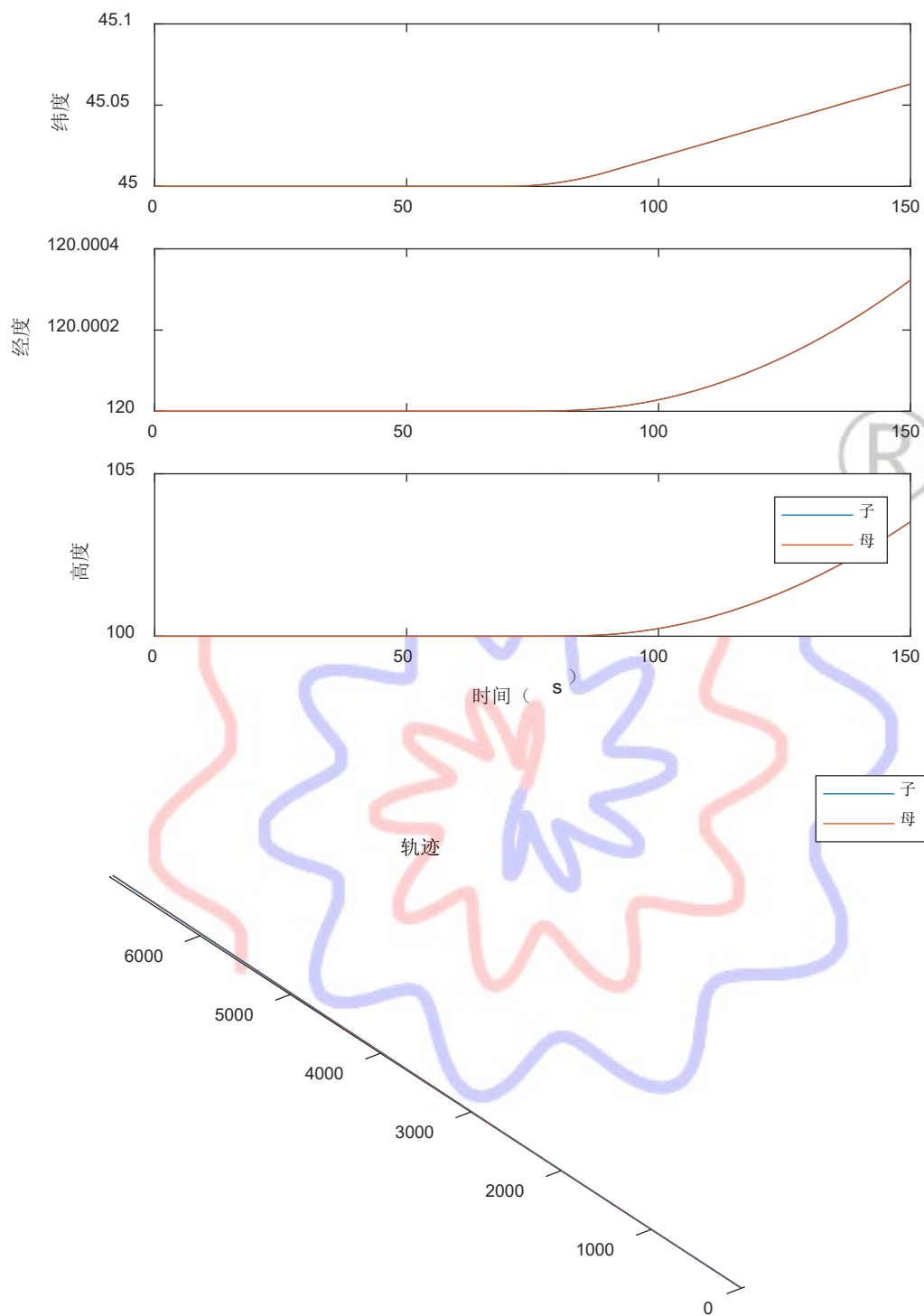
##### 1.2.1. 速度位置匹配

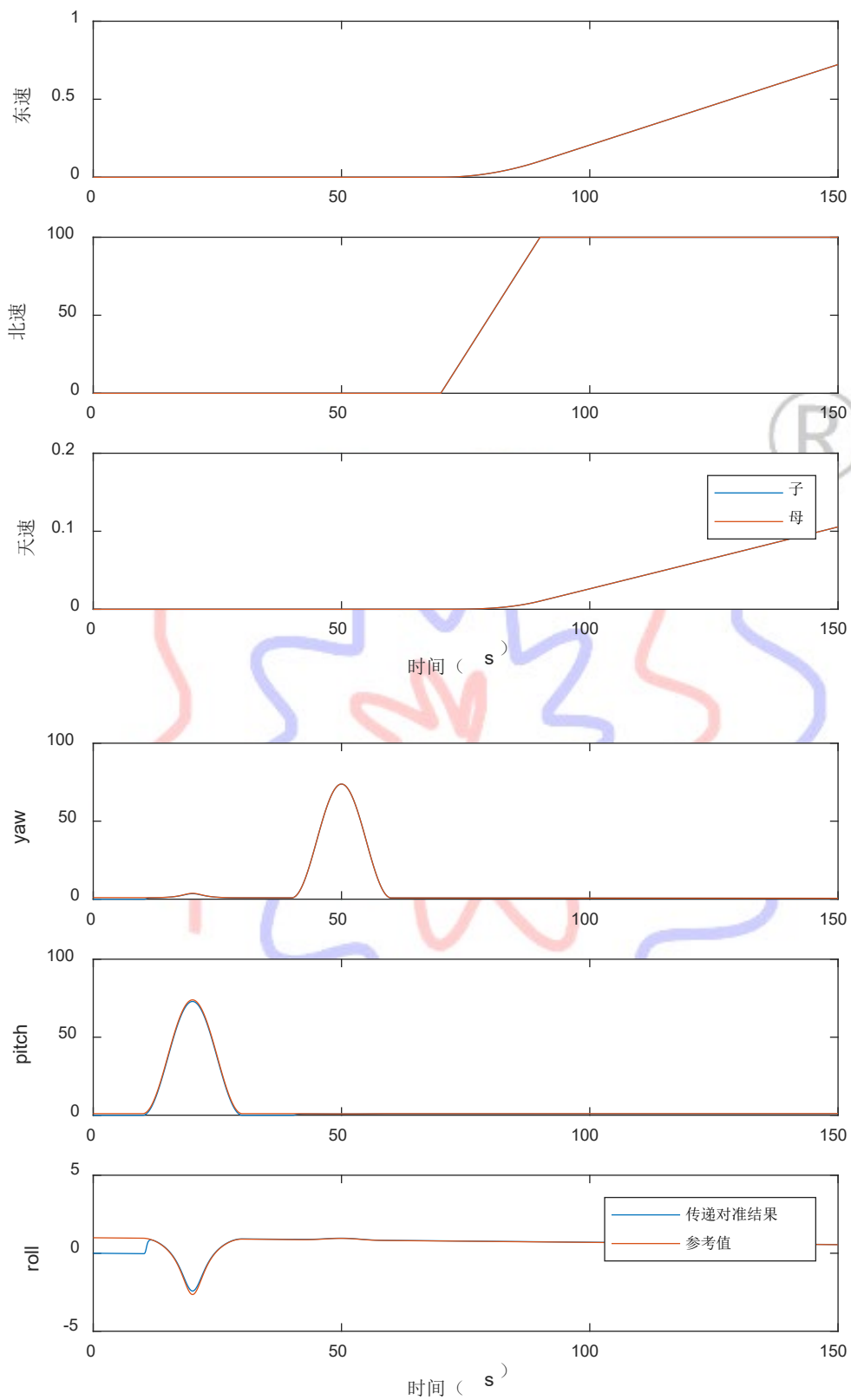


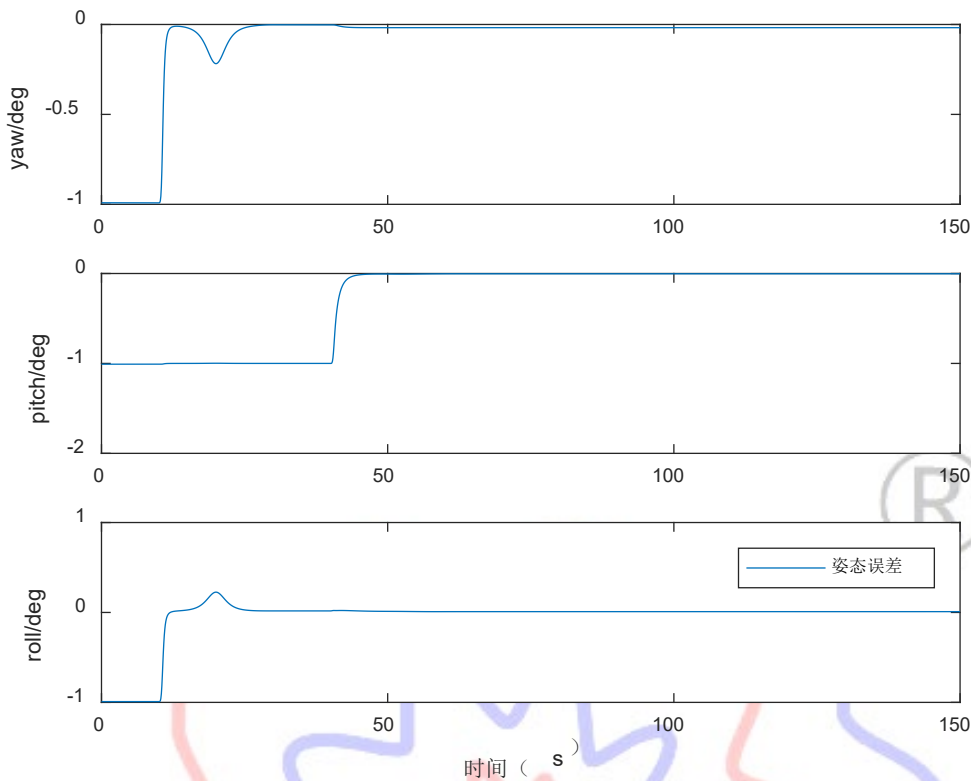




1.2.2. 姿态匹配







### 1.2.3. 结果分析

生成仿真数据时，子惯导和主惯导之间设置了固定的姿态差异。在传递对准之后，子惯导的姿态误差逐渐收敛，证明了传递对准方法有效。

### 1.3. 部分代码截图

```
1 %主惯导真数据
2 clear
3 rng(0);
4
5 %生成系统
6 dflms=0.005;
7 L=30000;
8 t=(1:L)*dflms;
9 s=sin(2*pi*(1:4000)/4000); %一个机动数据
10 w=zeros(L,3);
11 am=zeros(L,3);
12
13
14 w(1000:4000,1)=s*d0.2;
15 w(5001:10000,3)=s*d0.2;
16 am(14001:18000,2)=s;
17
18 ccm=cnb(qupdate([1;0;0;0], [1;1;1]/180*pi)); %主子惯导
19 cam=ccm %每颗子的转换矩阵
20
21 attil=seteula(0,0,0);
22 posi=[45/180*pi; 120/180*pi; 100];
23 speedi=zeros(3,1);
24 ["", ga]=earth2dofupdate(posi);
25
26 datai=zeros(L,18); %子惯导陀螺仪、加速度计、姿态、母惯导
27
28
29 %子惯导系统
30 for k=1:L
31     gyro=cnb(k,:);
32     Cbn=cnb(attil);
33     acc=cm*(am(k,:)+0.0*ga);
34     [attil,speedi,posi,accni]=insgyroacc(gyro,acc,cm);
35
36 %子惯导系统
