

导航算法示例代码包 CPP

著作权声明

本店保留著作权。

本说明书、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。

西安市雁塔区雾膜软件开发站

244917988@qq.com

陕西省西安市雁塔区电子正街 88 号 3 幢 21109 室

1. 内容介绍

本代码包的功能主要有：根据陀螺仪和加速度计信息计算惯性导航；根据卫星定位信息计算组合导航。

此次发货只包含了 C++ 程序包。Matlab 和 C 语言程序虽然在本说明书中提及，但是未实际发货。

本代码为了便于学习理解，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。本代码包充分照顾了通用性、可移植性；但是一些特定的计算平台，具有计算量或者内存量的限制，或者有专门的数学计算库，这些内容没有纳入本代码包。需要其余功能的用户可以联系本店定制开发。

2. 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。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。

3. MATLAB 代码包

3.1. 主程序

主程序为 instance.m。直接运行即可。

默认的工作模式为：导航的初始姿态存在故意设定的误差；导航装置静止；组合导航过程中，利用卫星数据修正导航误差。如果需要其它工作模式，直接更换相应数据即可。

程序主要工作流程为：

```

设定初值
循环
{

```

```

    惯性导航计算
    更新用于组合导航的变量
    如果, 间隔一段时间收到卫星数据
    {
        根据卫星计算组合导航
    }
    保存数据
}

```

3.2. 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。滤波参数的具体数值可以参考“调试技巧”章节。

3.3. 导航计算

惯性导航计算为 $insgyroacc(gyro, acc, atti1, speed1, pos1, dTins)$ 。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。

3.4. 导航结果

数据保存于 $dataA$ 矩阵。前 9 列为 3 个姿态角、3 个速度、3 个位置。姿态欧拉角单位为度。用于画图或保存。

4. C 和 C++代码包

4.1. 主程序

主函数为 $main.c$ 或 $main.cpp$ 的 $instance()$ 。在 VS 平台直接运行即可。C 和 C++代码包的导航计算内容与 MATLAB 基本一致。

qa 为姿态四元数, $tpos$ 为位置, $tspeed$ 为速度, $dTins$ 为时间间隔。根据需要修改。

卡尔曼滤波中, 初始状态的方差矩阵为 Pk 。测量的方差矩阵为 R 。系统噪声的方差矩阵为 Q , 与时间和 $Q0$ 有关。在 $para()$ 中改变参数设定值。

惯性导航计算为 $ins_gyroacc(gx, gy, gz, ax, ay, az)$ 。输入实际的陀螺仪和加速度计数据即可, 单位为 rad/s 和 $m/s/s$ 。

卫星数据的计算为 $sat(lati, longi, height, ve, vn, vu)$ 。输入纬度经度高度、东北天速度即可。

数据保存于 $d.txt$ 文件。前 9 列为 3 个姿态角、3 个速度、3 个位置。用于画图或保存。注意: 嵌入式系统中, 文件读写操作可能不兼容; 要去掉保存文件有关的代码。

4.2. 矩阵计算

为了矩阵计算, C++代码中有类 MAT , C 代码中有结构体 MAT 。其内容为:

```

int m; // 行数
int n; // 列数

```

```
double num[MAT_MAX][MAT_MAX]; //矩阵数据内容
```

可以根据需要直接修改矩阵的数值。特别注意，C 或 C++ 中数组元素的下标从 0 开始；而 MATLAB 的下标从 1 开始。

矩阵计算的功能已经包含在代码包中，通常情况下不需要修改。

如果要编写嵌入式系统的导航程序，建议矩阵计算部分替换为计算平台的专用库以改善效率。

5. 计算原理

本文描述核心思想和关键步骤，具体公式可以参考本代码包的 MATLAB 程序或相关书籍。

5.1. 惯性导航

5.1.1. 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。

5.1.2. 姿态更新

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 `wien` 的影响。此外，由于地球是球形的，位置变化时地面的角度就变了，所以速度会导致一个角速度 `wenn`。所以计算姿态时使用的角速度为 `wnbb=gyro1-Cbn'*(wien+wenn)`。

5.1.3. 速度更新

加速度要从传感器的坐标系换算到东北天，即 `accn=Cbn*acc1`。

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。所以计算姿态时使用的加速度为 `an=accn-cross(wien+wien+wenn,speed)+gn;`

5.1.4. 位置更新

速度积分即位置。但是需要除以地球半径、换算为经纬度。

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

5.2. 组合导航

5.2.1. 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。

5.2.2. 卡尔曼滤波

扩展卡尔曼滤波的求解过程针对导航做了专门简化，求解过程为：

```
Pkk=Phi*Pk*(Phi')+Q;
K=Pkk*(H')/(H*Pkk*(H')+R);
X=K*Z;
IKH=eye(15)-K*H;
Pk=IKH*Pkk*(IKH')+K*R*(K');
```

扩展卡尔曼滤波的状态方程是惯性卫星组合导航中较为复杂的公式，即程序中的 Φ_{k1} （或 Φ_k ）矩阵。 Φ_{k1} 矩阵由 F_k 矩阵得到， F_k 由函数 `getFk` 给出。

6. 调试技巧

6.1. 一般调试

只要给出正确的传感器数据和初值，本代码包即可给出导航结果。

如果没有取得预期效果，可以按照下面方法调试：

（1）核对初值和传感器数据的正确性、对应轴向、以及物理单位。要给定一个比较准确的初值，包括位置、速度、姿态。传感器要标定，去除粗大的传感器误差。特别要注意单位、正负号、轴向等对应关系。

（2）先注释掉卫星的部分，按照纯惯性导航计算；纯惯导计算无误后，再调试卫星和组合导航的计算。

（3）组合导航先按照 9 维度调试，即 P_{k1} 矩阵和 Q_1 矩阵的对角线最后面 6 个数都是 0。9 维调试成功后，再调试 15 维的。一般维数越多，收敛越困难；如果 9 维能满足要求，可以不使用 15 维。

（4）先静态实验；静态实验无误后，再动态实验。往往需要机动，扩展卡尔曼滤波才能达到更好的精度。

6.2. 卡尔曼滤波参数

先确保稳定性，再调整快速性和准确性。

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。如果有实验数据，可以直接根据实验数据设定 P_k 和 R 。

开始时 Q 尽量小一些，等待滤波器调试稳定后，逐步调大一点点 Q 。

参考资料

Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration（作者 Aboelmagd Noureldin, Tashfeen B. Karamat, Jacques Georgy，出版商 Springer）（预览版 <https://www.doc88.com/p-2106920107569.html>）