一、坐标系定义

1、IMU坐标系定义：前右下（右手系）

2、GNSS坐标系定义：ENU

3、电子罗盘坐标系定义：前右下

二、基于ekf的姿态跟踪算法框架

1. 传感器组合方案

单轴角度跟踪：

1. IMU（双轴加速度+单轴陀螺仪）
2. IMU+GNSS（单轴陀螺仪+双/单天线）
3. IMU+COMPASS（单轴陀螺仪+磁力计）
4. 算法接口

int AngleTrack\_Process(Config\_t\* conf, double gyo,double IMU\_time,double ExterAngle, double\* result);//陀螺仪与外部角度融合，输出单轴角度

3、数据协议

// 惯性观测数据

struct IMUdata

{

double imutimetarget; //IMU输出的时间戳，秒/毫秒，100Hz，计算采样时间间隔

double accx; //加速度计X轴输出 g

double accy; // Y g

double accz; // Z g

double gyox; //陀螺X轴输出， deg/s

double gyoy; // Y deg/s

double gyoz; // Z deg/s

int state\_acc; //IMU标志位，例如加速度计超量程状态位

};

typedef struct IMUdata IMUdata\_t;

// 电子罗盘观测数据

struct Compassdata

{

int comass\_type; //1 平面电子罗盘 2 三维电子罗盘

double accx; //加速度计X轴输出 g

double accy; // Y g

double accz; // Z g

double mx; //磁力计X轴输出， 高斯

double my; // Y 高斯

double mz; // Z 高斯

};

typedef struct Compassdata Compassdata\_t;

//GNSS观测数据

struct GNSSdata

{

double gnsstimetarget; //GNSS观测的时间戳，周内秒，20Hz

double lat; //纬度, WGS84 deg

double lon; //经度，WGS84 deg

double alt; //高程，WGS84 m

double gnss\_v[3]; //gnss速度，ecef/enu m/s

double heading; //双天线航向 deg

int state\_pos; //GNSS位置解算状态

int state\_yaw; //GNSS航向解状态

int heading\_flag; //gnss航向类型 0.双天线 1.单天线

};

typedef struct GNSSdata GNSSdata\_t;

//配置文件信息

struct Config

{

double abiasx; //加速度计x轴静态零偏 g

double abiasy; //加速度计y轴静态零偏 g

double abiasz; //加速度计z轴静态零偏 g

double gstdxthr; //陀螺仪静态阈值 deg/s

double gstdythr; //陀螺仪静态阈值 deg/s

double gstdzthr; //陀螺仪静态阈值 deg/s

double astdxthr; //加速度计静态阈值 g

double astdythr; //加速度计静态阈值 g

double astdzthr; //加速度计静态阈值 g

double gnoisex; //x轴角速度噪声 rad

double gnoisey; //y轴角速度噪声 rad

double gnoisez; //z轴角速度噪声 rad

double gstabilityx; //x轴陀螺仪零偏稳定性 rad/h

double gstabilityy; //x轴陀螺仪零偏稳定性 rad/h

double gstabilityz; //x轴陀螺仪零偏稳定性 rad/h

double gnssyawstd\_thr; //双天线航向角阈值 rad

double gnssyawvar; //双天线航向角量测噪声 rad

double install\_roll; //横滚角安装误差 deg

double install\_pitch; //俯仰角安装误差 deg

double install\_heading; //航向角安装误差 deg

double acc\_norm; //干扰加速度

int ekf\_strategy; //ekf量测噪声策略 1.加速度平滑滤波 2.动静态区分 3.其他策略

int angle\_flag; //输出角度类型 1、IMU单轴角度跟踪 2、单轴陀螺+外部角度

};

typedef struct Config Config\_t;//姿态跟踪观测数据

struct ATdata

{

IMUdata\_t imu; //IMU观测数据

GNSSdata\_t gnss; //GNSS观测数据

Compassdata\_t compass; //compass观测数据

Config\_t config; //配置文件结构体

int bimu\_updta; //IMU观测数据更新标志：0-未更新 1-更新

int bgnss\_updata; //GNSS观测数据更新标志： 0-未更新 1-更新

int bcompass\_updata; //compass观测数据更新标志 ： 0-未更新 1-更新

};

typedef struct ATdata ATdata\_t;

需要配置传感器的组合类型和输出角度类型。

4、demo说明

int AngleTrack\_Process\_Demo(ATdata\_t \*atd, Config\_t\* conf, double\* result);//demo函数

a、例子1：IMU单轴角度融合

if (conf->angle\_flag == 1)//陀螺仪+加速度计

{

acc\_flag = IMU\_AccAngle\_Process(&atd->imu,conf, &ExterAngle);//加速度计算角度

}

重点需要配置的信息：bimu\_updta=1，bgnss\_updata=0.bcompass\_updata=0，angle\_flag=1；

b、例子2：IMU+GNSS单轴角度融合

if (conf->angle\_flag == 2 && atd->bgnss\_updata == 1 && atd->bcompass\_updata == 0)//陀螺仪+gnss

{

gnss\_flag = GnssHeading\_Process(&atd->gnss, &ExterAngle);//单天线计算航向角

}

重点需要配置的信息：bimu\_updta=1，bgnss\_updata=1.bcompass\_updata=0，angle\_flag=1；

C、例子3：IMU+COMPASS单轴角度融合

if (conf->angle\_flag == 2 && atd->bgnss\_updata == 0 && atd->bcompass\_updata == 1)//陀螺仪+磁力计

{

compass\_flag = CompassHeading\_Process(&atd->compass, &ExterAngle);//磁力计计算航向角

}

重点需要配置的信息：bimu\_updta=1，bgnss\_updata=0.bcompass\_updata=1，angle\_flag=1；