37 gyro=cm*gyro;
38 acc=cm*acc;
39 cam=cnb(attil);
40 ccm=cam*ccm;
41 cnc=cm;
42
43     data(k,:)=['gyro',acc',getoulacnb(cnc)',getoulacnb(cnc)'];
44 end
45
46 %根据需要在下面加误差
47 % data(:,1)=data(:,1)+(1e-5)*randn(L,1)*(1e-6);
48 % data(:,2)=data(:,2)+(2e-5)*randn(L,1)*(1e-6);
49 % data(:,3)=data(:,3)+(1e-5)*randn(L,1)*(1e-6);
50 % data(:,4)=data(:,4)+(1e-5)*randn(L,1)*(1e-5);
51 % data(:,5)=data(:,5)+(2e-4)*randn(L,1)*(1e-5);
52 % data(:,6)=data(:,6)+(-1e-4)*randn(L,1)*(1e-5);
53
54 save('data.mat','data');
55
56 %数据保存
57 data(k,1:9)=[getoulacnb(attil)', speedi',posi'];
58 data(k,10:12)=ZI';
59 data(k,16:24)=XI';
60
61 %数据加工成标准格式
62 clear
63 close all
64
65 %% 卡尔曼滤波参数
66 Phi=diag([3e-4,3e-4,3e-4,3e-4,3e-4,0,0,0]);
67 Q=diag([1e-8,1e-8,1e-8,0,0,0,1e-12,1e-12,1e-12]);
68 R=diag([1e-6,1e-6,1e-6]);
69 QI=zeros(9);
70 XI=zeros(9,1);
71
72 %% 初始值
73 attil=seteula(0,0,0);
74 posi=[45/180*pi; 120/180*pi; 100];
