通过xyz加速度和陀螺仪计算姿态角（欧拉角）

#coding:utf-8

import math

#IMU算法更新

Kp = 100 #比例增益控制加速度计/磁强计的收敛速度

Ki = 0.002 #积分增益控制陀螺偏差的收敛速度

halfT = 0.001 #采样周期的一半

#传感器框架相对于辅助框架的四元数(初始化四元数的值)

q0 = 1

q1 = 0

q2 = 0

q3 = 0

#由Ki缩放的积分误差项(初始化)

exInt = 0

eyInt = 0

ezInt = 0

def Update\_IMU(ax,ay,az,gx,gy,gz):

global q0

global q1

global q2

global q3

global exInt

global eyInt

global ezInt

# print(q0)

#测量正常化

norm = math.sqrt(ax\*ax+ay\*ay+az\*az)

#单元化

ax = ax/norm

ay = ay/norm

az = az/norm

#估计方向的重力

vx = 2\*(q1\*q3 - q0\*q2)

vy = 2\*(q0\*q1 + q2\*q3)

vz = q0\*q0 - q1\*q1 - q2\*q2 + q3\*q3

#错误的领域和方向传感器测量参考方向之间的交叉乘积的总和

ex = (ay\*vz - az\*vy)

ey = (az\*vx - ax\*vz)

ez = (ax\*vy - ay\*vx)

#积分误差比例积分增益

exInt += ex\*Ki

eyInt += ey\*Ki

ezInt += ez\*Ki

#调整后的陀螺仪测量

gx += Kp\*ex + exInt

gy += Kp\*ey + eyInt

gz += Kp\*ez + ezInt

#整合四元数

q0 += (-q1\*gx - q2\*gy - q3\*gz)\*halfT

q1 += (q0\*gx + q2\*gz - q3\*gy)\*halfT

q2 += (q0\*gy - q1\*gz + q3\*gx)\*halfT

q3 += (q0\*gz + q1\*gy - q2\*gx)\*halfT

#正常化四元数

norm = math.sqrt(q0\*q0 + q1\*q1 + q2\*q2 + q3\*q3)

q0 /= norm

q1 /= norm

q2 /= norm

q3 /= norm

#获取欧拉角 pitch、roll、yaw

pitch = math.asin(-2\*q1\*q3+2\*q0\*q2)\*57.3

roll = math.atan2(2\*q2\*q3+2\*q0\*q1,-2\*q1\*q1-2\*q2\*q2+1)\*57.3

yaw = math.atan2(2\*(q1\*q2 + q0\*q3),q0\*q0+q1\*q1-q2\*q2-q3\*q3)\*57.3

return pitch,roll,yaw

原文：<https://blog.csdn.net/Forter_J/article/details/85278335>