

# IMU 运动识别技术文档

陈冲 2015/8/27

## 1. 模块简介

IMU 运动识别模块主要目的是通过对采集到的 IMU 信号（加速度 acc，角速度 gyr）的分析与计算，来对传感器进行运动状态识别（静止，振动，移动和旋转）。

## 2. 源代码以及数据文件

C 代码：包括 include/uno\_motionRecog.h 和 driver/uno\_motionRecog.c，这两个文件分别放置了识别所有的函数接口和实现细节。是进行实时识别必须需要的文件。

python 代码：在 analysisResults/文件夹下，主要包含了前期数据分析和算法原型设计，以及一些可视化的展现。

数据与标注：数据文件在 data/目录下，相应的标注文件在 label/下，这两个文件主要保存了一些采集的 IMU 数据，用于进行离线分析。

## 3. 函数接口

uno\_motionRecog.h 文件中主要包含了两个接口：

void motionRecogInitCB(MOTION\_STATE\_CB motion\_cb, float period, int Fs)，该函数是带有回调功能的识别 api，motion\_cb 是用户自己定义的回调函数，period 是滑动窗口的步长，单位 s，可以控制识别速度；参数 Fs 是 IMU 信号的采样频率。另外该函数内部通过开启一个新的线程来进行识别。

void motionRecogInt(float period, int Fs)，该接口是不带回调功能的识别函数，也不开启新线程，功能与前一个一样。

## 4. 重要功能实现

void getEnergy(float x[], float y[], float z[], float energy[], int N)，该函数主要利用 x,y,z 三轴来计算能量，包括 acc 的能量和 gyr 的能量，识别算法中需要根据能量来设置一些阈值，因此能量的计算非常重要。能量计算公式如下：

$$\text{energy} = \sqrt{x^2 + y^2 + z^2} \quad (1)$$

void calZeroCrossRate(float x[], int N)；该函数主要用于计算 gyr-x,y,z 的过零率，过零率表示信号由正到负，或者由负到正的次数，因为旋转的时候过零率基本为 0，所以对于旋转的识别非常有用。

float calEntropy(float energy[], int N)；该函数功能主要是计算信号的信息熵，因为在数据分析阶段发现振动事件的信号能量非常集中，信息熵很小，因此信息熵对于振动的识别很有用。公式如下，这里 energy 需要经过归一化

$$\text{entropy} = -\sum \text{energy}_i * \log(\text{energy}_i) \quad (2)$$

void correctBias(float arr\_gyrX[], float arr\_gyrY[], float arr\_gyrZ[], float energy\_acc[], int N)；该功能函数主要用于信号的 bias 消除，因为 IMU 或多或少都有 bias，而且不同姿势下的 bias 还会变化，对于后续的识别和能量计算都会有影响。但是功能只在静止的

状态下进行，各个变量的 bias 估计都是随时间在变化，因此 bias 的消除是一种随运动自适应的过程。

## 5. 运动识别的总体流程

算法细节可以参考函数：

```
int recognizeState(float arr_gyrX[], float arr_gyrY[], float arr_gyrZ[], float energy_acc[], float energy_gyr[], int interval);
```

识别算法中主要包含 8 个参数阈值：

T\_period 和 T\_entropy 分别代表时间长度和信息熵阈值，用于振动识别。

T\_acc\_energy\_slow, T\_gyr\_energy\_slow 分别代表 acc 和 gyr 的能量阈值，用于动静识别。

T\_gyr\_energy\_sum\_rotate\_lower, T\_gyr\_energy\_sum\_rotate\_upper, 代表 gyr 的能量和阈值，用于旋转识别

T\_gyr\_energy\_sum\_vibration, gyr 的能量和阈值，用于振动识别

T\_gyr\_energy\_sum\_acc, gyr 的能量和阈值，用于移动识别