75 speedi=zeros(3,1);
76 pos=posi;
77
78 dflms=0.005;
79 load('data.mat');
80 L=length(data);
81 datai=zeros(L,30);
82
83 biasgyro=zeros(3,1);
84 biasacc=zeros(3,1);
85 attie=[1;0;0;0];
86
87 %% 子惯导
88 for k=1:L
89     gyro=data(k,1:3)';
90     acc=data(k,4:6)';
91     [attil,"",accni]=insgyroacc(gyro,acc,attil,speedi,posi,dflms,biasacc,biasgyro);
92     Phi=stateupdate(dflms,attil,speedi,posi,accni,Phi);
93     QI=Q+dflms;
94     speedi=data(k,13:15)';
95     pos=data(k,16:18)';
96
97     if (1)
98         Cnm=cnb(seteula(data(k,10),data(k,11),data(k,12)));
99         Cbn=cnb(attil)';
100         Ccm=Cbn*Cnm;
101         am=Ccm*am;
102         ZI=as(3,2)*ms(2,3);ms(1,3)*ms(3,1);ms(2,1)*ms(1,2)'/2;
103         HI=[eye(3),-Cam,zeros(3,2)];
104
105         [XI,Phi]=kal(ZI,HI,Phi,QI,R,Phi);
106         Phi=eye(9);
107         QI=zeros(9);
108         attie=update(attie,cbn(attil))'*XI(1:3,1);
109         attie=update(attie,XI(4:6));
110         biasgyro=biasgyro-XI(7:9);
111     end
112
113     %数据保存
114     data(k,1:9)=[getoulacnb(attil)', speedi',posi'];
115     data(k,10:12)=ZI';
116     data(k,16:24)=XI';
117
118 %位置对速度影响的子矩阵
119 Ppp=[0, 0, 0; 0, 0, 0; 0, 0, 0];
120 P(1:3,1:3)=Ppp;
121
122 %速度对位置影响的子矩阵
123 Pvp=[0, 0, 0; 0, 0, 0; 0, 0, 0];
124 P(4:6,4:6)=Pvp;
125
126 %位置对位置影响的子矩阵
127 Ppp=[0, 0, 0; 0, 0, 0; 0, 0, 0];
128 P(1:3,1:3)=Ppp;
129
130 %速度对速度影响的子矩阵
131 Pvv=[(dt*tanhi-dt)*RgRl, 2.0*dt*tanhi*dRgRl*dtanhi, (-2.0)*dt*cosphi*(-RgRl)*U;
132 (-2.0)*dt*(vt*cosphi+dRgRl*dtanhi), (-RgRl)*U, 2.0*dt*cosphi*(-RgRl)*dtanhi;
133 (-2.0)*dt*(vt*cosphi+dRgRl*dtanhi), 2.0*dt*cosphi, 0];
134 P(4:6,4:6)=Pvv;
135
136 %姿态对速度影响的子矩阵
137 Fv=[0, -FU, FV;
138 -FU, 0, -FV;
139 -FR, FE, 0];
140 P(4:6,7:9)=Fv;
141
142 %位置对姿态影响的子矩阵
143 Ppp=[0, 0, 0; (-vt)*sinphi, 0, (-dt)*RgRl*dtanhi;
144 (vt)*sinphi, 0, (-dt)*RgRl*dtanhi];
145 P(1:3,10:12)=Ppp;
```

## 2. 计算原理

### 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。

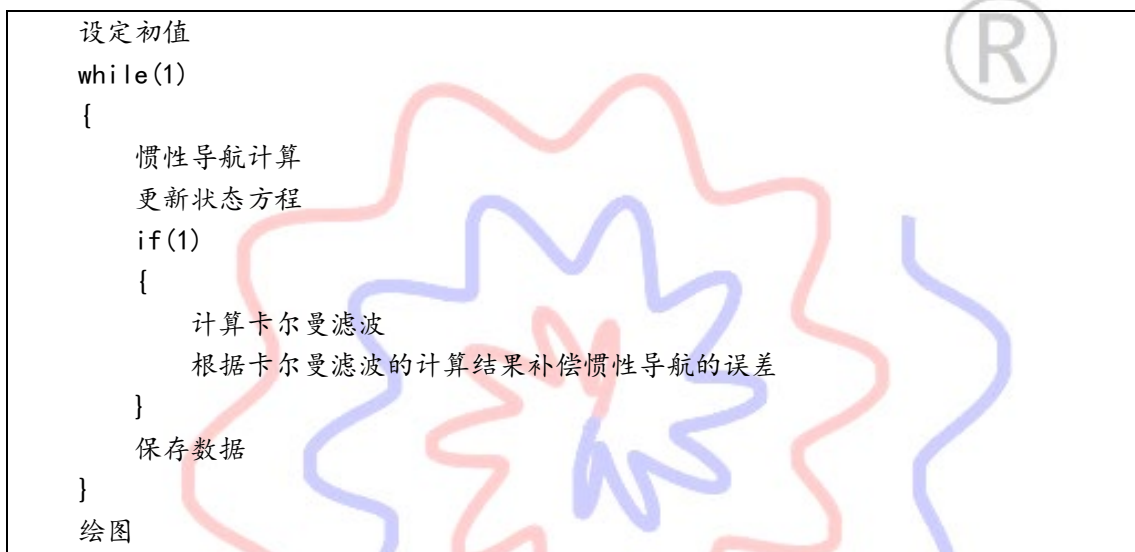
组合导航中，扩展卡尔曼滤波的状态变量定义为 15 维度的误差量：纬经高，东北天速度，东北天姿态，三轴陀螺仪零偏，三轴加速度计零偏。

若无特殊说明，一般采用国际单位制。角度单位为  $\text{rad}$ ，角速度单位为  $\text{rad/s}$ ，速度单位  $\text{m/s}$ ，加速度单位  $\text{m/s/s}$ 。

### 2.2. 主程序

主程序为 `instance1.m`。直接运行即可。

程序主要工作流程为：



### 2.3. 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，可以不考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  $e$ ，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  $\beta$  相对于坐标系  $\alpha$  的变化量  $x$  在坐标系  $\gamma$  的投影表示为  $x_{\alpha\beta}^{\gamma}$ 。例如，地球自转在地理系的坐标为

$$\omega_{ie}^t = \begin{bmatrix} 0 \\ \omega_e \cos L \\ \omega_e \sin L \end{bmatrix} \quad (2-1)$$

其中 $\omega_e$ 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

$$\mathbf{x}_{AB}^y + \mathbf{x}_{BC}^y = \mathbf{x}_{AC}^y \quad (2-2)$$

同一个变量在不同坐标系的换算可以用矩阵表示。

$$\mathbf{x}_{\alpha\beta}^\mu = \mathbf{C}_\gamma^\mu \mathbf{x}_{\alpha\beta}^y \quad (2-3)$$

坐标变换矩阵表示旋转关系。例如二维的坐标变换矩阵为

$$\mathbf{C}_n^b = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (2-4)$$

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

$$\mathbf{C}_\mu^y = (\mathbf{C}_\gamma^\mu)^{-1} = (\mathbf{C}_\gamma^\mu)^T \quad (2-5)$$

## 2.4. 惯性导航

### 2.4.1. 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。

### 2.4.2. 姿态更新

三维空间有 3 个旋转自由度。类似式(2-4)，依次绕三个坐标轴旋转，则坐标变换矩阵为

$$\mathbf{C}_n^b = \begin{bmatrix} \cos \theta_y & 0 & -\sin \theta_y \\ 0 & 1 & 0 \\ \sin \theta_y & 0 & \cos \theta_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & \sin \theta_x \\ 0 & -\sin \theta_x & \cos \theta_x \end{bmatrix} \begin{bmatrix} \cos \theta_z & \sin \theta_z & 0 \\ -\sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2-6)$$

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

$$d\mathbf{C} = \begin{bmatrix} 1 & 0 & -d\theta_y \\ 0 & 1 & 0 \\ d\theta_y & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & d\theta_x \\ 0 & -d\theta_x & 1 \end{bmatrix} \begin{bmatrix} 1 & d\theta_z & 0 \\ -d\theta_z & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2-7)$$

略去二阶小量，则有

$$d\mathbf{C} = \begin{bmatrix} 1 & d\theta_z & -d\theta_y \\ -d\theta_z & 1 & d\theta_x \\ d\theta_y & -d\theta_x & 1 \end{bmatrix} \quad (2-8)$$

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

$$[\boldsymbol{\theta}] = \begin{bmatrix} 0 & -\theta_z & \theta_y \\ \theta_z & 0 & -\theta_x \\ -\theta_y & \theta_x & 0 \end{bmatrix} \quad (2-9)$$

那么姿态矩阵更新公式为



$$\mathbf{C}_b^i(t+T) = \mathbf{C}_b^i(t) \lim_{k \rightarrow +\infty} \left( \mathbf{I} + \frac{[\boldsymbol{\theta}_{ib}^b]}{k} \right)^k = \mathbf{C}_b^i(t) \exp([\boldsymbol{\theta}_{ib}^b]) \quad (2-10)$$

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

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

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

$$\exp([\boldsymbol{\theta}]) \approx \mathbf{I} + [\boldsymbol{\theta}] \quad (2-12)$$

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

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

其中  $\theta$  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

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

其中  $\mathbf{A}$  是旋转轴的单位向量。

四元数姿态微分方程为

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

引入 4 维的角增量矩阵

$$[\boldsymbol{\theta}] = \begin{bmatrix} 0 & -\theta_x & -\theta_y & -\theta_z \\ \theta_x & 0 & \theta_z & -\theta_y \\ \theta_y & -\theta_z & 0 & \theta_x \\ \theta_z & \theta_y & -\theta_x & 0 \end{bmatrix} \quad (2-16)$$

四元数更新姿态的公式为

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

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  $\mathbf{b}$  系相对于  $\mathbf{i}$  系的转动，而导航解算中需要计算  $\mathbf{b}$  系相对于  $\mathbf{n}$  系的姿态。导航系  $\mathbf{n}$  系相对于惯性系  $\mathbf{i}$  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  $\mathbf{n}$  系相对  $\mathbf{e}$  系的姿态发生变化，角速率  $\boldsymbol{\omega}_{en}^n$  与速度有关

$$\boldsymbol{\omega}_{en}^n = \begin{bmatrix} -\frac{v_N}{R_m + H} \\ \frac{v_E}{R_p + H} \\ \frac{v_E \tan L}{R_p + H} \end{bmatrix} \quad (2-18)$$

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

$$\boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib}^b - \boldsymbol{\omega}_{ie}^b - \boldsymbol{\omega}_{en}^b \quad (2-19)$$

所以

$$\boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib}^b - \mathbf{C}_n^b(\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \quad (2-20)$$

根据上述角速率计算 $\mathbf{C}_n^b$ ，即姿态。

### 2.4.3. 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

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

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

$$\dot{L} = V_N / R_m \quad (2-22)$$

$$\dot{\lambda} = \frac{V_E}{R_p \cos L} \quad (2-23)$$

其中 $L$ 和 $\lambda$ 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。

## 2.5. 组合导航

### 2.5.1. 原理概述

连续计算惯性导航；当获取主惯导数据时，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。

### 2.5.2. 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：

$$\mathbf{x}_k = \boldsymbol{\Phi} \mathbf{x}_{k-1} + \mathbf{w}_{k-1} \quad (2-24)$$

$$\mathbf{z}_k = \mathbf{H} \mathbf{x}_k + \mathbf{v}_k \quad (2-25)$$

其中是 $\mathbf{x}$ 状态量，是希望获得而又难以准确测量的量。式(2-24)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(2-25)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 $\mathbf{F}$ 矩阵。

状态量的变化也可以描述为连续方程

$$\dot{\mathbf{x}}_k = \mathbf{F}\mathbf{x}_{k-1} \quad (2-26)$$

如果采样间隔足够小，离散方程与连续方程的关系为

$$\Phi = \mathbf{I} + \mathbf{F}T \quad (2-27)$$

其中 $T$ 为采样间隔， $\mathbf{I}$ 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

$$\hat{\mathbf{X}}_{k|k-1} = \Phi\hat{\mathbf{X}}_{k-1} \quad (2-28)$$

$\hat{\mathbf{X}}_{k-1}$ 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

$$\hat{\mathbf{X}}_k = \hat{\mathbf{X}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{X}}_{k|k-1}) \quad (2-29)$$

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

$$\mathbf{P}_{k|k-1} = \Phi\mathbf{P}_{k-1}\Phi^T + \mathbf{Q} \quad (2-30)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R})^{-1} \quad (2-31)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_{k|k-1}(\mathbf{I} - \mathbf{K}_k\mathbf{H})^T + \mathbf{K}_k\mathbf{R}\mathbf{K}_k^T \quad (2-32)$$

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

$$\hat{\mathbf{X}}_k = \mathbf{K}_k\mathbf{z}_k \quad (2-33)$$

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.主惯导修正时，比较惯性导航与主惯导导航的结果偏差，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。

## 2.6. 传递对准

### 2.6.1. 速度位置匹配

速度位置匹配，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个自由度，即

$$\mathbf{x} = [\delta L \quad \delta \lambda \quad \delta h \quad \delta v_E \quad \delta v_N \quad \delta v_U \quad \delta \phi_E \quad \delta \phi_N \quad \delta \phi_U \quad \varepsilon_x \quad \varepsilon_y \quad \varepsilon_z \quad \nabla_x \quad \nabla_y \quad \nabla_z]^T \quad (2-34)$$

组合导航系统关键在于具体列出状态矩阵 $\Phi$ ，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。

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{pp} & \mathbf{F}_{vp} & \mathbf{O}_3 & \mathbf{O}_3 & \mathbf{O}_3 \\ \mathbf{F}_{pv} & \mathbf{F}_{vv} & \mathbf{F}_{av} & \mathbf{O}_3 & \mathbf{C}_b^n \\ \mathbf{F}_{pa} & \mathbf{F}_{va} & \mathbf{F}_{aa} & -\mathbf{C}_b^n & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{O}_3 & \mathbf{O}_3 & \mathbf{O}_3 & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{O}_3 & \mathbf{O}_3 & \mathbf{O}_3 & \mathbf{O}_3 \end{bmatrix} \quad (2-35)$$

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

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

反映速度误差对位置误差影响的子矩阵为

$$\mathbf{F}_{vp} = \begin{bmatrix} 0 & \frac{1}{R_m + h} & 0 \\ \frac{\sec L}{R_p + h} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2-37)$$

反映位置误差对速度误差影响的子矩阵为

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

反映速度误差对速度误差影响的子矩阵为

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

反映姿态误差对速度误差影响的子矩阵为

$$\mathbf{F}_{av} = \begin{bmatrix} 0 & -f_U & f_N \\ f_U & 0 & -f_E \\ -f_N & f_E & 0 \end{bmatrix} \quad (2-40)$$

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。

$$\mathbf{f}_n = \begin{bmatrix} f_E \\ f_N \\ f_U \end{bmatrix} = \mathbf{C}_b^n \mathbf{f}_b \quad (2-41)$$

反映位置误差对姿态误差影响的子矩阵为

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

反映速度误差对姿态误差影响的子矩阵为

$$\mathbf{F}_{va} = \begin{bmatrix} 0 & -\frac{1}{R_m + h} & 0 \\ \frac{1}{R_p + h} & 0 & 0 \\ \frac{\tan L}{R_p + h} & 0 & 0 \end{bmatrix} \quad (2-43)$$

反映姿态误差对姿态误差影响的子矩阵为

$$\mathbf{F}_{aa} = \begin{bmatrix} 0 & \omega_e \sin L + \frac{v_E \tan L}{R_p + h} & -\omega_e \cos L - \frac{v_E}{R_p + h} \\ -\omega_e \sin L - \frac{v_E \tan L}{R_p + h} & 0 & -\frac{v_N}{R_m + h} \\ \omega_e \cos L + \frac{v_E}{R_p + h} & \frac{v_N}{R_m + h} & 0 \end{bmatrix} \quad (2-44)$$

导航计算机每次收到惯性数据时，要计算 $\mathbf{F}$ 矩阵，并更新 $\Phi$ 矩阵。导航计算机收到主惯导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。

观测矩阵为

$$\mathbf{H} = [\mathbf{I} \quad \mathbf{0}] \quad (2-45)$$

### 2.6.2. 姿态匹配

速度位置匹配，取扩展卡尔曼滤波的状态量 $\mathbf{x}$ 为 9 维向量，包含姿态误差、安装误差、陀螺仪零偏各 3 个自由度，即

其中 $\mathbf{F}$ 如下。

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{aa} & \mathbf{0}_3 & -\mathbf{C}_b^n \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \quad (2-46)$$

没有误差时，下面的矩阵应当为单位矩阵

$$\mathbf{M}_z = \mathbf{C}_m^n \mathbf{C}_s^m \mathbf{C}_n^s \quad (2-47)$$

其中 $\mathbf{C}_n^s$ 是子导航系统的姿态矩阵，由子导航系统惯性导航解算得到。 $\mathbf{C}_m^n$ 是主导航系统的姿态矩阵，由主导航系统数据得到。 $\mathbf{C}_s^m$ 是主导航系统和子导航系统的安装误差矩阵，会根据 EKF 的计算结果逐渐修正。注意这三个矩阵的上下标定义有所区别。把上述矩阵换算为三个角，即观测量

$$\mathbf{z}_2 = \begin{bmatrix} \mathbf{M}_z(3,2) - \mathbf{M}_z(2,3) \\ \mathbf{M}_z(1,3) - \mathbf{M}_z(3,1) \\ \mathbf{M}_z(2,1) - \mathbf{M}_z(1,2) \end{bmatrix} / 2 \quad (2-48)$$

观测矩阵为

$$\mathbf{H}_2 = [\mathbf{I}_3 \quad -\mathbf{C}_m^n \quad \mathbf{0}_3] \quad (2-49)$$

## 3. 著作权和服务

### 3.1. 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”。

### 3.2. 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。

### 3.3. 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。

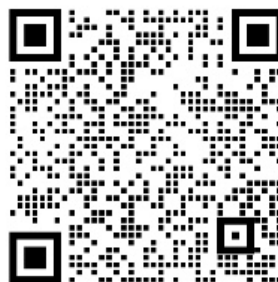
### 3.4. 联系方式

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](mailto:braun@wmsoft.wang)

网站：<http://wmsoft.xyz>